N° d'ordre 748

Année 1981

THÈSE

présentée à

L'UNIVERSITÉ PAUL SABATIER DE TOULOUSE

en vue de l'obtention

du Diplôme de Docteur-Ingénieur

par

J.-P. SOUBRIER

UN MODELE DE RESOLUTION DE PROBLEMES D'ORDONNANCEMENT DYNAMIQUE

soutenue le 23 juin 1981 devant le Jury composé de

MM. J. VIGNOLLE — Président

J. CARPENTIER
J. NOAILLES
F. RODRIGUEZ
F. ROUBELLAT
} Examinateurs

N° d'ordre 748

Année 1981

THÈSE

présentée à

L'UNIVERSITÉ PAUL SABATIER DE TOULOUSE

en vue de l'obtention

du Diplôme de Docteur-Ingénieur

par

J.-P. SOUBRIER

UN MODELE DE RESOLUTION DE PROBLEMES D'ORDONNANCEMENT DYNAMIQUE

soutenue le 23 juin 1981 devant le Jury composé de

MM. J. VIGNOLLE	Président
J. CARPENTIER	Examinateurs
J. NOAILLES	
F. RODRIGUEZ	
F. ROUBELLAT	

UNIVERSITE PAUL SABATIER

PRESIDENCE

M. MARTIN .... Président
M. LARENG .... 1er Vice-Président
M. MARPINARD ..... 2ème Vice-Président

ORDRE DES SCIENCES

HONORARIAT

M. AGID .... Professeur honoraire
M. BEDOS .... Professeur honoraire
M. BLAIZOT .... Doyen honoraire
M. CAPDECOMME .. Doyen honoraire, Recteur honoraire, Correspondant de l'Institut, Professeur honoraire
Mlle De FERRE ..... Professeur honoraire
M. DUPOUY ...... Membre de l'Institut, Doyen honoraire, Directeur honoraire du C.N.R.S., Professeur honoraire
M. DURAND Emile .. Doyen honoraire, Professeur honoraire
M. FERT .... Professeur honoraire
M. GALLAIS ..... Professeur honoraire
M. GAUSSEN ..... Professeur honoraire, Correspondant de l'Institut
M. LESBRE ..... Professeur honoraire
M. MARGULIS .... Professeur honoraire
M. MASDUPUY .... Professeur honoraire
M. MATHIS ...... Doyen honoraire
M. MIGNONAC ..... Professeur honoraire
M. MORQUER ...... Professeur honoraire, Correspondant de l'Institut
M. ORLIAC ..... Professeur honoraire
M. PERRIER ..... Professeur honoraire
M. SECONDAT ..... Professeur honoraire
M. SERFATY ...... Professeur honoraire
M. TEISSIE-SOLIER . Professeur honoraire
M. TRICHE .... Professeur honoraire

EMERITAT

M. AGID
M. CAPDECOMME
Mlle De FERRE
M. DUPOUY
M. FERT
M. GALLAIS
M. LESBRE
M. ORLIAC
M. SERFATY

CORPS ENSEIGNANT

PROFESSEURS DE CLASSE EXCEPTIONNELLE ET DE 1ère CLASSE

M. HURON .... Mathématiques Appliquées
M. LEDOUX .... Zoologie Appliquée
M. MATHIS .... Chimie
M. LAFOURCADE .. Physique
M. ANGELIER ..... Zoologie
M. FARRAN .... Minéralogie et Géotechnique
M. LAUDET .... Physique Théorique et Calcul Numérique
M. LAGASSE .... Electrotechnique
M. BLANC .... Physique Nucléaire
M. LEREDDE ...... Botanique
M. LELUBRE ...... Géologie
M. LALAGUE ..... Mathématiques Générales
M. BOUIGUE .... Astronomie
M. ASSELINEAU .... Chimie Biologique
M. MAURET .... Chimie Systématique
M. MONTANT ...... Cryptogamie
M. GAUTIER ...... Physique
M. CRUMEYROLLE .. Mathématiques
M. GOURINARD .... Géologie
M. PULOU ...... Minéralogie
M. CAMBOU ...... Physique Spatiale
M. LACOSTE ...... Electrotechnique
M. THIBAULT ..... Mécanique Rationnelle et Appliquée
M. MASCART ..... Mathématiques
M. MEDIONI ..... Psychophysiologie
M. RAYNAUD P ..... Physiologie Animale
M. ZALTA .... Chimie Biologique
M. SEVELY ...... Electrotechnique
M. POMMIEZ ..... Mathématiques
M. REY Paul ...... Biologie Végétale
M. COULOMB ..... Physique
M. TRINQUIER ..... Physique
M. MARONI ...... Chimie
M. BEETSCHEN ..... Biologie Générale
M. DERACHE ..... Physiologie Animale
M. SATGE .... Chimie Organique
M. LATTES .... Chimie
M. VEDRENNE ..... Géophysique
M. DURAND-DELGA Géologie, Correspondant de l'Institut
M. CARRARA ..... Physique
M. MAHENC ...... Chimie
M. MIROUSE ..... Géologie
M. BITSCH ...... Zoologie
M. DEGEILH ...... Physique
M. MARTIN J.C. .... Génie Electrique
M. REY Gérard ..... Génie Electrique
M. SICARD ...... Biologie Génétique
M. SOUQUET ...... Géologie
M. TOUZE ...... Physiologie Végétale
M. FRASNAY ...... Mathématiques (Algèbre et Combinatoire)
M. CASSAGNAU ... Zoologie
M. CAUSSINUS .... Mathématiques Appliquées (Statistiques Appliquées)
M. PESCIA ...... Physique
M. PICCA ...... Physique de l'Atmosphère
M. BAUDIERE .... Botanique Fondamentale et Pyrénéenne
M. BARRANS .... Chimie Physique Organique
M. POILBLANC ..... Chimie Minérale
M. PERENNOU ..... Informatique
M. ATTEIA ...... Mathématiques
M. CASTAN ...... Informatique
M. COLLETTE ..... Physique
M. REME .... Mesures Physiques
M. CUPPENS .... Mathématiques
M. BAUDRAS ...... Chimie Biologique

PROFESSEURS DE 2ème CLASSE

M. MERIC	Mathématiques Appliquées
Mme LECAL	Zoologie
M. PILOD	Physique
M. LARROQUE	Physique
Mme LAUDET LAPEYRE	Mathématiques - Informatique
M. BERTRAND	Chimie
M. DESQ	Mathématiques
M. ROCARD	Electronique
M. GUERIN	Mathématiques
M. SCHNEIDER	Biologie Cellulaire
M. de LOTH	Chimie Physique
M. SAPORTE	Physique
M. THENOZ	Génie Civil
M. DURAND Ph.	Physique
M. FONTAN	Physique Nucléaire
M. CALVET	Mécanique des Fluides
M. PAGANI	Physique
M. BEAUFILS	Informatique
M. BERTHELEMY	Zoologie
M. TERJANIAN	Mathématiques
M. MORUCCI	Génie Biologique et Médical
M. BONEL	Chimie
M. SOTIROPOULOS	Chimie Organique
M. VERDIER	Physique
M. ETTINGER	Mathématiques
M. BONNET Louis	Biologie
M. JOSSERAND	Mesures Physiques
M. ROUTIE	Génie Chimique
M. COTTU	Génie Mécanique
M. HURAUX	Physique
Mme GERVAIS	Chimie Inorganique
M. BANCEL	Mathématiques
M. LOUARN	Génétique
M. BOUDET	Physiologie Végétale
M. LETAC	Mathématiques
M. HERAULT	Chimie
M. GRANDET	Génie Civil
Mlle BARBANCE	Mathématiques
M. GILLY	Génie Mécanique
M. MARAL	Mesures Physiques
M. LEGRAND	Génie Civil
M. ABATUT	Electronique, Electrotechnique, Automatique
M. MAUSS	Mécanique
M. BETOURNE	Informatique
M. CAMPAN	Psychophysiologie
M. CLERC	Mécanique
M. GRIFONE	Mathématiques
M. COUOT	Mathématiques, Analyse Numérique
M. NGUYEN THANH VAN	Mathématiques
M. TRAVERSE	Problèmes Chimiques de l'Energie
M. ALRAN	Génie Chimique
M. REY J.	Géologie Sédimentaire et Paléontologie
M. DARTIGUENAVE	Chimie Minérale Moléculaire
M. PRADINES	Mathématiques
M. GALINIER	Informatique
M. VIGNOLLE	Informatique
M. DEPARIS	Embryologie
M. CAVALIE	Physiologie Végétale
M. MASSOL	Chimie des Composés Organiques et Organominéraux d'intérêt biologique
M. HARTMANN	Mécanique
M. ROUSSET	Chimie Appliquée (Matériaux)
M. TARDY	Sciences de la Terre et Aménagement
M. HOLLANDE	Biologie Cellulaire
M. DUGAS	Physique des Energies Nouvelles
M. BENOIT-CATTIN	Physique
M. COMTAT	Chimie Appliquée
M. LANEELLE	Biochimie
M. LUGUET	Informatique Fondamentale et Appliquée
M. BONNET J.J.	Chimie Minérale
M. PERAMI	Minéralogie et Matériaux
M. AUDOUNET	Mathématiques
M. PERIE	Chimie Organique
M. AMBID	Physiologie
M. AURIOL	Biologie
M. COURVOISIER	Electronique
M. FORTUNE	Géologie
Mlle RIVIERE	Chimie
M. TIRABY	Biologie
M. BARONNET	Thermodynamique Energétique
M. COMBES	Génie Electrique
M. DUBAC	Mesures Physiques

PROFESSEURS ASSOCIES

M. GUMOWSKI	Mathématiques
M. COLLINS	Génie Mécanique

CHERCHEURS DU C.N.R.S.

DIRECTEURS DE RECHERCHE

M. ESTEVE Daniel
M. GALY Jean
M. GIRALT Georges
M. LABARRE Jean-François
M. LAURENT Jean-Pierre
M. LEGRIS
M. MARTINOT Henri
M. MAZEROLLES
M. PRADAL
M. WOLF Robert

MAITRES DE RECHERCHE

M. AGUILAR-MARTIN José
Mme ASSELINEAU Cécile
M. AZEMA Pierre
Mme BENAZETH Nicole
M. BUXO Jean
Mme DARTIGUENAVE M.
Mme DUPRAT Anne-Marie
M. HAWKES Peter
M. HOUALLA Doureid
M. JEREBZOFF
M. MALRIEU J.P.
Mme MARONI Yvette
Mme MATHIS
M. MUNOZ Aurélio
M. NAVECH
M. PRAJOUX Roland
M. SEVELY Jean
M. VACQUIE Serge

CORPS DES OBSERVATOIRES ASTRONOMIQUES
ET INSTITUTS DE PHYSIQUE DU GLOBE

Mme ANDRILLAT Y.	Astronome titulaire
M. COUPINOT G.	Astronome adjoint
M. LEROY J.Louis	Astronome adjoint
M. MIANES	Astronome adjoint
M. PEDOUSSAUT A.	Astronome adjoint
M. ROBLEY Robert	Physicien titulaire
M. ROSCH Jean	Astronome titulaire
M. ROZELOT J.P.	Physicien adjoint
M. SAISSAC Joseph	Physicien titulaire

ADMINISTRATION

M. PRINEAU	Secrétaire Général de l'Université

Liste établie au 1er Octobre 1980 pour l'année universitaire 1980 - 1981

Je tiens à exprimer ma très sincère reconnaissance à tous les membres du jury

à Monsieur le Professeur Jean VIGNOLLE qui a bien voulu présider le jury de cette thèse,

à Monsieur le Professeur Joseph NOAILLES qui a été à l'origine de ce travail,

à Monsieur CARPENTIER, Conseiller Scientifique à l'E.D.F. et Monsieur ROUBELLAT, Maître de Recherche C.N.R.S. au L.A.A.S., pour l'intérêt qu'ils ont porté à mon travail,

à François RODRIGUEZ, Maître-Assistant dont les critiques constructives et les encouragements incessants m'ont permis de mener à bien cette thèse.

Je tiens à remercier enfin mes collègues du Département Informatique de l'E.N.S.E.E.I.H.T. et notamment Monsieur A. BRUEL, son Directeur.

Merci également à Annie BOIDE et Marie-Blandine VOLPATO qui ont assuré la dactylographie de ce document avec leur habituel dévouement ainsi qu'à Monsieur BERNARD et l'équipe de reprographie qui en ont réalisé le tirage.

Un modèle de Résolution de Problèmes d'Ordonnancement Dynamique

J.P. Soubrier

SOMMAIRE

INTRODUCTION

CHAPITRE I PRESENTATION DES PROBLEMES D'ORDONNANCEMENT 1

I.1. Contexte et définition de l'ordonnancement 3

I.1.1. Le contexte de l'ordonnancement d'atelier 3

I.1.2. Définition du problème 4

I.1.2.1. Tâches et machines 5

I.1.2.2. Restrictions sur les tâches et les machines 6

I.1.2.3. Différents types de problèmes d'ordonnancement 7

I.2. Objectifs de l'ordonnancement et étude des critères 8

I.2.1. Objectifs et classement des critères 9

I.2.2. Les critères simples 12

I.2.2.1. Critère de la date d'achèvement 12

I.2.2.2. Critère du temps total de passage dans l'atelier 15

I.2.2.3. Critère des délais 16

I.2.2.4. Critère des en-cours 19

I.2.2.5. Critère de l'utilisation 22

I.2.2.6. Critère d'adaptation des machines 24

I.2.3. Etude des multicritères 25

I.2.4. Temps et coûts 27

CHAPITRE II	PROBLEMES D'ORDONNANCEMENT STATIQUES : MODELES ET METHODES DE RESOLUTION	29
II.1.	Formulation des problèmes d'ordonnancement	31
II.1.1.	Programmes linéaires en nombres mixtes	31
	II.1.1.1. Problèmes n/m/G/P et n/M/G/P	31
	II.1.1.2. Formulation de MANNE pour le problème n/m/G/V	35
	II.1.1.3. Conclusion	37
II.1.2.	Contraintes disjonctives dans les graphes	38
	II.1.2.1. Ordonnancement d'atelier simple n/m/G/P	40
	II.1.2.2. Ordonnancement d'atelier multiple n/M/G/P	42
	II.1.2.3. Limites des graphes disjonctifs : extension proposée	43
	II.1.2.4. Conclusion	45
II.1.3.	Conclusion sur les formalisations	45
II.2.	Les méthodes de résolution	46
II.2.1.	Méthodes analytiques combinatoires	46
II.2.2.	Programmation linéaire en nombres mixtes	47
	II.2.2.1. Une expérience de résolution par P.L.M.	47
	II.2.2.2. Des possibilités limitées	49
	II.2.2.3. Approche par fonction de pénalisation	49
II.2.3.	Enumération implicite	50
	II.2.3.1. Algorithme SEP	51
	II.2.3.2. Enumération implicite de BALAS	52
	II.2.3.3. Méthodes heuristiques	53
II.3.	Cas particuliers résolus	54
II.3.1.	Problèmes n/1//V	55
II.3.2.	L'atelier à flot unique	55
II.3.3.	L'atelier à flot quelconque	56
II.4.	Conclusion sur les problèmes statiques	57

CHAPITRE III LES PROBLEMES D'ORDONNANCEMENT DYNAMIQUE : UN MODELE ET UNE METHODE DE RESOLUTION 59

III.1. Le problème dynamique 61

III.1.1. Justification du choix "contexte dynamique" 61

III.1.2. Les données 61

III.1.2.1. Les machines 62

III.1.2.2. Les tâches 62

III.1.3. Les hypothèses 63

III.1.4. Les objectifs 64

III.1.5. Les choix possibles 66

III.2. Le modèle 70

III.2.1. Formalisation des conditions 70

III.2.2. Période et unité de temps 72

III.2.3. Objectifs et principes du modèle 74

III.2.4. Etude du chargement 75

III.2.4.1. Discussion sur le cadrage initial 75

III.2.4.2. Mode de chargement 78

III.2.5. Régulation de la charge 85

III.2.5.1. Problème R_p 85

III.2.5.2. Principe de la régulation 86

III.2.6. Algorithme général 88

CHAPITRE IV	APPLICATIONS ET RESULTATS	91
IV.1.	Implémentation du modèle	92
IV.2.	Exemples résolus et résultats obtenus	93
IV.3.	Exemple simple de régulation	100
IV.4.	Discussion sur les hypothèses	103
CONCLUSION		106
BIBLIOGRAPHIE		110
ANNEXE A	Notations utilisées	
ANNEXE B	Résolution graphique du problème n/m/F/P	
ANNEXE C	Listing du Programme	

INTRODUCTION

La réalisation de projets complexes tels que la construction d'un immeuble, la détermination du cycle de fabrication d'un atelier ou l'établissement d'un emploi du temps, imposent généralement l'accomplissement de tâches nombreuses et variées soumises à des contraintes de types très divers. Etablir un ordonnancement, c'est déterminer l'ordre et le calendrier d'exécution de l'ensemble de ces tâches en satisfaisant, si possible, l'ensemble des contraintes. En général ce problème a plusieurs solutions admissibles et il faut déterminer celle qui optimise un critère fixé à l'avance. Ce critère peut être la minimisation d'un délai de réalisation, la maximisation d'un profit, le lissage des effectifs de la main d'oeuvre, l'étalement des charges financières ou tout autre critère économique.

Le problème de l'ordonnancement comporte donc deux parties essentielles : la recherche d'une solution compatible avec les contraintes et l'optimisation de celle-ci.

De nombreux modèles de résolution ont déjà été proposés, mais ils appartiennent tous à deux grandes classes suivant la nature des contraintes considérées : les contraintes dites "potentielles" et les contraintes "cumulatives et disjonctives". A titre d'exemple, les contraintes de localisation temporelle ou les contraintes de succession sont potentielles. Par contre des limites sur les disponibilités ou la disjonction, dans le temps, de l'exécution de deux tâches forment des contraintes cumulatives et disjonctives.

Pour le premier type de problème, à contraintes potentielles, le diagramme de Gantt, ou planning à barres, a été couramment utilisé jusqu'en 1958. Mais à partir de cette année là se sont développés deux types de méthodes fondées sur le calcul de chemin ou de tensions dans un graphe valué. Ce sont la méthode de chemin critique P.E.R.T. et la méthode des potentiels de B. ROY. Ces méthodes ont apporté une amélioration considérable par rapport au planning à barres, mais permettent uniquement de minimiser le délai de réalisation dans des problèmes d'ordonnancement ne comportant que des contraintes potentielles.

Dès qu'il y a des contraintes cumulatives ou disjonctives, on est en présence du deuxième type de problème d'ordonnancement pour lequel les techniques précédentes sont défaillantes. Dès 1954, JOHNSON [15] a étudié ce type d'ordonnancement suivi par de nombreux autres chercheurs parallèlement au développement des techniques de Recherche Opérationnelle et de leurs applications dans des domaines qui étaient auparavant dominés par l'intuition et l'expérience.

Dans la réalité l'ordonnancement d'une entreprise doit tenir compte de ressources limitées et entre donc dans cette deuxième catégorie. En effet, lorsqu'un certain nombre d'opérations, non ordonnées entre elles par des contraintes de succession, doivent être réalisées à l'aide d'un même moyen leurs temps d'exécution sont nécessairement disjoints.

Dans cette thèse nous nous intéressons donc à un problème d'ordonnancement d'atelier qui, selon la terminologie consacrée, peut s'énoncer ainsi : un ensemble de machines doit exécuter un ensemble de tâches affectées d'un délai de réalisation. Chaque tâche est constituée d'opérations élémentaires ordonnées dont les durées sont connues. Il faut déterminer les dates de passage des opérations sur les machines de façon à respecter la charge des machines et si possible les délais fixés. C'est suivant CONWAY [1] une extension du "n/m/G/T job shop scheduling problem" : n tâches, n machines, un atelier à flot quelconque G (l'ordre de passage des opérations sur les machines n'étant pas unique pour toutes les tâches), le critère d'évaluation étant ici le respect des délais par la mesure du retard des tâches.

A notre connaissance, les modèles de résolution de ces problèmes d'ordonnancement à ressources limitées proposés jusqu'ici nécessitent tous au moins la première des hypothèses restrictives suivantes :

(1) le nombre de tâches est connu à l'avance.

(2) les machines sont toutes distinctes et continuellement disponibles (hypothèse non retenue [4])

(3) les opérations d'une même tâche sont liées par un ordre linéaire.

L'ordonnancement effectué est donc de type statique puisqu'aucune modification ne peut être introduite au programme de fabrication pendant la période de gestion. Le problème est alors formulé en termes de programmes linéaires en nombres mixtes ou de contraintes disjonctives dans des graphes. Il est le plus souvent résolu par des méthodes combinatoires telles que "Branch and Bound" [3] ou arbitrage sur les disjonctions dans des graphes. Toutefois, il faut citer la démarche de ERSCHLER, FONTAN, ROUBELLAT [20] qui ont élaboré un outil d'aide à la décision et celle de NEPOMIASCHY [4] qui utilise une méthode de pénalisation.

Les problèmes ainsi traités ne paraissent pas suffisamment représentatifs des problèmes d'ordonnancement concrets. L'inconvénient majeur de ces méthodes est la nécessité de connaître au même instant toutes les tâches. La production est ainsi fixée à l'avance, ce qui parait peu réaliste. C'est pourquoi nous introduisons ici une méthode d'ordonnancement dynamique qui pallie cet inconvénient majeur : l'ensemble des tâches peut évoluer dans le temps. Ce modèle permet donc de prendre en compte le cas d'un atelier fonctionnant avec une production "à la demande".

L'approche de tels problèmes concrets interdit l'utilisation des méthodes de type précédent. En effet toute adaptation de ces méthodes à une production "à la demande" parait irréalisable par son coût prohibitif dû au très grand nombre de variables à introduire. L'algorithme que nous proposons est donc nouveau dans sa conception et particulièrement efficace pour traiter des problèmes réels. Il permet de fournir, avec une relative rapidité, une solution qui, dans tous les cas, respecte la contrainte de charge maximale des machines. De plus cette solution est convenable du point de vue de la tenue des délais fixés.

Notre étude des problèmes d'ordonnancement d'atelier se décompose en quatre parties.

Dans le premier chapitre nous présentons le problème général d'ordonnancement à ressources limitées avec ses deux contextes : le contexte statique et le contexte dynamique. Nous étudions ensuite les objectifs de l'ordonnancement d'atelier ainsi que les divers critères permettant de mesurer si les objectifs sont atteints. Cette étude est conduite dans le but de déterminer les seuls critères pertinents dans le cas dynamique.

Le deuxième chapitre est consacré aux grandes classes de modèles proposés pour formuler le problème d'ordonnancement statique : les programmes linéaires en nombres mixtes et les graphes à arcs disjonctifs. Puis sont étudiées, en montrant leurs limites, les méthodes de résolution le plus souvent utilisées pour ce type de problème. Nous faisons plus particulièrement la distinction entre d'une part les méthodes qui peuvent fournir une solution optimale (P.L.M. et énumération implicite) et d'autre part celles qui donnent simplement une solution réalisable plus ou moins proche de l'optimum (heuristiques et énumération implicite) ou un ensemble de solutions admissibles (méthodes analytiques combinatoires). Nous précisons de plus les résultats obtenus dans des cas particuliers d'ordonnancement.

Notre modèle d'ordonnancement dynamique fait l'objet du troisième chapitre. Nous y présentons d'abord les hypothèses spécifiques de notre système :

- l'atelier est à flot quelconque. C'est à dire chaque tâche possède son propre ordre technologique et de plus, plusieurs machines différentes peuvent être nécessaires simultanément pour réaliser la même tâche.

- Les machines sont réparties en groupes. Chaque groupe est décrit en fonction du temps, ce qui permet de prendre en compte des modifications de composition pour des raisons telles que la maintenance.

- Les tâches arrivent de façon continue dans l'atelier avec une date de début au plus tôt et un délai de réalisation.

Puis nous exposons notre méthode en la justifiant. Un ordonnancement périodique des tâches connues est réalisé. Il respecte les limitations de charge des machines. En outre, l'assignation des dates de début aux opérations est conduite dans le double but de réduire les en-cours et de réduire les retards. La méthode se décompose en deux phases se répétant périodiquement : un chargement et une régulation. Le chargement des machines avec toutes les opérations connues utilise un système de priorités dynamiques suivant un algorithme heuristique inspiré de [5]. Les dates de début des opérations sont ainsi assignées de manière semi-définitive. La régulation de la charge des machines est réalisée par une méthode de pénalisation semblable à celle référencée en [4]. L'ordonnancement de la période immédiate est ainsi réalisée.

Dans le quatrième chapitre nous donnons l'implémentation de notre modèle sur ordinateur. Nous y testons notre méthode sur un certain nombre d'exemples d'ateliers. Pour montrer les résultats obtenus dans différentes situations, nous faisons varier la charge des machines. Nous commentons enfin les hypothèse introduites dans notre modèle en faisant ressortir celles qui sont vraiment restrictives et celles qui peuvent être levées moyennant des modifications mineures.

Dans la conclusion nous indiquons la limite essentielle de notre système : la méthode d'ordonnancement accorde la même importance à toutes les tâches et à toutes les machines. Toutefois, les temps de calcul étant relativement courts, notre méthode peut être utilisée en tant qu'outil d'aide à la décision en permettant des essais successifs avec des données éventuellement modifiées. Une solution admissible, proche de l'optimum, peut donc être atteinte en un temps raisonnable.

Il est clair enfin que cette solution sera beaucoup plus réaliste que les solutions très théoriques des divers modèles présentés à ce jour.

CHAPITRE I

PRÉSENTATION DES PROBLÈMES D'ORDONNANCEMENT

Ce premier chapitre est consacré à une présentation générale des problèmes d'ordonnancement. Selon la tradition, nous utilisons le vocabulaire propre aux ateliers de production en parlant de tâches et de machines.

Il est clair cependant que des problèmes tels que :

- Ordonnancement de malades sur des lits spécialisés dans un hôpital,
- Ordonnancement de classes d'élèves dans des salles du collège,
- Ordonnancement de voitures en réparation dans un atelier de mécanique,
- Ordonnancement de bateaux sur les quais d'un port,
- Ordonnancement de programmes sur un ordinateur,
- etc...,

répondent à ce modèle.

Dans une première partie, nous présentons le contexte de l'ordonnancement en introduisant les notions de tâche et de machine et les restrictions le plus souvent adoptées dans les modèles classiques. Nous montrons alors la nécessité de considérer des modèles dynamiques.

La seconde partie est consacrée à l'étude des objectifs possibles d'un ordonnancement et aux critères d'optimisation qui y sont associés.

Pour chacun de ces critères, nous étudions le cas d'un ordonnancement statique et le cas d'un ordonnancement dynamique. Nous soulignons enfin qu'un bon ordonnancement nécessite, le plus souvent, de considérer simultanément plusieurs critères. Les méthodes de résolution de ces modèles classiques seront étudiées plus précisément dans les chapitres suivants.

ches avec comme première obligation de s'assurer du respect de l'ordre technologique. De plus cet ordonnancement doit être conduit de manière à approcher au mieux la situation idéale suivante : des machines continuellement en marche, une main d'oeuvre juste suffisante pour assurer cette charge des machines, des montants d'en-cours à un niveau raisonnable et des délais respectés.

La plus grande partie de la recherche sur l'ordonnancement d'atelier est centrée sur le problème des séquences. Ce dernier consiste à déterminer l'ordre d'une série de tâches sur une série de machines. Ce problème est naturellement seulement une partie du problème de contrôle de production dans un système d'ateliers. En effet la détermination des séquences est précédé par l'étude de la tâche ; elle est suivi par une phase de contrôle qui consiste à réguler et ajuster la production pour se conformer aux plans. Ces trois éléments forment un tout : le contrôle de production.

Du fait du développement des techniques de Recherche Opérationnelle et de la prise de conscience des gains réalisables avec un "bon" ordonnancement, des modèles théoriques représentant des problèmes d'ordonnancement plus ou moins simplifiés ont été formulés. Les techniques généralement utilisées sont les suivantes :

Programmation en nombres entiers, Analyse combinatoire, Graphes, Heuristiques, Simulation. Nous reviendrons sur les principales difficultés de ces approches dans le chapitre II. En particulier nous insisterons sur l'effort de calcul qui augmente très vite avec la taille du problème dans le cas des deux premières approches.

I.1.2. Définition du problème

Nous définissons d'abord les notions de tâches et groupes de machines. Ensuite nous examinons les hypothèses restrictives couramment adoptées, ce qui nous permettra de distinguer différentes classes d'ordonnancement d'atelier.

I.1.2.1. Tâches et machines

On dispose de plusieurs groupes de machines et dans un groupe donné les machines sont identiques. Des tâches spécifiques doivent être réalisées par chacune de ces machines. Chaque tâche se compose d'une ou plusieurs unités : dans ce dernier cas, dans la terminologie de l'atelier, la tâche est appelée un "lot" et l'unité une "pièce". Une tâche demande une ou plusieurs opérations ; chacune de ces opérations est effectuée par une machine et une seule d'un groupe donné. Ces différentes opérations d'une même tâche doivent être effectuées dans un certain ordre qui est en général fixé par la contrainte technologique. A chaque opération est lié un temps d'opération qui peut se décomposer en temps de fabrication, temps de transfert d'une machine à la suivante, temps de réglage de la machine avant le passage de l'opération sur cette machine.

Le problème d'ordonnancement s'énonce alors ainsi : déterminer la séquence des tâches -donc des diverses opérations- sur chaque machine de telle sorte que cette séquence soit "optimale" vis-à-vis d'un ou plusieurs critères.

Les objectifs, ainsi que les critères permettant d'évaluer dans quelle mesure ils sont atteints, seront étudiés en détail dans le paragraphe 2 de ce chapitre.

Exemples

Dans le cas de problèmes réels d'ordonnancement autres que l'atelier de fabrication tels que :

- les malades sur des lits spécialisés dans un hôpital,
- les classes d'élèves dans les salles du collège,
- les voitures en réparation dans l'atelier de mécanique,
- les bateaux sur les quais d'un port,
- les programmes sur un ordinateur,

les tâches sont respectivement les malades, les classes d'élèves, les voitures, les bateaux, les programmes, tandis que les machines représentent les lits, les salles, les ouvriers et les outils, les quais, les ressources de l'ordinateur.

I.1.2.2. Restrictions sur les tâches et les machines

Dans la plupart des études de problèmes d'ordonnancement d'atelier, des hypothèses restrictives sont souvent introduites (cf par exemple [1], [2], [3]). Les principales restrictions adoptées sont les suivantes :

H1) Le nombre de tâches et le nombre de machines sont connus et fixes.

H2) Toute machine est continuellement disponible ; on suppose qu'il n'y a pas de maintenance ni de panne.

H3) Les tâches sont des séquences d'opérations sans sous-ensembles. Pour une opération donnée $O_{i,s}$, opération de la tâche i passant sur la machine s, il y a au plus une opération $O_{i,r}$ telle que $O_{i,r}$ précède immédiatement $O_{i,s}$, (nous noterons $O_{i,r} < O_{i,s}$), et au plus une opération $O_{i,t}$ telle que $O_{i,s}$ précède immédiatement $O_{i,t}$ ($O_{i,s} < O_{i,t}$).

H4) Toute opération ne peut être réalisée que par une seule machine de l'atelier.

H5) Il n'y a qu'une seule machine de chaque type dans l'atelier.

H6) Une opération commencée doit être achevée avant d'en entreprendre une autre sur la même machine.

H7) Les temps de fabrication ne se recouvrent pas pour une même tâche, sauf si la tâche est un lot de pièces identiques.

H8) Toute machine ne peut faire qu'une seule opération à la fois.

Parmi ces hypothèses, nous verrons, dans les chapitres II et III, celles qui peuvent être levées afin de rendre le problème plus réaliste.

I.1.2.3. Différents types de problèmes d'ordonnancement

Dans les systèmes réels, on peut rencontrer une grande variété de situations qui vont évidemment conditionner le modèle d'ordonnancement à chaque cas. Les particularités à prendre en compte sont essentiellement l'ordre technologique des tâches, la connaissance ou non du nombre de tâches, et des données sur les tâches et le nombre de machines fixe ou variable.

La particularité sur l'ordre technologique des tâches conduit aux deux types d'atelier extrêmes : Atelier à flot unique et Atelier à flot quelconque. Dans le premier cas l'ordre de passage des opérations sur les différentes machines est le même pour toutes les tâches, tandis que dans le second cas chaque tâche possède son propre ordre technologique. L'atelier à flot unique peut bien sûr être considéré comme cas particulier de l'atelier à flot quelconque. Il est cependant plus simple de le traiter par des modèles spécifiques.

La distinction entre problèmes déterministes et problèmes stochastiques est due à la connaissance des données sur les tâches et les machines. Pour les problèmes déterministes, les données sont fixes, de sorte que les résultats de l'ordonnancement seront appliqués sans modification de nature aléatoire. Pour les problèmes stochastiques certains éléments ne sont connus qu'en terme de probabilité. Cela peut être le cas, par exemple, pour des caractéristiques des tâches du point de vue arrivée, délais, temps de fabrication ou des caractéristiques des machines du point de vue capacité de charge en raison de pannes éventuelles.

La dernière particularité, concernant le nombre de tâches à ordonnancer, paraît être la plus importante ; elle permet en effet de distinguer deux grandes classes de problèmes d'ordonnancement : les problèmes statiques et les problèmes dynamiques.

Les problèmes statiques : c'est le cas où les tâches en nombre n fini et fixé arrivent simultanément dans l'atelier. Aucune autre tâche n'arrivera tant que l'atelier n'aura pas fini d'exécuter ces n tâches. S'ils sont intéressants, ces problèmes statiques n'en sont pas moins généralement peu réalistes. Mais ils peuvent a priori permettre de traiter le cas dynamique en tant que série de cas statiques. Nous les étudierons dans le chapitre II.

Les problèmes dynamiques : les tâches arrivent de façon aléatoire. Leurs dates d'arrivée dans l'atelier ainsi que leur nombre ne sont connus à l'avance qu'en termes de probabilités.

Nous verrons dans les chapitres suivants que des méthodes entièrement différentes sont requises pour aborder ces deux types de problèmes. C'est pourquoi nous serons amenés à tenir compte, dès l'étude des critères du paragraphe suivant, de cette particularité essentielle qu'est le mode d'arrivée des tâches dans l'atelier.

I.2. OBJECTIFS DE L'ORDONNANCEMENT ET ETUDE DES CRITERES

L'évaluation de la solution du problème d'ordonnancement exige le choix d'un critère ou d'un ensemble de critères permettant d'évaluer dans quelle mesure sont atteints les objectifs fixés à l'avance.

Nous étudions dans cette section, de manière exhaustive, les divers critères proposés dans les différentes études d'ordonnancement ([1], [2], [3], [4],...) ainsi que ceux utilisables dans les problèmes réels.

Nous discutons ensuite les équivalences et interrelations qui existent entre ces divers critères. Pour chacun d'entre eux nous examinons le problème statique et le problème dynamique.

Notations utilisées dans la suite de ce travail :

Les lettres majuscules représenteront les variables qui décriront la solution du problème d'ordonnancement. Les lettres minuscules représenteront les quantités définissant le problème d'ordonnancement.

Les notations et les définitions des termes de base seront introduites au fur et à mesure des besoins et récapitulées dans l'annexe A.

I.2.1. Objectifs et classement des critères

Le double objectif d'un ordonnancement doit être l'efficacité de l'atelier de production et la satisfaction des clients.

Les mesures de performance permettant d'évaluer si le double objectif est atteint peuvent prendre différentes formes. Dans les ateliers réels nous devons considérer principalement (du strict point de vue "ordonnancement") :

- l'utilisation des moyens de production,
- le respect des délais,
- les temps morts aussi bien pour les machines que pour la main d'oeuvre,
- le montant des en-cours de fabrication, des stocks de matières premières et de produits finis,
- les pertes de temps en reconversion des machines,
- le temps total réel de fabrication de chaque tâche,
- les transferts des tâches dans l'atelier,
- les effets des pannes des machines ou plus généralement les aléas de fabrication.

Il est bien sûr illusoire de chercher à optimiser toutes ces mesures à la fois. Par contre il est possible de combiner ces critères pour établir une fonction globale de coût. Personne n'a, à notre connaissance, essayer de développer un tel modèle d'ordonnancement. La raison essentielle la plus vraissemblable est que l'on se heurte au problème de la connaissance approchée de certains coûts. L'imprécision des données ne justifie plus alors les efforts d'optimisation.

Plusieurs classifications des critères ont été proposées dans diverses études ([1], [25], [28],...). Nous citerons les deux principales : celle qui distingue ([1]) les critères relatifs aux tâches et les critères relatifs aux machines et celle qui sépare ([25]) les critères basés sur la date d'achèvement d'un ensemble de tâches et les critères basés sur les délais de chaque tâche.

1. La classification de CONWAY [1]

Les critères relatifs aux tâches sont, par exemple, le temps minimum de passage des tâches dans l'atelier, le minimum de retard des tâches ou le minimum d'en-cours de fabrication.

Parmi les critères relatifs aux machines, les plus fréquemment considérés sont l'utilisation maximale de l'atelier et le temps minimum d'adaptation des machines.

Cette distinction n'a plus de sens lorsqu'on parle de coûts au lieu de temps.

2. La classification de GERE [25]

Les critères basés sur la date d'achèvement d'un ensemble de tâches comprennent, par exemple, la durée minimum de passage dans l'atelier et l'utilisation maximale des machines.

Les critères basés sur le délai de chaque tâche sont, par exemple, le critère du minimum de tâches en retard et celui du temps total minimum de retard.

Cette classification revient à distinguer les ordonnancements de type statique et ceux de type dynamique.

Aux classements qui précèdent, nous préfèrerons personnellement les trois classifications suivantes basées sur différentes caractéristiques importantes telles que les notions de coût, de pondération et de multicritère.

1. Les critères de temps et les critères de coût

Pour le temps nous citerons : le temps minimum d'ordonnancement, le temps minimum de passage des tâches dans l'atelier et le retard total minimum.

Pour le coût, on peut choisir pour critère : le coût minimum des retards (pénalités), le coût minimum d'attente des tâches dans l'atelier (en-cours) ou le coût d'immobilisation minimum des machines par inactivité.

2. Les critères pondérés et les critères non pondérés

Il existe des cas où une tâche à priorité sur une autre (cela peut être par exemple en liaison avec son coût). On associe alors aux différentes tâches une valeur pondérée fonction de leurs importances relatives. On peut agir de même avec les machines. Nous considèrerons par exemple (cf paragraphe I.2.2.) la somme minimale pondérée des retards des tâches et la somme minimale pondérée des taux d'utilisation des machines. Si toutes les tâches ou toutes les machines sont estimées d'égale importance, alors on utilisera des critères non pondérés tels que ceux énoncés précédemment.

3. Les critères simples et les critères multiples

Un critère simple est un critère à une seule composante. Dans la pratique, il ne pourra satisfaire à lui tout seul l'exigence du double objectif de l'efficacité de l'atelier et de la satisfaction des clients. Pour un système opérationnel, il sera donc nécessaire "d'optimiser" un critère multiple c'est-à-dire un critère qui sera une combinaison de plusieurs critères simples en général antagonistes.

Après avoir étudié séparément les divers critères simples ainsi que les relations éventuelles entre eux, nous reviendrons plus précisément dans le paragraphe I.2.3. sur les problèmes posés par les critères multiples.

I.2.2. Les critères simples

Dans ce paragraphe, nous étudions en détail les critères simples relatifs à la date d'achèvement des tâches, au temps total réel de passage des tâches dans l'atelier, aux délais, aux en-cours de fabrication, à l'utilisation des machines et aux temps d'adaptation des machines. Pour chacun de ces critères, nous distinguerons le cas d'un problème d'ordonnancement statique et le cas d'un ordonnancement dynamique.

I.2.2.1. Critère de la date d'achèvement

L'optimisation de ce critère apparaît dans de nombreuses études d'ordonnancement (cf par exemple [6], [24], [27], [32]). C'est en effet le plus simple à mettre en oeuvre.

Soit D_i la date d'achèvement réelle de la tâche i, c_i sa date de début au plus tôt, f_i sa durée de fabrication et W_i son temps d'attente ; alors

$$D_i = c_i + f_i + W_i \qquad (1)$$

c_i et f_i étant des données, D_i ne dépend que de W_i.

Cas d'un problème statique :

Soit D la date d'achèvement d'une série de n tâches qui ont toutes c comme date de début au plus tôt. Alors D, la date de fin d'ordonnancement est donnée par

$$D = \sup_i (D_i) = c + \sup_i (f_i + W_i) \qquad (2)$$

La séquence optimale vis-à-vis de D est donc celle où le temps total d'attente de toutes les tâches est minimum. En particulier, plus la tâche sera longue (i.e. plus f_i sera grand), moins cette tâche devra attendre dans l'atelier (i.e. W_i devra être le plus cours possible).

Soit $\overline{D}$ la moyenne de la date d'achèvement pour les n tâches

$$\overline{D} = c + \overline{f} + \overline{W} \qquad (3)$$

$\overline{f}$ et $\overline{W}$ étant respectivement les moyennes des f_i et des W_i. c et $\overline{f}$ étant des constantes, minimiser $\overline{D}$ équivaut à minimiser $\overline{W}$.

Deux séquences ayant même date d'achèvement D n'ont pas nécessairement la même moyenne $\overline{D}$, comme le montre l'exemple suivant :

<u>Exemple avec 3 tâches et 2 machines différentes</u> :

Si nous notons $O_{i,j}$ l'opération de la tâche i passant sur la machine j, l'ordre technologique est donné par

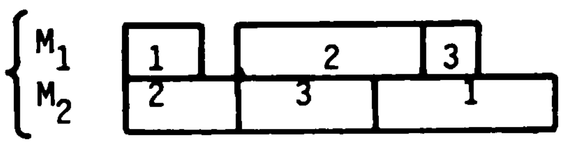
$$O_{1,1} < O_{1,2},\ O_{2,2} < O_{2,1},\ O_{3,2} < O_{3,1}$$

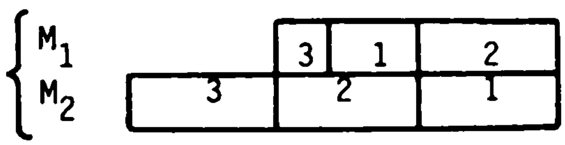
Les durées des opérations sont données par la matrice

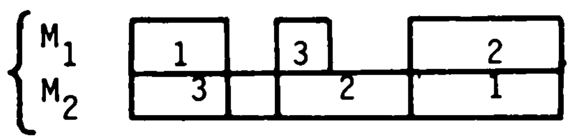
$$\begin{bmatrix} 2 & 4 \\ 4 & 3 \\ 1 & 3 \end{bmatrix}$$

où la ligne i concerne la tâche i et la colonne j détermine la machine j.

La valeur optimale de D, notée D^*, est $D^* = 10$ et 3 séquences fournissent cette valeur. Ce sont

S_1  $W = 8$

S_2  $W = 7$

S_3  $W = 7$

Dans le cas où il n'y a que 2 machines il existe un algorithme de Johnson et Jackson ([15], [16]) qui permet de trouver au moins une séquence optimale vis-à-vis de D. Cet algorithme donnerait ici la séquence S_1. Or si l'on considère la somme W des temps d'attente, S_1 est moins bonne que les deux autres. En raisonnant sur les moyennes $\bar{D}$ nous retrouvons ce résultat

Séquences	S_1	S_2	S_3
$\bar{D}$	8,33	8	8

$\bar{D}^* = 8$

S_2 et S_3 sont optimales vis-à-vis de $\bar{D}$.

Cas d'un problème dynamique :

Le critère de la date D de fin d'ordonnancement n'est pas du tout approprié lorsque les tâches arrivent de façon continue dans l'atelier. Dans l'égalité $D_i = c_i + f_i + W_i$ les c_i sont différents pour toute tâche i. Il est donc préférable de considérer la différence $D_i - c_i$ c'est-à-dire le temps de passage P_i dans l'atelier de la tâche i.

I.2.2.2. Critère du temps total de passage dans l'atelier

Pour la tâche i nous noterons P_i le temps total réel de passage dans l'atelier. Il peut être évalué ainsi

$$P_i = D_i - c_i \qquad (4)$$

$$P_i = f_i + W_i \qquad (5)$$

avec D_i la date d'achèvement réelle, c_i la date de début au plus tôt, f_i le temps de fabrication et W_i le temps total d'attente de la tâche i.

En moyenne on considèrera :

$$\overline{P} = \overline{D} - \overline{c} = \overline{f} + \overline{W} \qquad (6)$$

Cas d'un problème statique :

Nous pouvons, sans perte de généralité, prendre les dates de début au plus tôt des tâches (qui sont communes) comme origine et égale à 0. D'où $\overline{P} = \overline{D}$. Tout ce qui a été dit sur D reste vrai pour P.

Cas d'un problème dynamique :

D'après les formules (5) et (6), nous déduisons que le temps de passage varie dans le même sens que le temps d'attente. Lorsque nous étudierons le critère des en-cours nous verrons que, pour une période donnée, la moyenne $\overline{P}$ augmente avec l'accroissement du nombre de tâches dans l'atelier.

Remarque :

Si on se place du point de vue en-cours de fabrication, on peut considérer que tant que la première opération de la tâche i n'est pas commencée l'en-cours correspondant est nul. Cela revient à diminuer le temps total d'attente W_i de la durée d'attente de la lère opération de la tâche i. Pour cette raison il est alors plus logique de considérer, dans le cas dynamique, la différence D_i - C_i comme temps réel de passage où C_i désigne la date de début effective de la 1ère opération de la tâche i.

Cependant dans la suite de ce chapitre et dans le chapitre II le temps de passage P_i sera toujours prix égal à D_i - c_i.

I.2.2.3. Critère des délais

La date de fin au plus tard est assignée à chaque tâche en général par "négociations" entre le client et l'entreprise. Il est bien évident qu'en réalité elle dépend beaucoup de la charge de l'atelier et donc de l'ordonnancement. Cependant nous considèrerons ces dates comme fixées préalablement à l'ordonnancement.

Le respect des délais attribués aux tâches nous paraît être un des buts primordial de l'ordonnancement. En effet des pénalités à payer au client pour retard sur la date contractuelle peuvent avoir été prévues. Cela peut aussi entraîner des frais indirects, tels expéditions spéciales, pouvant aller jusqu'à la perte future du client. La gestion de l'atelier doit donc se préoccuper de minimiser ces coûts de pénalité à défaut de pouvoir les éliminer complètement. Ceci ne pourra se faire qu'aux dépens d'autres facteurs tels que l'augmentation des en-cours ou la diminution de l'utilisation de l'atelier.

Le respect du délai pour la tâche i se mesure de trois façons : l'écart, le retard et l'avance.

L'écart c'est-à-dire la non-coïncidence entre la date d'achèvement réelle et celle promise notée d_i pour la tâche i

$$\text{soit} \quad E_i = D_i - d_i$$

Le retard de la tâche i correspond à un écart positif

$$(E_i \geq 0) \qquad R_i = \sup(0, E_i)$$

L'avance de la tâche i correspond à un écart négatif

$$(E_i \leq 0) \qquad A_i = \sup(0, -E_i)$$

Nous allons étudier chaque possibilité en essayant de déterminer l'utilité propre de chacune de ces mesures.

1. Ecart E_i

Le temps alloué à la tâche i dans l'atelier étant $p_i = d_i - c_i$ et le temps réel de passage étant $P_i = D_i - c_i$ alors.

$$E_i = D_i - d_i = P_i - p_i$$

L'écart moyen $\overline{E}$ est donc

$$\overline{E} = \overline{P} - \overline{p} \qquad (7)$$

ou encore

$$\overline{E} = \overline{F} + \overline{W} - \overline{p} \qquad (8)$$

Distinguons maintenant les deux cas d'ordonnancement.

Cas d'un problème statique :

Nous avons

$$\overline{E} = \overline{D} - \overline{p} - \overline{c} \qquad (7')$$

(7') et (8) impliquent alors le résultat suivant :

Une séquence qui est optimale vis-à-vis de $\overline{E}$ est optimale pour $\overline{D}$, $\overline{P}$ et $\overline{W}$.

Cas d'un problème dynamique :

$\overline{D}$ n'ayant pas de sens ici, seules subsistent les relations (7) et (8). Celles-ci impliquent le résultat suivant :

Une séquence qui est optimale vis-à-vis de $\overline{E}$ l'est aussi vis-à-vis de $\overline{P}$ et $\overline{W}$.

Remarque générale sur le critère $\overline{E}$:

L'écart moyen $\overline{E}$ donne une indication sur le respect des délais mais il n'est pas vraiment satisfaisant. En effet si $\overline{E} = 0$ pour une séquence, cela peut résulter du fait que certaines tâches se terminant avec beaucoup d'avance compensent d'autres tâches qui finissent avec beaucoup de retard. Pour cette raison $\overline{E}$ est insuffisant pour mesurer le respect des délais. Il serait donc nécessaire d'utiliser l'écart-type. En fait les deux critères suivant nous permettent de mieux contrôler les écarts.

2. Retard R_i

$$R_i = \sup (0, E_i)$$

Ce critère permet de vérifier que la tâche i se termine à temps ou dès que possible après la date de fin au plus tard. La moyenne des retards $\overline{R}$ est particulièrement importante. Elle apparaît comme une fonction de pénalisation à minimiser. Une pondération adéquate permettrait de tenir compte de cas où les retards des tâches ne sont pas d'égale importance, les poids étant proportionnels aux coûts de pénalités.

3. Avance A_i

$$A_i = \sup (0, -E_i)$$

Ce critère n'a pas d'intérêt en tant que mesure du respect des délais. Il peut être rattaché à la notion de mesure des en-cours. En effet, dans certains cas, les tâches en avance peuvent donner lieu à des pénalités dues aux coûts de stockage ou, dans le cas dynamique, à leur incidence sur les ordonnancements futurs.

De même que pour les retards il peut être intéressant de minimiser la moyenne $\bar{A}$, ou mieux, la somme pondérée des avances.

<u>Remarque sur ce critère des délais</u> :

Pour toute tâche i on a évidemment $E_i = R_i - A_i$. La relation avec les moyennes $\bar{E} = \bar{R} - \bar{A}$ ne présente aucun intérêt comme nous l'avons signalé plus haut puisqu'elle n'est vraie que dans les deux cas extrêmes : toutes les tâches sont en retard ou toutes les tâches sont en avance.

I.2.2.4. Critère des en-cours

Dans l'atelier, une file d'attente des tâches se crée devant chaque machine. Ceci implique un montant d'en-cours de fabrication qui peut être mesuré par les variables suivantes (ou à partir de celles-ci) :

1) N, le nombre total de tâches dans l'atelier pour une période $[\theta_0,\theta]$ donnée, qui comprend le nombre de tâches en cours de fabrication N_e et le nombre de tâches en attente N_a.

2) F, le travail total de l'atelier sur la même période, se décomposant en F_e la somme des temps de fabrication des opérations en cours, F_t la somme des temps de fabrication des opérations terminées et F_a la somme des temps de fabrication des opérations restant à faire.

3) Γ_e le coût des en-cours.

1. Nombre moyen de tâches dans l'atelier

Nous le noterons $\overline{N}$. Sur la période $[\theta_0, \theta]$, $\overline{N}$ est donné par :

$$\overline{N} = \frac{1}{\theta - \theta_0} \sum_{i=1}^{N} P_i = \frac{N\overline{P}}{\theta - \theta_0}$$

où P_i est le temps total réel de passage de la tâche i.

Or N devient infini avec θ et $\lim\limits_{N \to +\infty} \frac{\theta_0}{N} = 0$. Le processus devient donc continu, et $\lim\limits_{N \to +\infty} \frac{\theta}{N} = \frac{1}{\lambda}$ (où $\frac{1}{\lambda}$ est le temps moyen séparant deux arrivées de tâches) et par conséquent $\overline{N} = \lambda\overline{P}$, avec λ taux d'arrivée des tâches dans l'atelier. Pour plus de précisions se reporter à Little [7] qui donne la démonstration générale dans le cadre des files d'attente.

Les figures 1, 2 et 3 donnent une illustration graphique de la formule (9).

Donc si on se limite à mesurer les en-cours par le nombre moyen de tâches dans l'atelier $\overline{N}$, minimiser les en-cours est équivalent à minimiser le temps moyen de passage $\overline{P}$ des tâches dans l'atelier.

Ce résultat est acquis aussi bien dans le cas d'un ordonnancement statique que dynamique.

Remarque :

On prouverait, de même, que le nombre moyen de tâches en attente $\overline{N}_a$ est donné par $\overline{N}_a = \overline{W}$, où $\overline{W}$ est le temps moyen d'attente.

2. Travail total dans l'atelier

Le travail total de l'atelier, noté F, sur la période

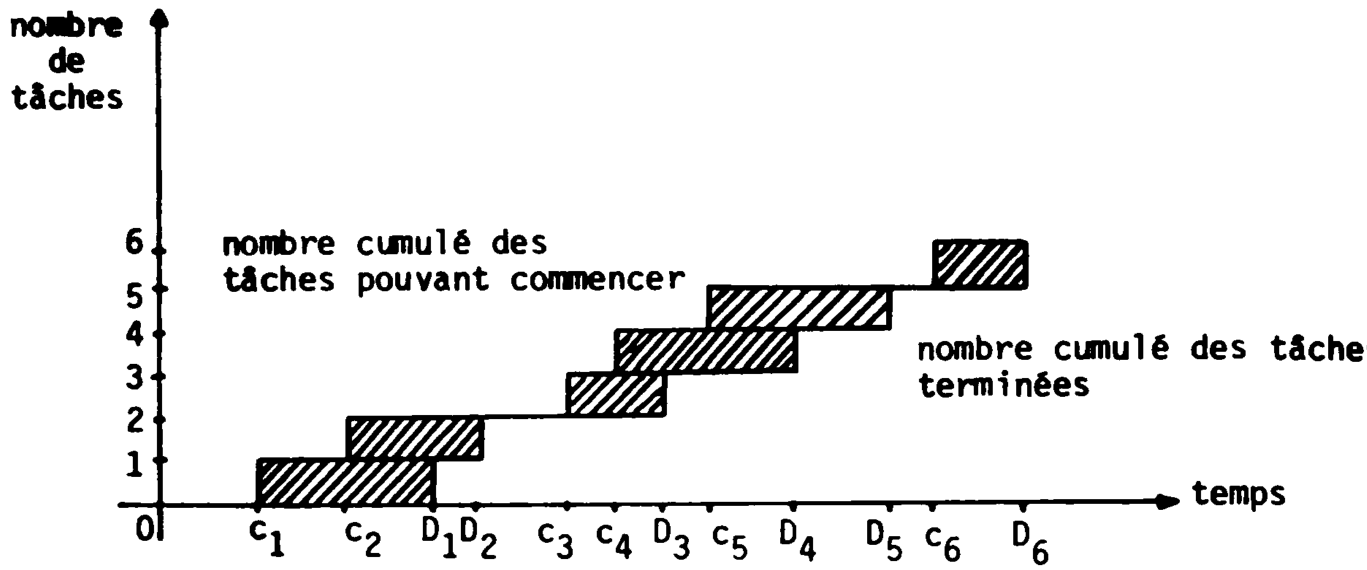


Figure 1

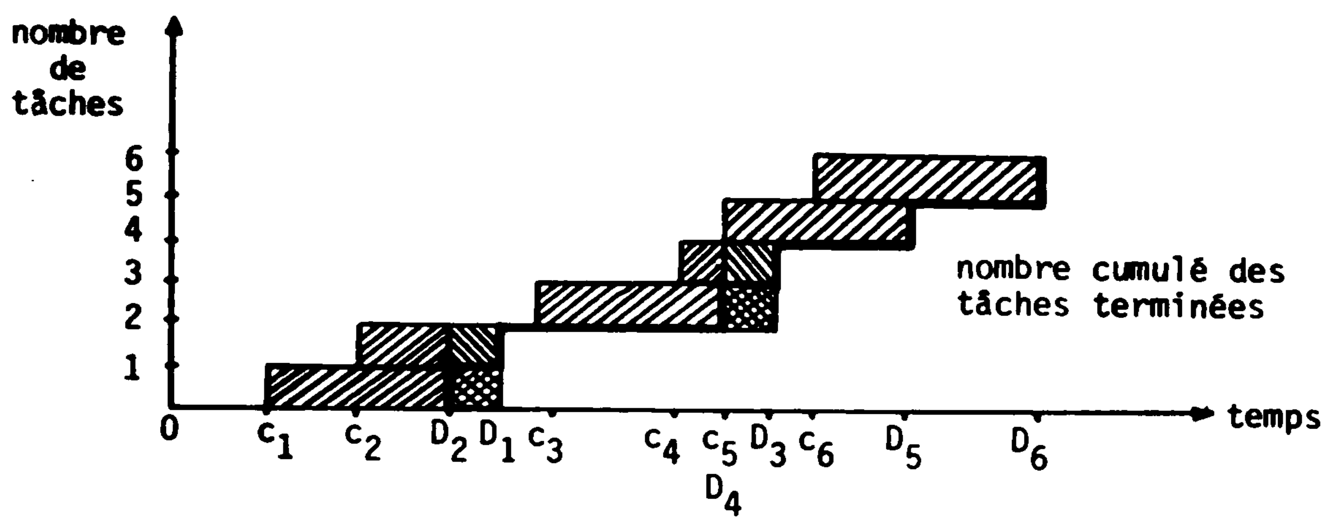


Figure 2

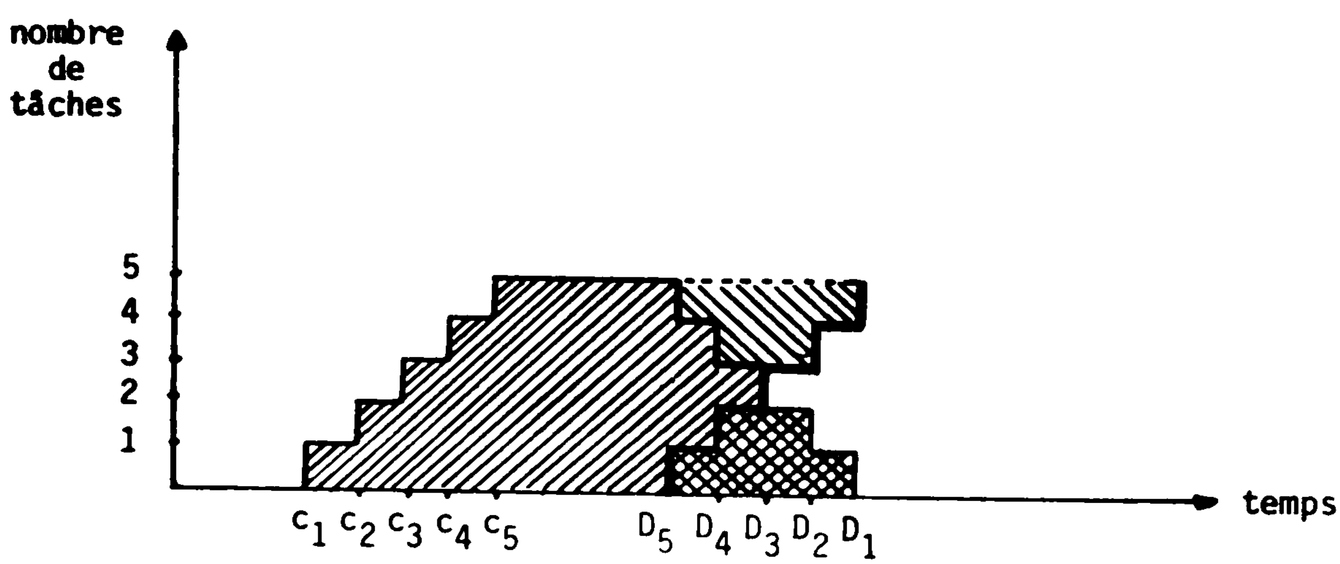


Figure 3

(Les aires sont conservées)

d'ordonnancement est la somme des temps de fabrication de toutes les opérations de toutes les tâches. Il n'est adapté qu'au cas statique puisque la notion de délai n'intervient pas. Dans ce cas, la somme des temps de fabrication des opérations terminées, F_t, peut être une estimation du montant des en-cours de fabrication. Une mesure appropriée de la charge de l'atelier est alors la somme des temps de fabrication F_a des opérations restant à faire.

Remarque :

Il est intuitif que le nombre moyen de tâches en cours $\bar{N}_e$ ainsi que la moyenne des temps de fabrication des opérations en-cours $\bar{F}_e$ deviennent indépendant de l'ordonnancement lorsque la longueur de sa période augmente.

En effet elles permettent de mesurer l'utilisation de l'atelier qui, on le verra, est sous certaines conditions une constante de l'ordonnancement.

3. Le coût des en-cours

Γ_e, coût des en-cours, peut être estimé en fonction du degré d'achèvement des tâches. Celui-ci sera fonction de l'attente des tâches entre les diverses opérations ainsi que de l'avance A définie précédemment. Ce critère doit en effet être relié au fait qu'il existe des délais pour chaque tâche. Il faut aussi noter qu'en général toutes les tâches ne sont pas d'égale importance. Pour la tâche i il faudra minimiser cette valeur instantanée représentant le coût de l'en-cours. Ceci équivaudra à fabriquer la tâche i le plus tard possible tout en respectant son délai.

I.2.2.5. Critère de l'utilisation

Une des conséquences de l'ordonnancement peut être l'inactivité de certaines machines pendant certaines périodes de temps. Il en résulte une sous-utilisation de la capacité de production.

Soit $I_{i,j,k}$ la durée d'inactivité de la machine j avant d'effectuer l'opération $O_{i,j,k}$ (i désigne le numéro de la tâche concernée, k permet de prendre en compte des tâches non linéaires ; cf paragraphe I.1.2.2. et paragraphe III.1.2.2.). Pour une période d'ordonnancement donnée, le temps total d'inactivité de la machine j sera $I_j = \sum_i \sum_k I_{i,j,k}$, et pour l'ensemble des M machines de l'atelier l'inactivité totale sera $I = \sum_{j=1}^{M} I_j$.

Minimiser l'inactivité des machines équivaut à maximiser leur utilisation. C'est la mesure du taux d'utilisation U de l'atelier que nous prendrons comme critère dans ce cas.

Le taux d'utilisation U sera le rapport entre la somme des temps de fabrication et la capacité totale de production disponible pendant la période donnée.

Remarque :

Nous ne tenons pas compte du fait qu'il peut y avoir sous-utilisation des machines par manque de tâches à effectuer. Cela suppose donc implicitement qu'il existe un accord global entre les tâches à effectuer et la configuration de l'atelier.

Distinguons les deux types d'ordonnancement.

Cas d'un problème statique :

Si n est le nombre de tâches dans l'atelier sur l'intervalle [O,D] le taux d'utilisation U est

$$U = (\sum_{i=1}^{n} f_i)/D$$

Alors le taux moyen d'utilisation sera

$$\overline{U} = n\overline{f}/MD$$

où M est le nombre de machines de l'atelier.

Cas d'un problème dynamique :

Supposons le nombre de machines dans le groupe j donné par $M(j,\theta)$ en fonction du temps θ. Alors le taux moyen d'utilisation des machines est défini par

$$\overline{U} = \frac{\lambda \overline{f}}{\overline{m}} \tag{10}$$

où $\overline{m}$, le nombre moyen total de machines disponibles est donné par

$$\overline{m} = \lim_{t \to +\infty} \frac{1}{t} \left(\sum_{j=1}^{m} \sum_{\theta=0}^{t} M(j,\theta) \right)$$

d'où d'après (6), (9) et (10) les relations

$$\overline{P} = \overline{W} + \overline{f} = \frac{\overline{N}\,\overline{f}}{\overline{m}\,\overline{U}} \tag{11}$$

Le taux moyen d'utilisation de l'atelier est donc une donnée du problème. Il dépend essentiellement du taux d'arrivée des tâches. En toute rigueur cependant, le taux d'utilisation $\overline{U}$ dépend de la méthode d'ordonnancement. C'est pourquoi nous sommes obligés d'introduire la notion de compatibilité entre les charges à absorber et les moyens de production.

Si on s'autorise à augmenter les $M(j,\theta)$, c'est-à-dire si l'on a recours à la sous-traitance ou à des heures supplémentaires, alors $\overline{U}$ devient une variable du problème puisque $\overline{m}$ n'est plus constant.

I.2.2.6. Critère d'adaptation des machines

Ce critère est totalement indépendant des précédents en ce sens qu'il intervient dans des ateliers très spécifiques. Il doit être pris en compte dans les problèmes d'ordonnancement où un réglage ou une transformation d'une machine est nécessaire entre deux opérations consécutives sur cette machine.

En général ce temps de réglage dépend, pour une machine j donnée, de l'opération $O_{i,j,k}$ se terminant et de l'opération $O_{\ell,j,h}$ qui va commencer. Nous le noterons $t(O_{i,j,k},O_{\ell,j,h})$.

Ces temps de réglage immobilisent la machine et entraînent donc un coût. Il faut donc les minimiser lors de l'ordonnancement spécialement dans les cas où ces temps varient beaucoup en fonction de la séquence choisie : c'est-à-dire si $t(O_{i,j,k},O_{\ell,j,h})$ est très différent de $t(O_{\ell,j,h},O_{i,j,k})$. Cela peut se faire en prenant comme critère le taux d'utilisation des machines étudié dans le paragraphe précédent.

I.2.3. Etude des multi-critères

En pratique, il sera très rarement construit un ordonnancement en fonction d'un seul critère. Dans la plupart des situations réelles il est nécessaire de trouver une séquence qui réalise un bon compromis entre les ordonnancements fournis en fonction de plusieurs critères simples pris séparément. Ceci est difficile car si certains critères simples sont parfois compatibles, d'autres sont souvent antogonistes.

Comme nous l'avons vu, les objectifs de l'ordonnancement sont l'efficacité de l'atelier (obtenue par une bonne utilisation des machines et par un montant d'en-cours réduit) et la satisfaction des clients par le respect des délais. Dans un système opérationnel, il sera donc nécessaire de considérer simultanément les critères pertinents qui rendent le mieux compte du respect de ce double objectif.

Parmi les mesures étudiées, celles permettant d'évaluer l'efficacité de l'atelier seront :

- pour les en-cours, l'avance moyenne $\bar{A}$ des opérations ;
- pour l'utilisation des machines, le taux moyen d'utilisation $\bar{U}$.

La satisfaction des clients sera mesurée par le retard moyen des tâches $\bar{R}$.

Un multi-critère réaliste sera donc une combinaison de ces trois critères simples. La mise en oeuvre de l'évaluation de ce multi-critère reste à définir. Il s'agit de réaliser "l'optimisation" de l'ordonnancement en fonction de ce critère à trois composantes.

Pratiquement trois approches de mise en oeuvre sont réalisables : l'évaluation composite, l'évaluation séquentielle et l'évaluation sous contraintes.

1. L'évaluation composite

Elle consiste à pondérer chaque critère simple et par regroupement à former un "grand critère" donnant par exemple un coût. Le seul problème délicat de mise en oeuvre est la détermination des différents poids à affecter aux critères simples.

2. L'évaluation séquentielle

On définit un ordre sur les critères simples. Ceux-ci sont alors envisagés séquentiellement par ordre d'importance décroissante. Chaque critère permet de réduire le sous-ensemble des solutions optimales vis-à-vis du précédent. L'algorithme s'arrête lorsqu'il n'y a plus qu'une seule solution.

3. L'évaluation sous contraintes

Elle est semblable à la précédente en ce sens qu'il faut choisir un premier critère simple. Si l'on raisonne en termes de programme linéaire, ce critère sera la fonction à optimiser. Les autres critères simples sont introduits sous forme de contraintes supplémentaires sur l'espace des solutions. Cette méthode est intéressante dans le cas où les critères simples sont duaux c'est-à-dire si la maximisation de l'un entraîne la minimisation de l'autre.

I.2.4. Temps et coûts

Nous avons introduit pour chaque critère des notions de moyennes. Celles-ci sont d'une grande importance dans les applications pratiques. Il faut cependant noter que deux ordonnancements distincts donnant, pour un critère donnée, une même valeur moyenne ne sont pas forcément équivalents (voir par exemple le paragraphe I.2.2.1.).

Les critères simples n'ont été étudiés que par rapport au temps. Nous n'avons considéré que des critères simples non pondérés. Il est possible cependant d'introduire des coûts pour différencier les importances relatives des tâches entre elles ou des machines entre elles. Cela revient à en faire des critères simples pondérés où les poids sont des fonctions de coûts.

Tous les résultats acquis en raisonnant sur les temps restent valables si l'on introduit une pondération par les coûts et, en particulier, les conclusions sur les moyennes sont identiques.

Chapitre II

Problèmes d'Ordonnancement Statiques : Modèles et Méthodes de Résolution.

Nous présentons, dans ce chapitre, les modèles les plus caractéristiques qui ont été proposés pour formuler le problème d'ordonnancement statique d'un atelier. Nous étudions ensuite les méthodes de résolution le plus souvent utilisées pour ce type de problème.

Cette étude est aussi complète que possible mais ne prétend pas être exhaustive. Nous nous intéressons surtout aux grandes familles de formulation et de résolution avec les restrictions généralement introduites que nous avons rappelées au chapitre I. Nous rappelons enfin les résultats obtenus dans certains cas bien particuliers d'ordonnancement [9] [15] [16] [23] [24].

La notation que nous utilisons dans la suite est celle de CONWAY [1]. Elle permet de définir simplement le type de problème envisagé par un quadruplet A/B/C/D, où A caractérise les tâches, B définit les machines, C est le type de l'atelier et D l'un des critères d'évaluation de l'ordonnancement étudiés dans le chapitre précédent.

Nombre de machines

Sauf exception que nous signalons, la majorité des études concernent des problèmes où il n'y a pas de machines identiques, c'est-à-dire des problèmes sans groupes de machines. Dans ce cas B est noté m : atelier simple à m machines. Pour un atelier multiple, où il y a m groupes de machines, B sera noté M.

Type de l'atelier

Numérotons les groupes de machines de 1 à m et désignons par $O_{i,j}$ l'opération de la tâche i nécessitant la machine j. L'atelier est de type à flot unique (Flow-shop) et nous noterons F si la propriété suivante est vraie :

$$\text{pour toute tâche } i \qquad O_{i,j} < O_{i,\ell} \text{ si et seulement si } j < \ell$$

Dans le cas général où les opérations des diverses tâches passent dans un ordre quelconque sur les machines, le type de l'atelier est dit à flot quelconque et noté G.

Dans le cas d'un atelier à une machine le 3° élément du quadruplet n'a pas de sens. Nous le noterons A/1//D.

Critère

Lorsque le critère d'évaluation de l'ordonnancement peut être quelconque nous prenons comme notation le quadruplet A/B/C/V.

Exemple : n/m/G/P pour un problème à n tâches, m machines, un atelier à flot quelconque et le critère du temps de passage maximum des tâches.

II.1. FORMULATIONS DES PROBLEMES D'ORDONNANCEMENT

Nous nous bornons à indiquer les deux formulations les plus couramment utilisées. La première conduit à un programme linéaire en nombres mixtes (par exemple [17]). La deuxième s'appuie sur la théorie des graphes en introduisant des arcs disjonctifs (par exemple [19]).

II.1.1. Programmes linéaires en nombres mixtes

Parmi les nombreuses formulations de problèmes d'ordonnancement avec contraintes de moyens, nous avons choisi d'en exposer deux qui paraissent assez représentatives de l'ensemble des programmes linéaires en nombres mixtes proposés.

II.1.1.1. Problème d'ordonnancement n/m/G/P et n/M/G/P ([35])

Soit t la période élémentaire de temps. Si j caractérise un moyen (machine, personnel,...) notons I_j l'ensemble des tâches i utilisant le moyen j et $q_{i,j}$ la consommation de moyen j utilisée pour réaliser l'opération $O_{i,j}$ de la tâche i. Soit $C_{i,j}$ la date de début de l'opération $O_{i,j}$.

Si on introduit deux opérations fictives $O_{0,0}$ et $O_{n+1,m+1}$, alors $C_{0,0} = 0$ et $C_{n+1,m+1} = P$, durée de l'ordonnancement.

Le problème général peut se formuler ainsi :

Trouver des nombres $C_{i,j} \geq 0 \begin{cases} i \in N = \{1,2,\ldots,n+1\} \\ j \in K = \{1,2,\ldots,m+1\} \end{cases}$

qui minimisent P avec

$$(1) \quad \begin{cases} C_{i,k}-C_{i,j} \geq f_{i,j} \quad i=1,\ldots,n \ ; \ (j,k) \in H \subset K \times K \\ \qquad f_{i,j} \text{ étant la durée de l'opération } O_{i,j} \\ C_{i,j}-C_{0,0} \geq C_i \quad i=1,\ldots,n \ ; \quad j \in K \end{cases}$$

C_i étant la date de début au plut tôt de la tâche i.

$$\begin{cases} \sum_{i\in I_j} q_{i,j} X^t_{i,j} \leq b_j \qquad j = 1,\ldots,m \\ \quad \text{avec} \\ \qquad X^t_{i,j} = \begin{cases} 1 \text{ si } C_{i,j}<t\leq C_{i,j}+f_{i,j} \\ 0 \text{ sinon} \end{cases} \quad \begin{vmatrix} i=1,\ldots,n \\ j=1,\ldots,m \end{vmatrix} \end{cases}$$

b_j représentant la capacité en moyen j

<u>Remarque</u>

Dans la formulation précédente, si toutes les tâches ont même date de début au plus tôt (par exemple $C_i = 0$ pour tout i) il suffit d'affecter à l'opération fictive $O_{0,0}$ la durée 0.

Deux cas sont particulièrement importants dans les problèmes d'atelier, suivant que chaque groupe de machines comporte une seule ou plusieurs machines. Nous parlerons alors respectivement d'<u>atelier simple</u> et d'<u>atelier multiple</u>.

1. Ordonnancement d'atelier simple : n/m/G/P

Les hypothèses sont les suivantes :

(H1) : chaque tâche doit passer 0 ou 1 fois sur chaque groupe de machine dans un ordre bien déterminé.

(H2) : le choix de l'ordre dans lequel les différentes tâches passent sur les machines est indifférent.

(H3) : chaque groupe de machine comprend une seule machine.

Le système (2) s'écrit alors

$$(2a) \quad \begin{cases} \sum_{i \in I_j} X_{i,j}^t \leq 1 & j=1,\ldots,m \\ \text{avec } X_{i,j}^t = \begin{cases} 1 & \text{si } C_{i,j} < t \leq C_{i,j}+f_{i,j} \\ 0 & \text{sinon} \end{cases} \end{cases}$$

2. Ordonnancement d'atelier multiple : n/M/G/P

Les hypothèse précédentes (H1) et (H2) subsistent mais (H3) est remplacée par l'hypothèse (H'3)

(H'3) : le groupe de machine j comprend M_j machines identiques

$(M_j \geq 1 \quad$ pour tout $j = 1,2,\ldots,m)$

Le système (2) devient alors

$$(2b) \quad \begin{cases} \sum_{i \in I_j} X_{i,j}^t \leq M_j & j=1,\ldots,m \\ \quad \text{avec } X_{i,j}^t = \begin{cases} 1 \text{ si } C_{i,j} < t \leq C_{i,j} + f_{i,j} \\ 0 \text{ sinon} \end{cases} \end{cases}$$

3. Evaluation du nombre de variables et de contraintes

Le nombre de contraintes (2a) ou (2b) et surtout le nombre de variables booléeiennes $X_{i,j}^{t}$ peuvent être très grands car la période élémentaire t intervient explicitement dans les formules.

Evaluons ces nombres pour le problème d'atelier simple n/m/G/P en supposant que toutes les tâches arrivent à la date 0 (date de début au plus tôt) et utilisent toutes les machines, c'est-à-dire que $I_j=\{1,\dots,n\}$ pour tout $j\in\{1,\dots,m\}$

Variables	nombre
$C_{i,j}\geq 0$ entières	n m
$X_{i,j}^{t}=\begin{cases}1\\0\end{cases}$	n m P

Contraintes	nombre
$C_{i,k}-C_{i,j} \geq f_{i,j}$	n m
$\sum\limits_{i\in N} X_{i,j}^{t} \leq 1$	m P

On obtient donc un programme linéaire en variables entières comportant nm(P+1) variables et n(n+P) contraintes, la fonction à minimiser étant $C_{n+1,m+1}$

Exemple : 10/4/G/P : dans ce cas on obtient un programme linéaire avec 1040 variables et 140 contraintes pour une durée d'ordonnancement de 25 unités de temps.

Cette méthode apparait donc comme particulièrement lourde.

II.1.1.2. Formulation de MANNE pour le problème n/m/G/V

La modélisation de MANNE [17], est celle qui est le plus souvent citée dans les articles consacrés à l'ordonnancement. Elle part d'un programme non linéaire en nombres mixtes qui est rendu linéaire en donnant une borne B au temps d'ordonnancement maximum P.

Les hypothèses sont celles (H_1, H_2, H_3) de l'atelier simple précédent.

Soit les nouvelles variables $b_{i,j,k}$ valant 1 si la $k^{\text{ième}}$ opération de la tâche i nécessite j, 0 sinon.

En remarquant que $\sum_j b_{i,j,k} C_{i,j}$ est la date de début de la $k^{\text{ième}}$ opération de la tâche i, les contraintes de précédence sur les opérations d'une même tâche s'écrivent :

$$(3) \qquad \sum_j b_{i,j,k} (C_{i,j} + f_{i,j}) \leq \sum_j b_{i,j,k+1} C_{i,j}$$

Pour une machine j donnée il faut déterminer l'ordre de passage des tâches r et s, c'est à dire vérifier l'une ou l'autre des inégalités

$$(4) \quad \begin{cases} C_{r,j} - C_{s,j} \geq f_{s,j} \\ \text{ou} \\ C_{s,j} - C_{r,j} \geq f_{r,j} \end{cases}$$

Soit $Y_{r,s,j} = \begin{cases} 1 & \text{si la tâche r précède (pas nécessairement directement)} \\ & \text{la tâche s sur la machine j} \\ 0 & \text{sinon} \end{cases}$

et soit B une constante assez grande (par exemple $B = \sum_i \sum_j f_{i,j}$).

Le système (4) est alors équivalent au système (5).

$$(5)\quad \begin{cases} (B+f_{s,j})\ Y_{r,s,j} + (C_{r,j}-C_{s,j}) \geq f_{s,j} \\ (B+f_{r,j})(1-Y_{r,s,j}) + (C_{s,j}-C_{r,j}) \geq f_{r,j} \end{cases}$$

dans lequel une des contraintes est toujours vérifiée.

Le programme linéaire à résoudre est donc le suivant :

Trouver des entiers positifs $C_{i,j} (i,j) \in H \subset \{1,\ldots,n\} \times \{1,\ldots,m\}$

et des booléens $Y_{r,s,j}$ $\left| \begin{array}{l} j = 1,\ldots,m \\ (r,s) \in K \subset \{1,\ldots,n\} \times \{1,\ldots,n\} \end{array} \right.$

vérifiant les contraintes

$$\sum_j b_{i,j,k}(C_{i,j}+f_{i,j}) \leq \sum_j b_{i,j,k+1}\ C_{i,j} \qquad \left| \begin{array}{l} i = 1,\ldots,n \\ k = 1,\ldots,\mathrm{Card}\ (C_{i,j})^{-1} \end{array} \right.$$

$$\begin{array}{l} (B+f_{s,j})\ Y_{r,s,j} + (C_{r,j}-C_{s,j}) \geq f_{s,j} \\ (B+f_{r,j})(1-Y_{r,s,j})+(C_{s,j}-C_{r,j}) \geq f_{r,j} \end{array} \qquad \left| \begin{array}{l} j = 1,\ldots,m \\ (r,s) \in K \end{array} \right.$$

et qui optimisent une certaine fonction "économique".

Cette fonction à optimiser peut prendre différentes formes suivant le type de critère. Les trois cas les plus classiques sont les suivants.

<u>1.n/m/G/$\bar{P}$</u>

$$\min \bar{P} \iff \min \left(\sum_i \sum_j b_{i,j,\ell}\ C_{i,j}\right)$$

ℓ étant la dernière opération de la tâche i ($\ell \leq m$)

<u>2. n/m/G/P</u>

$$\min P \text{ avec } (6) \sum_j b_{i,j,k}(C_{i,j}+f_{i,j}) \leq P \qquad i=1,\ldots,n$$

3. n/m/G/R

Dans le cas où chaque tâche i a une date de fin imposée d_i, nous avons vu que $R_i - A_i = P_i - p_i$ avec $p_i = d_i - c_i$

La fonction à optimiser est $\min(\sum_i R_i)$

Evaluons les nombres de variables et de contraintes dans le cas du problème n/m/G/P en faisant l'hypothèse que chaque tâche comporte m opérations :

Variables	nombre
$C_{i,j} \geq 0$ entières	n m
$Y_{r,s,j}$ booléennes	$m \cdot \frac{n(n-1)}{2}$

Contraintes	nombre
(3)	n(m-1)
(5)	mn(n-1)
(6)	n

Cela donne, par exemple, pour un problème 10/4/G/P 220 variables et 400 contraintes, ce qui est important pour un problème de petite taille.

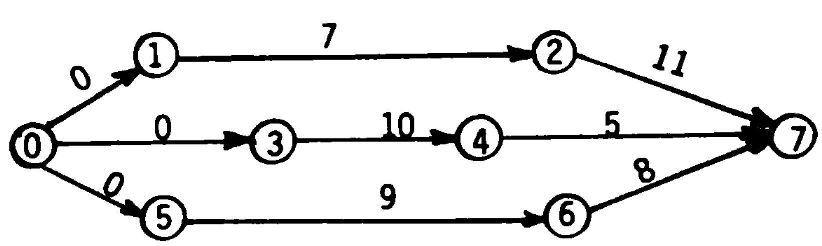
II.1.1.3. Conclusion

La modélisation par des programmes linéaires en variables mixtes permet de formuler le problème général n/m/G/V mais nous obtenons rapidement un nombre important de variables et de contraintes. En effet, s'il est relativement facile d'introduire les contraintes technologiques définissant les séquences d'opérations d'une même tâche, il est plus difficile de formuler les contraintes de charge de machines. C'est pour cette raison que nous n'avons pas donné l'extension de la formulation de MANNE [17] dans le cas d'ateliers multiples. Les notations y sont particulièrement alourdies par l'introduction de nouvelles variables et la modification des contraintes.

II.1.2. Contraintes disjonctives dans les graphes

A tout problèmed'ordonnancement ne comportant que des contraintes de succession, on peut associer un graphe conjonctif. C'est la représentation classique potentiels-tâches dans laquelle les contraintes technologiques sont représentées par des arcs entre les sommets du graphe qui définissent les opérations. Le graphe conjonctif traduit le problème technologique.

Exemple : 3 tâches comportant 2 opérations chacune.

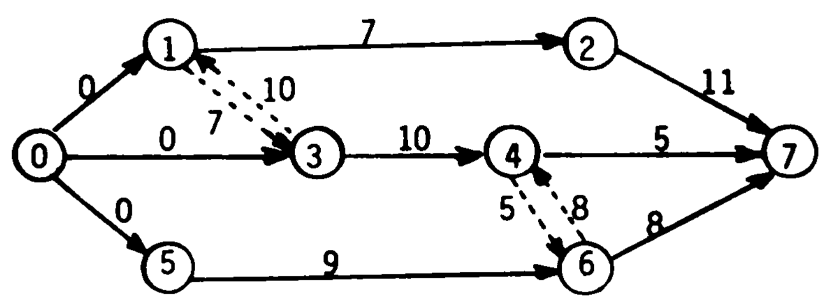


tâche	Opérations
1	1,2
2	3,4
3	5,6

Lorsqu'un certain nombre d'opérations doivent être effectuées à l'aide d'un même moyen, leurs temps d'exécution sont nécessairement disjoints. Il faut alors introduire (cf. [19]) ces contraintes disjonctives dans le graphe du problème technologique. Cela se traduit par des paires d'arcs disjonctifs entre des couples de sommets du graphe.

Exemple

Les opérations (1) et (3) d'une part, (4) et (6) d'autre part utilisent la même machine. Il faut donc introduire deux paires d'arcs disjonctifs entre (1) et (3) d'une part et (4) et (6) d'autre part. D'où le graphe :



Formellement, nous dirons :

Définition 1

Deux arcs d'un graphe G sont disjonctifs et forment une paire disjonctive si tout chemin dans G peut utiliser au plus un élément de la paire.

Les problèmes d'ordonnancement à ressources limitées peuvent donc être pris en compte mathématiquement dans le cadre d'un graphe généralisant celui de la formulation potentiels-tâches. Comme dans ceux-ci il s'agit de déterminer un chemin critique de longueur minimale. Le seul critère mis en oeuvre ici est la durée D de l'ordonnancement (ou P puisque la date de début au plus tôt de toutes les tâches est supposée commune et prise comme origine). Les problèmes ainsi formulés sont donc du type n/m/G/P.

A tout problème d'ordonnancement de ce type on associe un graphe G = (X,Y,Z) avec X l'ensemble des sommets, Y l'ensemble des arcs conjonctifs (problème technologique) et Z l'ensemble des arcs disjonctifs (contrainte de charge des machines).

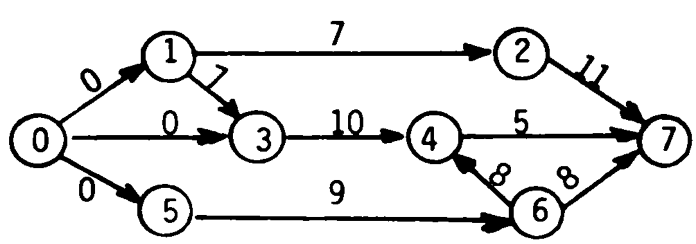
Le problème consiste alors à trouver un graphe G' = (X,U), où U est un ensemble d'arcs conjonctifs, à partir du graphe G = (X,Y,Z) en faisant un arbitrage complet et compatible des disjonctions.

Définition 2

Un sous ensemble $S_h \subset Z$ contenant au plus un arc par paire disjonctive est appelée Sélection (d'arcs disjonctifs). Une sélection comportant exactement un arc de chaque paire disjonctive est appelée sélection complète (par opposition à une sélection partielle) ou arbitrage complet.

Exemple

Reprenons l'exemple précédent. Une sélection complète consiste à choisir par exemple l'arc (1 , 3) et l'arc (6 , 4). Ce qui donne le graphe suivant :



Nous allons maintenant distinguer deux classes de problèmes et nous envisagerons, pour terminer, une extension possible par l'introduction des délais pour les tâches.

II.1.2.1. Ordonnancement d'atelier simple n/m/G/P

Si G = (X,Y) est le graphe conjonctif obtenu sans tenir compte des contraintes de charge des machines, soit $X_k \subset X$ le sous-ensemble des sommets correspondant aux tâches à exécuter sur la machine k.

Introduisons l'ensemble Z d'arcs disjonctifs tel que

$$Z = \bigcup_{k=1}^{m} \left\{ (i,j) \in X_k \times X_k \;/\; i \neq j \text{ et } (i,j) \notin Y \text{ et } (j,i) \notin Y \right\}.$$

Alors $\begin{pmatrix} (i,j) \in Z \\ (r,s) \in Z \end{pmatrix}$ forment une paire disjonctive si et seulement si $s=i$ et $r=j$.

L'arc (i,j) sera valué par la durée f_i de l'opération i. Chaque sélection complète S_h génère un graphe conjonctif noté $G_h = (X, Y \cup S_h)$. Soit S l'ensemble des sélections complètes, alors

$$S' = \{S_h \in S / G_h = (X, Y \cup S_h) \text{ n'a pas de circuit}\}$$

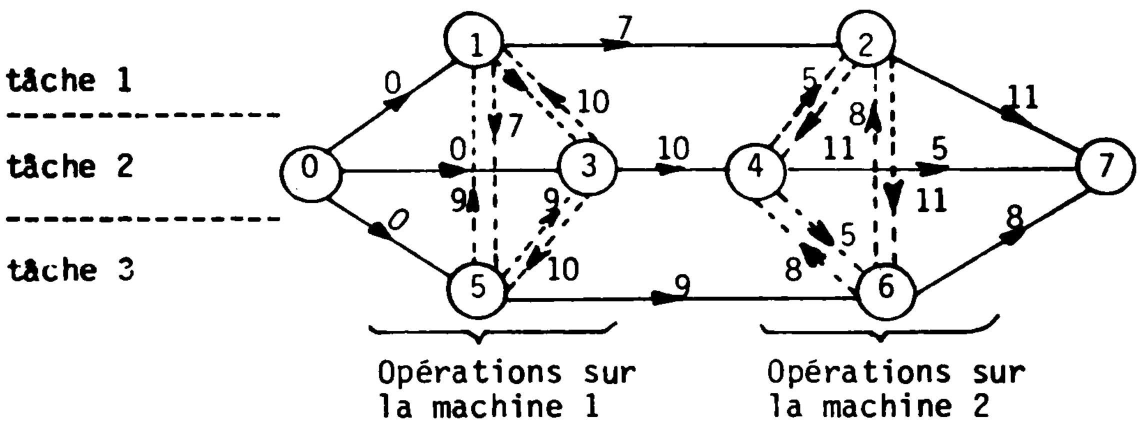
$$G' = \{G_h = (X, Y \cup S_h) / S_h \in S'\}.$$

Le problème d'ordonnancement d'atelier simple se réduit alors à trouver $S_r \in S'$ et $G_r \in G'$ tel que, si L_r est son plus long chemin, alors $L_r = \underset{S_h \in S'}{\mathrm{Min}} (L_h)$.

Le nombre de sélections complètes est $\mathrm{Card}(S) = 2^{\alpha}$ avec $\alpha = \frac{1}{2}\,\mathrm{Card}\,(Z)$, celui des sélections compatibles Card (S') est très inférieur puisque borné par m.n!.

Par exemple, pour un problème de n=4 tâches, m=10 machines et 40 opérations, le nombre de paires disjonctives est $m(\frac{n(n-1)}{2}) = 60$ et $\mathrm{Card}\,(S) = 2^{60}$. Pour chaque machine k, $\mathrm{Card}\,(S^k) = 2^6 = 64$ alors qu'en fait le nombre de sélections compatibles est seulement $\mathrm{Card}(S'^k) = 4! = 24$.

La représentation du graphe correspondant devient très vite inextricable, surtout dans le cas d'un atelier de type à flot quelconque. Comme illustration, nous donnons ci-dessous le cas d'un ordonnancement à 3 tâches et 2 machines pour un atelier à flot unique (3/2/F/P).



0 et 7 sont deux opérations fictives respectivement début et fin de l'ordonnancement.

II.1.2.2. Ordonnancement d'atelier multiple n/M/G/P

Dans le groupe de machines $k(k=1,\ldots,M)$ il y a M_k machines identiques et N_k opérations sont à exécuter sur ce groupe k. Le sous-graphe S_h^k correspondant aux machines du groupe k devra comporter N_k-M_k arcs formant M_k chemins disjoints de longueur non négative. S_h^k sera donc une sélection partielle dans le sous-graphe disjonctif Z^k.

Donc, dans le cas d'un atelier multiple, il suffit de chercher parmi toutes les sélections partielles $S_h = \bigcup_{k=1}^{M} S_h^k$ celle qui minimise P. S_h appartient à S", ensemble des sélections partielles que $G_h=(X,Y\cup S_h)$ est sans circuit.

Le nombre de sélections partielles possibles, Card(S"), est de l'ordre de Mxnxn!. Il dépend essentiellement du nombre de tâches concernées et varie en fonction du nombre de machines par groupe.

Prenons deux exemples où l'on s'impose d'affecter une opération au moins à chaque machine de chaque groupe.

1. 3/2/G/P

avec $M_k=2$, $N_k=3$ pour $k=1,2$.

$S_h \in S''$ comporte 3-2=1 arc par machine, ce qui est un cas particulièrement simple. Card(S") est égal à 6.

2. 10/5/G/P

avec $M_k=2$, $N_k=10$ pour $k=1,\ldots,5$.

Chaque sélection S_h^k comporte 8 arcs formant deux chemins disjoints. Le nombre de sélections partielles est 25 x 10!.

II.1.2.3. Limites des graphes disjonctifs : extension proposée

Tels que formulés précédemment, les graphes à contraintes disjonctives représentent les problèmes d'ordonnancement où le critère mis en oeuvre est la durée maximale P de l'ordonnancement. Si les dates d'arrivées différentes pour les tâches peuvent être facilement prises en compte (voir exemple suivant), il n'en est pas de même pour la notion de délai.

Nous avons trouvé personnellement un moyen artificiel d'introduire les contraintes de délai de réalisation pour chaque tâche : pour toute tâche i, on introduit une opération supplémentaire "fin de tâche i" notée σ_i. Soit d_i la date de fin au plus tard de la tâche i et prenons 0 pour origine de l'ordonnancement ce qui permet de faciliter l'exposé de la suite.

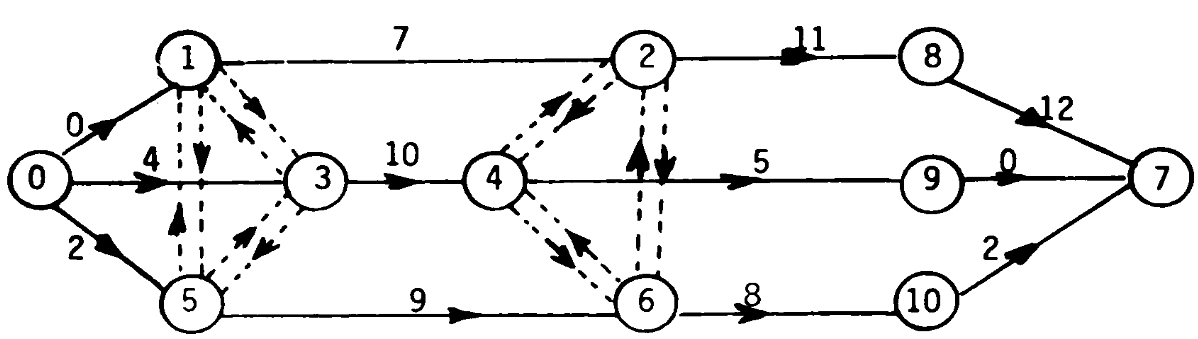
Si $P = \underset{i}{\mathrm{Max}}(d_i)$ alors σ_i aura pour durée $P-d_i$. Dans ce graphe $G = (X \cup \{\sigma_i\}, Y, Z)$ si on trouve un chemin critique minimum $L^* \leq P$ alors le problème est possible (i.e. les tâches ne seront pas en retard).

Exemple

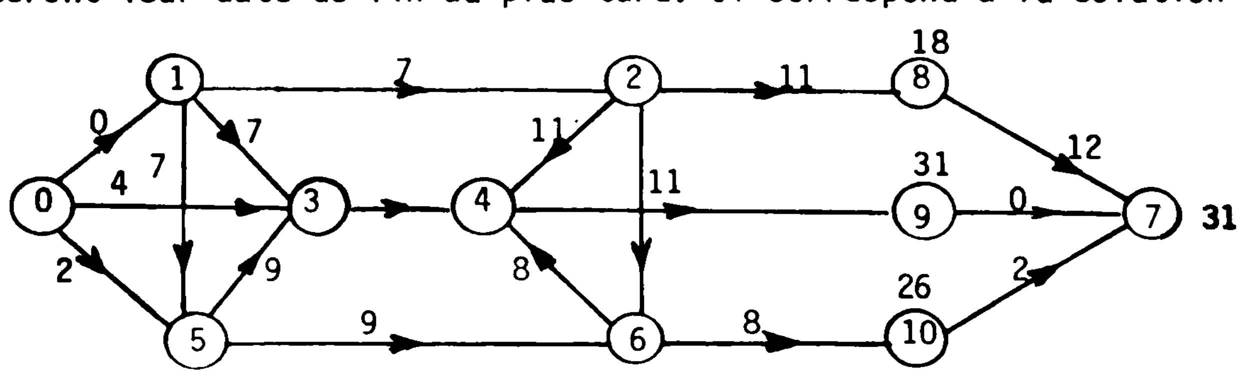
Reprenons l'exemple précédent en introduisant des dates de début au plus tôt c_i et des dates de fin au plus tard d_i données par le tableau

tâche$_i$	c_i	d_i	σ_i
1	0	20	8
2	4	32	9
3	2	30	10

P est égal à 32 donc les durées des opérations fictives (8), (9), (10) seront respectivement 12, 0 et 2.



Le chemin critique minimum est $L^* = 31$: donc les tâches respecteront leur date de fin au plus tard. Il correspond à la solution suivante:



II.1.2.4. Conclusion

L'accroissement de la taille du modèle est très rapide. En particulier cela est du au très grand nombre d'arcs disjonctifs à introduire lorsque le nombre de tâches est élevé.

Par sa nature même cette méthode est bien adaptée à la recherche de la longueur d'un chemin. Mais, malgré l'extension proposée il est difficile de l'adapter à une résolution optimale vis à vis d'un autre critère que le temps de passage P.

II.1.3. Conclusion sur les formalisations

Tout problème d'ordonnancement peut donc se formuler soit, comme un programme linéaire en nombres mixtes, soit comme un problème de chemin critique dans un graphe comportant des arcs conjonctifs et des arcs disjonctifs. On peut également formuler ce problème comme le fait par exemple FORTET [18] ou PIERCE [33] à l'aide de programmes booléens.

La formulation reste relativement simple lorsqu'on s'intéresse uniquement à la durée de l'ordonnancement. Lorsqu'on veut modéliser un problème plus complexe, tel atelier multiple ou atelier dans lequel des délais sont affectés aux tâches, la taille du modèle s'accroit de façon très importante.

II.2. LES METHODES DE RESOLUTION

Les méthodes de résolution des problèmes d'ordonnancement d'atelier formulés dans le paragraphe précédent peuvent être réparties en quatre grandes familles :

- les méthodes analytiques combinatoires,

- les codes de programmation linéaire en variables mixtes

- les méthodes d'énumération implicite

- les méthodes heuristiques

Nous laissons volontairement de côté les méthodes qui procèdent par énumération complète des ordonnancements possibles. Pour un problème n/m/G/P il y a de l'ordre de $(n!)^m$ ordonnancements parmi lesquels beaucoup sont irréalisables pour des raisons technologiques. Ces méthodes ne peuvent être envisageables étant donné leur coût prohibitif.

Nous nous intéresserons donc aux quatres groupes précédents en donnant pour chacun les caractéristiques principales et les avantages ou inconvénients relatifs.

II.2.1. Méthodes analytiques combinatoires

Ces méthodes procèdent toutes du même principe. L'analyse de l'ensemble des solutions permet de dégager des sous-ensembles remarquables et de réduire la dimension des ensembles à explorer. Cette propriété est la dominance qui signifie que toute solution dominante est meilleure au sens du critère considéré qu'une solution dominée correspondante.

Une première approche est due à CONWAY [1] qui introduit les notions d'ordonnancements actifs et semi-actifs. Le principal résultat établit que la solution optimale à tout problème avec critère régulier (par exemple le temps de passage P ou le retard R) appartient à l'ensemble des ordonnancements actifs.

Une deuxième approche concerne la caractérisation d'un ensemble d'ordonnancements admissibles par l'analyse sous contraintes ([8], [20]). L'objectif est dans ce cas le respect des délais attribués aux tâches.

Des conditions nécessaires de dominance sont démontrées dans deux thèses ([11] et [12]). Pour COUZINET [11] cette recherche est faite par l'étude des structures pyramidales des diagrammes à barres (diagrammes de GANTT). Par contre dans [12], les relations sont établies par la recherche de circuits dans des graphes. La dimension du sous-ensemble d'ordonnancements dominants est alors d'autant plus réduite que le corps d'hypothèses est plus complet. FONTAN [12] utilise ensuite une optimisation par la méthode de Séparation et Evaluation Progressive (méthode S.E.P. exposée au paragraphe II.2.3.), les critères retenus étant P ou R pour des problèmes à une seule machine.

Les résultats obtenus par ces méthodes sont intéressants car ils peuvent être utilisés dans le cadre d'une aide à la décision. Malheureusement, ils sont surtout obtenus pour des problèmes n/1//V et partiellement dans le cas n/m/F/V. La résolution du problème général n/m/G/V ne peut, pour l'instant, être envisagé par ces méthodes.

II.2.2. Programmation linéaire en nombres entiers

Nous présentons d'abord une des tentatives de résolution par un code de programmation linéaire en nombres mixtes (P.L.M.). Puis, nous indiquons les limites de ces méthodes. Enfin, nous inclurons dans cette classification une approche intéressante par fonction de pénalisation [4].

II.2.2.1. Une expérience de résolution par P.L.M.

Il s'agit de la résolution d'un problème de type $n/3/F/I_3$ due à WAGNER et GIGLIO [21]. L'objectif I_3 est de minimiser l'inactivité de la machine 3 :

$$\text{Min} \left(\sum_{j=1}^{n} X_j^3 \right)$$ où X_j^k représente l'inactivité de la machine k avant le début de la tâche en position j.

Si l'on définit les variables suivantes :

$$z_{i,j} = \begin{cases} 1 \text{ si la tâche i est ordonnancée en } j^{\text{ième}} \text{ position} \\ 0 \text{ sinon} \end{cases}$$

$Z(j) = [z_{1,j}, z_{2,j}, \ldots, z_{n,j}]$ vecteur colonne

Y_j^k l'attente de la tâche en position j entre sa fin sur la machine k et son début sur la machine k+1,

A,B,D les vecteurs lignes des temps de fabrication des tâches 1,2,...,n respectivement sur les machines 1,2,3.

Les contraintes s'écrivent alors

$$\sum_i z_{i,j} = 1 \qquad \sum_j z_{i,j} = 1$$

$$\left. \begin{array}{l} X_{j+1}^2 + BZ(j+1) + Y_{j+1}^2 - Y_j^2 - DZ(j) - X_{j+1}^3 = 0 \\ AZ(j+1) + Y_{j+1}^1 - Y_j^1 - BZ(j) - X_{j+1}^2 = 0 \end{array} \right| \; j=1,\ldots,n-1$$

La résolution de ce P.L.M. utilise l'algorithme de GOMORY [34]. Le tableau ci-dessous est un extrait des résultats obtenus. Il montre le nombre d'itérations nécessaires pour obtenir la solution de cinq problèmes $6/3/F/I_3$. Pour chaque problème trois variantes ont été étudiées en introduisant des contraintes supplémentaires.

Problème / Variante	A	B	C	D	E
1	44	1647	218	pas de sol	9548
2	50	710	63	858	2811
3	50	208	58	245	304

(pas de sol. signifie que 10000 itérations n'ont pas permis de trouver une solution).

II.2.2.2. Des possibilités limitées

Ces résultats ne sont pas très encourageants d'autant plus qu'il s'agit d'un atelier à flot unique F plus simple à résoudre que l'atelier de type G.

Théoriquement, la résolution d'un programme linéaire en nombres mixtes permet d'obtenir la solution optimale. En pratique, cela est possible pour de petits ateliers, mais le coût devient beaucoup trop élevé lorsque la taille de l'atelier augmente ([3]).

En l'état actuel des codes de PLM, peu de progrès peuvent être réalisés dans cette voie. Finalement il apparaît très maladroit de résoudre un problème d'ordonnancement sous contraintes de cette manière car on n'utilise pas la structure particulière de tels problèmes. Ces caractéristiques sont mises en évidence dans la modélisation par les graphes disjonctifs (cf. paragraphe II.1.2.).

II.2.2.3. Approche par fonction de pénalisation

Une extension du problème n/M/G/R a été proposée par NEPOMIASTCHY [4]. La résolution est conduite par l'introduction d'une fonction de pénalisation. Cette dernière permet de donner un sens économique et réaliste au problème.

L'extension principale est la suivante : l'atelier est multiple et à ressources variables dans le temps. Chaque groupe j comporte $M(j,\theta)$ machines (θ désignant le temps).

Soit $E = \{u=\{C_{i,j}\}/\forall(i,j)\ C_{i,j}$ entier : $a_{i,j} < C_{i,j} < d_{i,j}\}$
et $G(u)$ une fonction de pénalisation positive ou nulle de la contrainte

$$J(j,\theta) \leq M(j,\theta) \quad \left| \begin{array}{l} \forall j=1,\dots,m \\ \forall \theta \in T \text{ période de l'ordonnancement} \end{array} \right.$$

où $J(j,\theta)$ est le nombre d'opérations exécutées sur le groupe j pendant l'intervalle de temps $[\theta,\theta+1[$.

Le problème à résoudre est :

Minimiser $G(u)$ avec $u \in E$ ou si possible trouver $u \in E$ tel que $G(u) = 0$.

L'ordonnancement recherché est donc tel que les délais sont tous respectés. Si c'est possible on recherche un ordonnancement qui respecte en plus la charge des machines, ou sinon celui qui viole le moins la contrainte des ressources variables. Dans ce cas, on obtient l'ordonnancement qui entraîne la coût supplémentaire le plus faible.

L'algorithme donne de bons résultats mais il peut se poser des problèmes de convergence. En effet, on part d'une solution initiale quelconque et s'il l'on n'obtient qu'un optimum local, on est obligé de calculer une nouvelle solution initiale. Par la suite, nous utilisons cet algorithme comme partie de notre modèle, mais nous le faisons de manière non standard ce qui nous permet d'éviter ces problèmes de convergence.

II.2.3. Enumération implicite

Les méthodes que nous regroupons sous ce nom procèdent par parcours dans l'arbre des solutions possibles en n'envisageant cependant qu'un nombre restreint d'entre elles. Le nom généralement donné à ces méthodes est "Séparation et évaluation progressive" (S.E.P. ou Branch and

Bound). Ci-dessous nous montrons le principe général, puis nous nous interessons à une autre méthode procédant aussi par énumération implicite mais de philosophie différente puisque le but est d'améliorer une solution initiale. La méthode S.E.P. construit au contraire une solution de toute pièce.

II.2.3.1. Algorithme S.E.P.

Cette méthode de séparation et évaluation progressive est basée sur un algorithme comportant trois procédures :

- une procédure de séparation ("Branching") qui partitionne l'ensemble des solutions en deux sous-ensembles complémentaires.

- une procédure d'évaluation ("Bounding") de bornes inférieures pour le critère qui est généralement le temps de passage P ou la durée $\mathcal{D}$ de l'ordonnancement.

- une procédure de progression qui permet de choisir une branche de l'arbre dans laquelle l'exploration va se poursuivre.

Le critère d'évaluation de l'ordonnancement est contenu dans la procédure d'évaluation. Dans la majorité des études faites (cf. par exemple [14], [26], [29]), c'est le temps de passage maximum P que l'on cherche à minimiser, pour des problèmes n/m/F/P ou n/m/G/P. A notre connaissance, une seule étude utilise une résolution par énumération implicite avec les délais comme critère ([13]) pour un problème n/m/G/E en se ramenant à un problème n/m/G/P.

Le succès des méthodes S.E.P. dans la résolution du problème général d'ordonnancement d'atelier est surtout fonction de l'efficacité du calcul des bornes inférieures. Les efforts de calcul restent cependant très importants. Très souvent, une solution est obtenue assez rapidement, mais il reste à prouver qu'elle est vraiment optimale, ce qui est très coûteux en temps. Toutefois, on peut se contenter de rechercher des solutions proches ou non de l'optimum, ce qui permet une résolution relativement rapide.

Il faut remarquer que la formulation du problème d'ordonnancement par des arcs disjonctifs dans des graphes est particulièrement bien adaptée à une résolution par S.E.P. En effet, descendre dans l'arbre revient à arbitrer une paire de disjonctions (cf. paragraphe II.1.2.).

La racine de l'arbre correspond au graphe sous les arcs disjonctifs. Chaque étape consiste à introduire l'arbitrage d'une paire disjonctive. Une solution est obtenue lorsque le graphe disjonctif est complètement arbitré.

II.2.3.2. Enumération implicite de BALAS [10]

Cette méthode est fondée sur deux propriétés des graphes d'ordonnancement d'atelier simple que nous rappelons ci-dessous.

P1 :

Si C_h est un chemin critique dans $G_h = (X,Y \cup S_h)$, tout graphe G_k obtenu à partir de G_h en remplaçant un arc d'une paire disjonctive par l'autre du chemin critique est sans circuit.

P2 :

Si C_h est un chemin critique de longueur L_h dans G_h, et s'il existe G_k de chemin critique C_k de longueur $L_k < L_h$, alors la sélection S_k contient le complément d'au moins un arc d'une paire disjonctive appartenant au chemin C_h.

Le principe de l'algorithme de Balas est de partir d'un ordonnancement réalisable, où tous les arcs disjonctifs sont arbitrés. On génère ensuite de nouveaux graphes conjonctifs tous réalisables par deux types de pas : pas en avant et pas en arrière qui correspondent au changement d'arc dans une paire disjonctive. Les problèmes résolus ainsi sont des problèmes n/m/G/P.

La propriété la plus intéressante de cette méthode est de fournir dès le début des solutions réalisables contrairement à la méthode S.E.P. Pour un problème de grandes dimensions ceci permet d'améliorer aisément une solution de départ sans être nécessairement obligé d'effectuer de longs calculs.

II.2.3.3. Méthodes heuristiques

Avec les méthodes décrites ici il n'est plus possible d'obtenir la solution optimale à un problème donné, mais seulement une solution relativement bonne.

Les procédures de génération d'ordonnancements par ces méthodes heuristiques sont caractérisées par l'ordre suivant lequel les opérations sont sélectionnées et la manière dont les dates de début sont déterminées. Ceci permet de faire la distinction fondamentale entre les procédures à une passe et les procédures à essais successifs.

Dans les procédures à une passe, toute affectation d'une date de début à une opération est définitive. Dans les procédures à essais successifs, chaque affectation de date de début à une opération est une tentative sujette à modifications jusqu'à ce que l'ordonnancement tout entier soit parfaitement déterminé.

A notre connaissance, la majorité des algorithmes heuristiques sont à une passe (cf. par exemple [22], [25]). Ces méthodes font appel à la notion d'ensemble d'opérations "ordonnançables" qui sont en général celles sans prédécesseurs non ordonnancés. Ces opérations forment des files d'attente devant les machines concernées. La sélection est faite par des règles de priorité du type premier arrivé premier servi, plus courte durée de fabrication, plus court délai, etc... (pour d'autres exemples, voir [1]). Ces règles de priorité peuvent être plus complexe en faisant intervenir une pondération sur les différents critères (voir par exemple certains produits-programmes de constructeurs comme MAPICS d'IBM, TZAR de Production System ou ORACLE de CII-HB). Cela permet de trouver une ou plusieurs bonnes solutions (bonne relativement à un des critères) au problème posé.

Pour un problème donné, il est toujours possible de trouver une procédure à une passe qui fournira le même ordonnancement qu'une procédure à essais successifs. Il suffit de changer les règles de priorités. Mais ceci est un inconvénient majeur puisqu'il faudra trouver une procédure à une passe pour chaque problème, alors qu'une procédure "récursive" permettra sans modification de les traiter tous.

Un exemple de telle procédure à essais successifs a été testée par CRABILL [1] pour des problèmes n/m/F/P. Il s'agit d'une méthode ordonnançant tâche après tâche. Pour chaque opération à ordonnancer, on calcule la pénalité sur le temps de passage maximum et on déduit ainsi la meilleure position pour l'insérer dans l'ordonnancement partiel provisoire. Les opérations déjà ordonnancées sont éventuellement déplacées pour pouvoir réaliser cette insertion. Une approche itérative est d'autre part utilisée pour déterminer l'ordre de prise en compte des tâches.

Actuellement, les méthodes heuristiques, bien qu'adaptées à des problèmes restreints par les hypothèses introduites, permettent cependant de résoudre au mieux les problèmes généraux de type n/m/G/P ou n/m/G/R de taille relativement importante.

II.3. CAS PARTICULIERS RESOLUS

Nous considérons ici des résultats obtenus dans des cas particuliers qui ne nécessitent donc pas d'être résolus par les méthodes vues précédemment.

II.3.1. Problèmes n/1//V

Ce sont évidemment les problèmes les plus simples car il suffit de déterminer une séquence d'opérations sur la machine unique. Tout en étant simples, ils sont pourtant intéressants. En effet, ils permettent d'une part une approche des problèmes plus généraux (au niveau des mesures de performances). D'autre part ils existent dans la réalité, ne serait-ce que dans des ateliers à plusieurs machines indépendantes les unes des autres.

De très nombreuses recherches ont été effectuées dans ce domaine et la séquence optimale a été trouvée pour la plupart des critères possibles (pour plus de détails voir par exemple [9] et [1]).

II.3.2. L'atelier à flot unique

L'étude et les résultats les plus souvent cités sont ceux du problème n/2/F/P du à Johnson [15]. C'est certainement à lui qu'il faut attribuer le fait que le critère du temps de passage P soit la mesure de performance la plus souvent mise en oeuvre dans les problèmes généraux (cela peut aussi provenir du fait que c'est la plus simple à implémenter).

L'atelier à deux machines n'a pas donné lieu à d'autres résultats pour les autres critères. En particulier les résultats de Johnson ne peuvent pas s'appliquer au critère du temps moyen de passage $\bar{P}$.

Pour le problème n/3/F/P aucune règle de construction de la solution optimale n'a été trouvée sauf dans les cas limites (cf. [15] et [16]).

Le cas très spécial du problème 2/m/F/P a été proposé sous forme graphique par ACHERS [23]. Il est particulièrement irréaliste puisqu'il est limité à deux tâches. Néanmoins il est intéressant à notre point de vue car cette résolution pourrait s'adapter au cas de plusieurs tâches. Il suffit de partitionner ainsi : d'une part n-1 tâches déjà ordonnancées, d'autre part la $n^{\text{ième}}$ tâche. Cela revient à insérer une tâche dans un ordonnancement existant, le problème étant de trouver simplement une bonne solution. Cette méthode semble intéressante pour des ateliers à flot unique. Nous donnons en annexe B un exemple avec 5 machines.

Nous terminerons par le problème général n/m/F/V en citant les deux théorèmes de CONWAY [1] concernant les ordonnancements dominants.

THEOREME 1

Il suffit de considérer les ordonnancements où les tâches passent dans le même ordre sur les deux premières machines pour les critères suivants : P, $\bar{P}$, R, $\bar{R}$.

THEOREME 2

Pour le problème n/m/F/P il suffit de considérer les ordonnancements où les tâches ont le même ordre sur les machines 1 et 2 et le même ordre sur les machines m-1 et m.

Remarque

Ces deux théorèmes ne sont valables que sous l'hypothèse suivante : toutes les tâches ont la même date de début au plus tôt. Ils ne peuvent donc s'appliquer qu'à des problèmes d'ordonnancement de type statique.

II.3.3. L'atelier à flot quelconque

Le seul problème pour lequel on connait une méthode est le problème n/2/G/P où chaque tâche a au plus deux opérations. C'est Jackson [16] qui a adapté l'algorithme de Johnson pour ce problème.

Signalons enfin une adaptation de la méthode graphique (paragraphe II.3.2) au problème 2/m/G/P par NEMHAUSER [24] qui procède de manière analogue. L'extension à n tâches ne nous parait guère intéressante dans ce cas. En effet le flot arbitraire des opérations sur les machines complique beaucoup trop le problème. La résolution suggérée en annexe B ne peut pas être adaptée de façon simple et efficace pour le cas d'un atelier à flot quelconque.

II.4. CONCLUSION SUR LES PROBLEMES STATIQUES

L'étude des différentes méthodes de résolution nous a permis de faire la classification suivante : d'une part les méthodes qui peuvent fournir une solution optimale (P.L.M, énumération implicite) et d'autre part celles qui donnent simplement une solution réalisable plus ou moins proche de l'optimum (heuristiques et énumération implicite).

Parmi les premières la méthode S.E.P. est limitée par le nombre de variables. La méthode P.L.M. l'est à la fois par le nombre de contraintes et de variables. Cette limite est d'autant plus évidente que notre but est de traiter des problèmes réalistes : ateliers multiples, tâches comportant plusieurs opérations sur une même machine, etc...

Les secondes peuvent résoudre assez rapidement un problème donné en fournissant une solution : le problème est alors celui de l'évaluation de cette solution parmi l'ensemble des solutions admissibles. Il faut aussi noter qu'elles ne sont pas suffisamment générales puisqu'elles dépendent essentiellement du type de problème concerné et du critère que l'on veut privilégier.

Le principal inconvénient des méthodes précédentes est clairement le suivant : on ne traite que des ordonnancements statiques où toutes les tâches sont connues à l'avance. On peut cependant prendre en compte le fait que les dates de début au plus tôt sont différentes pour toutes ces tâches, mais alors certains résultats, comme nous l'avons vu au paragraphe II.3.2., ne sont plus valables.

Dans le chapitre suivant nous allons voir comment il est possible de prendre en compte de nouvelles hypothèses permettant de définir un problème d'ordonnancement d'atelier plus réaliste.

CHAPITRE III

LES PROBLÈMES D'ORDONNANCEMENT DYNAMIQUE : UN MODÈLE ET UNE MÉTHODE DE RÉSOLUTION

Les problèmes traités dans le chapitre précédent ne sont pas représentatifs des problèmes d'ordonnancement concrets. Toutes ces méthodes nécessitent la connaissance au même instant de toutes les tâches. La production est ainsi fixée à l'avance, ce qui est peu réaliste. Cependant cette hypothèse est essentielle pour la plupart des méthodes étudiées. De même le choix des critères mis en oeuvre a souvent été influencé par la possibilité d'obtenir des résultats. Nous avons vu que la durée de l'ordonnancement était le critère généralement utilisé comme mesure car il est le plus simple à mettre en oeuvre. Cependant ce n'est pas toujours le plus réaliste.

Nous introduisons ici une méthode d'ordonnancement dynamique qui pallie cet inconvénient majeur. Notre modèle permet de prendre en compte le cas d'un atelier fonctionnant avec une production "à la demande", autrement dit où la connaissance des tâches évolue dans le temps. Nous considérons d'autre part des hypothèses plus générales du type "tâches à structure non linéaire", "groupes de machines" et enfin "délais assignés aux tâches" que nous considérons comme nécessaires à un problème réaliste.

L'approche de telles situations concrètes rendent le modèle nécessairement plus complexe. Nous avons vu que les résultats obtenus par les méthodes de type P.L.M. ou S.E.P. dans les cas statiques ne sont guère encourageants. L'adaptation de ces méthodes dans le cas dynamique paraît encore plus irréalisable par son coût prohibitif, le nombre de variables à introduire devenant beaucoup trop élevé. Par leur nature même, de tels problèmes de production à la demande nécessitent donc une approche différente. Il nous semble, en effet, illusoire de chercher à tout prix une "solution optimale" qui risque d'être remise en cause par des aléas de fabrication.

Nous allons donc préciser le problème dynamique avec les hypothèses propres et les objectifs à atteindre. Après avoir effectué ces choix, nous exposerons en le justifiant, notre modèle.

III.1. LE PROBLEME DYNAMIQUE

III.1.1. Justification du choix "Contexte Dynamique"

Le mode de fabrication de l'atelier dépend essentiellement de la vocation même de l'entreprise et des besoins des clients. Pour une période de temps donnée, on peut distinguer les deux types de production suivants :

- Une production fixée à l'avance où toutes les tâches sont connues dès le début de la période à ordonnancer. C'est le cas, par exemple, d'un atelier qui envisage un réapprovisionnement de stocks. On est alors en présence d'un problème d'ordonnancement de type statique.

- Une production "à la demande" où l'on attend les commandes des clients pour fabriquer. C'est le cas général d'un atelier qui doit réaliser des tâches très diversifiées et où de nombreuses options peuvent intervenir. A la date t on ne connaît qu'un sous-ensemble $E_{T,t}$ des tâches à ordonnancer sur l'intervalle de temps T. C'est la caractéristique d'un cas d'ordonnancement de type dynamique.

Dans la plupart des cas réels, les ateliers fonctionnent avec ces deux types de production imbriqués. Leur ordonnancement, s'il veut correspondre à la réalité, doit donc être dynamique.

III.1.2. Les données

Les notations précédentes sont conservées : les lettres minuscules désigneront les données du problème tandis que les majuscules représenteront les variables. Les cas litigieux, où cette règle est difficile à respecter, seront signalés.

III.1.2.1. Les machines

On dispose de m groupes de machines et toutes les machines d'un même groupe sont identiques. Dans chaque groupe j ($j = 1,\ldots,m$) le nombre de machines dépend du temps θ. Nous noterons $M(j,\theta)$ ce nombre. Ceci permettra de prendre en compte les modifications du parc des machines et en particulier les aléas de fabrication tels que panne ou maintenance.

La capacité de production est donc connue dans le temps. Elle doit être du même ordre que la charge à absorber. S'il n'y a pas accord global entre les possibilités de fabrication et les demandes des tâches, nous verrons (cf chapitre IV) que l'ordonnancement ne pourra pas atteindre ses objectifs. Il faudra alors ajuster les moyens de production. C'est pourquoi $M(j,\theta)$, bien qu'étant une donnée, est noté avec une majuscule.

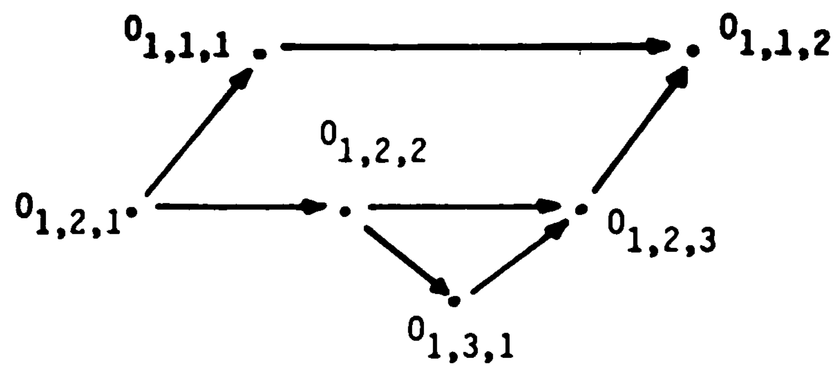
III.1.2.2. Les tâches

Elles sont au nombre de n, mais ces n tâches ne sont pas connues au même instant. Pour chaque tâche i on connaîtra sa date d'arrivée a_i (date à laquelle elle est connue de manière certaine), sa date de début au plus tôt c_i et son délai sous la forme de sa date de fin au plus tard d_i. Chaque tâche est composée d'opérations élémentaires notées $O_{i,j,k}$ (tâche i, groupe j de machines, k désignant la $k^{ième}$ opération sur le groupe j de la tâche i).

L'ordre technologique des différentes opérations pour une tâche i est connu et peut être représenté par un graphe quelconque (i.e. la tâche n'est pas nécessairement linéaire).

Exemple :

Tâche 1, 3 machines, 6 opérations.



Chaque opération $O_{i,j,k}$ a une durée donnée $f_{i,j,k}$. Pour chaque opération $O_{i,j,k}$ et pour tout r et tout s tels que $O_{i,j,k} < O_{i,r,s}$, où $<$ désigne la relation "précède immédiatement", il doit s'écouler un temps $t\ (O_{i,j,k}, O_{i,r,s})$ entre la fin de $O_{i,j,k}$ et le début de $O_{i,r,s}$. Les temps $t(O_{i,j,k}, O_{i,r,s})$ permettent donc de prendre en compte les temps de transit dans l'atelier.

III.1.3. Les hypothèses

Nous avons choisi de les classer en trois groupes, la répartition étant la suivante :

(GH1) : Hypothèses liées à la réalité de l'atelier

(1) Une opération commencée sur une machine ne peut être interrompue par une autre et doit être menée à son terme.

(2) A tout instant, une machine ne peut être chargée avec plus d'une opération.

(3) Chaque tâche peut comporter plusieurs opérations sur un même groupe de machines (indice k).

(4) Comme nous l'avons vu précédemment, une opération peut avoir plusieurs précédesseurs et plusieurs successeurs ce qui entraîne un ordre technologique non linéaire pour la tâche.

(5) Un délai est affecté à chaque tâche.

(GH2) : Hypothèses qui peuvent être levées sans difficulté

(1) On considère que les machines peuvent travailler en continu afin de pouvoir effectuer une opération sur deux périodes consécutives sans interruption.

(2) Les temps de transit seront prix égaux à zéro.

(GH3) : Hypothèses qui sont éminemment restrictives

(1) Nous supposons qu'il n'y aura pas d'aléas de fabrication concernant les opérations ordonnancées définitivement sur la période immédiate.

(2) Lorsque deux opérations se succèdent sur une même machine il n'y a pas de temps d'adaptation de la machine.

Dans le chapitre IV nous étudierons la possibilité de supprimer les groupes d'hypothèses (GH2) et (GH3).

III.1.4. Les objectifs

Le critère le plus souvent rencontré dans le cas statique (cf chapitre II) est celui de la minimisation du temps de passage P des tâches dans l'atelier. En fait c'est plutôt la minimisation du temps total d'exécution des tâches D, puisque les tâches sont généralement supposées avoir même date de début au plus tôt. Ce critère n'est pas du tout approprié dans le cas dynamique pour deux raisons. D'une part le processus est ici continu, d'autre part cet objectif n'a plus de sens lorsque des délais sont affectés aux différentes tâches.

Avec les hypothèses précédentes, les seuls critères fondamentaux envisageables pour évaluer un ordonnancement dynamique sont donc les suivants :

Critère des délais

Nous avons introduit l'écart E_i entre la date de fin réelle D_i et la date de fin prévue d_i : $E_i = D_i - d_i$. Le retard de la tâche i est $R_i = \sup(0,E_i)$, son avance est $A_i = \sup(0,-E_i)$.

Si l'on fait l'hypothèse que les moyens de production sont fixés ainsi que les délais il peut ne pas y avoir de solution. Si l'on considère qu'on ne peut agir sur les moyens de production cela entraînera des tâches en retard. Nous verrons quelles conclusions on peut en tirer.

Un premier objectif est donc de minimiser soit le retard moyen $\bar{R}$, soit le nombre de tâches en retard.

Critère des en-cours

Nous avons vu (cf chapitre I) que l'on pouvait mesurer les en-cours par le nombre moyen de tâches en cours de fabrication $\bar{N}$ et que $\bar{N} = \lambda\bar{P}$ (avec $P_i = D_i - c_i$). Or $\bar{P}$, temps moyen de passage, n'est d'aucune utilité lorsque les tâches ont des délais. $\bar{N}$, mesure des en-cours, n'est donc pas un bon critère dans notre cas.

Nous préfèrerons définir le degré d'achèvement de la tâche i dont l'en-cours total est donné par

$$EC_i = \sum_{(j,k)} (D_i - D_{i,j,k})\, f_{i,j,k}$$

où $D_{i,j,k}$ est la date de fin effective de l'opération $O_{i,j,k}$. Tant que la tâche n'est pas commencée son en-cours est nul.

Un second objectif est donc de minimiser l'en-cours global moyen $\overline{EC}$ avec

$$EC = \sum_{i=1}^{n} EC_i$$

La solution optimale, si elle était admissible, consisterait donc à ordonnancer toutes les tâches au plus tard de façon que les délais soient respectés et qu'il n'y ait pas d'attente entre les différentes opérations d'une même tâche.

Critère des moyens de production

L'utilisation moyenne des machines est $\overline{U} = \frac{\lambda \overline{T}}{\overline{M}}$ avec λ le taux d'arrivée des tâches, $\overline{T}$ la durée moyenne d'une tâche et $\overline{M}$ le nombre moyen de machines réparties dans les m groupes

$$\overline{M} = \lim_{t \to +\infty} \frac{1}{t} \left(\sum_{j=1}^{m} \sum_{\theta=0}^{t} M(j,\theta) \right).$$

Si $M(j,\theta)$ est une donnée, $\overline{U}$ est normalement une donnée du problème. Si ce nombre peut être dépassé, $\overline{U}$ devient une variable.

Le problème est donc de savoir si l'on cherchera un ordonnancement qui respecte les $M(j,\theta)$ pour tout j et tout θ (quitte à avoir des dépassements de délais pour certaines tâches) ou inversement qui respecte tous les délais (quitte à violer la contrainte sur la charge des machines).

III.1.5. Les choix possibles

Soit à ordonnancer des tâches sur un intervalle de temps $[T_\alpha, T_\beta]$. La connaissance des tâches évolue dans le temps : donc la date d'arrivée a_i appartient à $[T_\alpha, T_\beta]$ pour i = 1,...,n. L'ordonnancement devra donc être fait périodiquement (période T) à des époques $t_p = pT-t_0$

(p entier). Il ne peut concerner évidemment que le sous-ensemble des n tâches telles que

$$t_{p-1} \leq a_i < t_p \quad \text{(les tâches connues avant } t_p\text{)}$$

ou une partie de ce sous-ensemble si on se limite à réaliser l'ordonnancement de l'intervalle $[pT,(p+1)T]$.

Le problème est de trouver un ordonnancement admissible (A) entièrement déterminé par les dates de début effectives $C_{i,j,k}$ des opérations des différentes tâches. Il n'est pas nécessaire de donner l'affectation des opérations aux machines d'un même groupe. En effet, une opération commencée sur une machine ne se terminera pas sur une autre du même groupe. C'est la conséquence du mode de fixation des $C_{i,j,k}$ et de la relation $D_{i,j,k} = C_{i,j,k} + f_{i,j,k}$. Il suffit de charger, par exemple, la même machine du groupe j avec les couples $(O_{i,j,k}, O_{\ell,j,s})$ tels que $D_{i,j,k} = C_{\ell,j,s}$ ou sinon tels que $C_{\ell,j,s} - D_{i,j,k}$ soit minimum positif. Cette affectation peut comporter plusieurs solutions équivalentes (cf exemple suivant). On sait cependant qu'il existe au moins une solution.

<u>Exemple d'affectation</u>

Soit 11 opérations, dont les dates de début et de fin sont données par le tableau ci-dessous, à répartir sur 3 machines formant un groupe

Opération o	1	2	3	4	5	6	7	8	9	10	11
C_o	0	4	7	14	0	8	10	0	3	9	12
D_o	4	7	13	17	5	10	15	3	8	12	17

D'après la remarque précédente il faudra charger sur la même machine les ensembles d'opérations suivants

$$\alpha = \{1,2,3\},\ \beta = \{8,9,6,7\},\ \gamma = \{10,11\}$$

avec les exclusions

$$\alpha \text{ et } \beta,\ \beta \text{ et } \gamma,\ \alpha \text{ et } \gamma$$

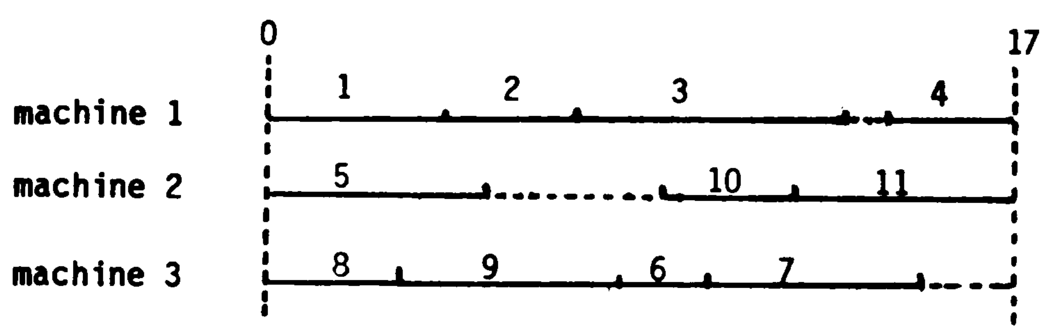
qui impliquent que les 3 machines seront utilisées.

Pour les opérations restantes {4} et {5} il n'y a pas de choix possible à cause des exclusions :

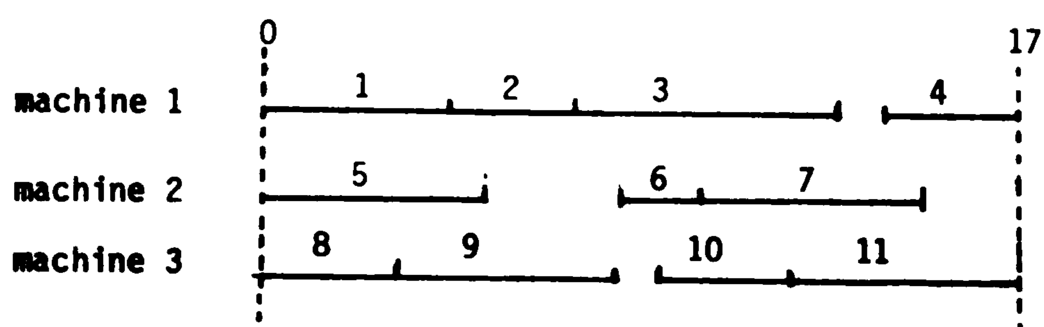
$$5 \text{ et } \alpha,\ 5 \text{ et } \beta,\ \alpha \text{ et } \beta$$

$$\text{puis } 4 \text{ et } \beta,\ 4 \text{ et } \gamma,\ \beta \text{ et } \gamma$$

d'où l'affectation suivante



Une autre solution consisterait à permuter γ et {6,7}, d'où



Définition d'un ordonnancement admissible (A)

Les cinq conditions suivantes doivent être vérifiées.

1 Toute tâche i commence à une date C_i telle que $c_i \leq C_i$.

2 Toute tâche i finit à une date D_i telle que $D_i \leq d_i$.

3 Toute opération ne doit pas être interrompue lorsqu'elle a commencé à être exécutée sur une machine quelconque du groupe concerné.

4 A tout instant θ, la somme des opérations sur le groupe j doit être inférieur ou égal à $M(j,\theta)$.

5 Toute opération $O_{i,j,k}$ doit vérifier avec ses prédécesseurs immédiats $O_{i,r,s}$:

$$C_{i,r,s} + f_{i,r,s} + t(O_{i,r,s},O_{i,j,k}) \leq C_{i,j,k}$$

où $t(O_{i,r,s},O_{i,j,k})$ est le temps de transit.

Trouver un ordonnancement admissible (A) peut être impossible. Les délais des tâches et la composition des groupes de machines sont fixés à l'avance. Il peut être alors impossible de tenir tous les délais (violation de la contrainte 2) ou de respecter la charge maximale des groupes de machine (violation de la contrainte 4).

Dans ce cas, le choix de ne pas respecter 2 ou 4 dépend essentiellement du contexte dans lequel ou recherche l'ordonnancement.

Pour notre part nous nous sommes fixés pour but de respecter absolument la contrainte 4 .

III.2. LE MODELE

III.2.1. Formalisation des conditions

Nous avons vu précédemment que la solution du problème d'ordonnancement était parfaitement définie par la fixation des dates de début effectives $C_{i,j,k}$ de chaque opération.

Sans nuire à la généralité du problème, nous pouvons supposer que toutes les données $(a_i, c_i, d_i, f_i, \ldots)$ sont entières : on peut donc se limiter à la recherche de solutions entières ($C_{i,j,k}$ entier).

Pour la tâche i, posons

$E_{i1} = \{O_{i,j,k} / O_{i,j,k}$ sans précédesseur$\}$

$E_{i2} = \{O_{i,j,k} / O_{i,j,k}$ sans successeur$\}$

$E_{i3} = \{O_{i,j,k} / O_{i,j,k}$ a au moins un prédécesseur ou un successeur$\}$

alors

tâche i $= \{O_{i,j,k} / O_{i,j,k} \in E_{i1} \cup E_{i2} \cup E_{i3}\}$

Les conditions devant être vérifiées pour obtenir un ordonnancement admissible au sens précédent (A) sont :

- Condition 1 $\leftrightarrow$

$$\begin{cases} \forall i \text{ et } \forall (j,k) / O_{i,j,k} \in E_{i1} : c_i \leq C_{i,j,k} \\ \forall i \text{ et } \forall (j,k) / O_{i,j,k} \in E_{i3} : C^-_{i,j,k} \leq C_{i,j,k} \end{cases}$$

avec $C^-_{i,j,k}$ date de début au plus tôt de $O_{i,j,k}$ qui dépend des opérations précédentes

$$C^-_{i,j,k} = \sup_{r,s} (D_{i,r,s})$$

avec r et s tels que $O_{i,r,s} < O_{i,j,k}$ ($D_{i,r,s}$ est la date de fin effective de $O_{i,r,s}$).

- Condition 2 $\longleftrightarrow$

$$\begin{cases} \forall i \text{ et } \forall (j,k)/O_{i,j,k} \in E_{i2} : D_{i,j,k} \leq d_i \\ \forall i \text{ et } \forall (j,k)/O_{i,j,k} \in E_{i3} : C_{i,j,k} \leq C^+_{i,j,k} \end{cases}$$

avec $C^+_{i,j,k}$ date de début au plus tard de $O_{i,j,k}$ qui dépend des opérations suivantes :

$$C^+_{i,j,k} = \operatorname{Inf}_{r,s} (C_{i,r,s}) - f_{i,j,k}$$

avec r et s tels que $O_{i,j,k} < O_{i,r,s}$ ($f_{i,j,k}$ est la durée de fabrication de $O_{i,j,k}$).

- Condition 3 $\longleftrightarrow$

$$\forall i,j,k : D_{i,j,k} = C_{i,j,k} + f_{i,j,k}$$

- Condition 4

θ étant entier, soit $M(j,\theta)$ le nombre de machines du groupe j disponibles pendant l'intervalle de temps $[\theta,\theta+1[$. Si nous posons

$$F_j(\theta) = \{O_{i,j,k}/C_{i,j,k} \leq \theta < D_{i,j,k}\}$$

alors $\quad$ 4 $\longleftrightarrow \begin{pmatrix} \forall j \\ \forall \theta \end{pmatrix} \quad \operatorname{Card}(F_j(\theta)) \leq M(j,\theta)$

Si nous introduisons la variable booléenne suivante

$$J_{i,k}(j,\theta) = \begin{cases} 1 \text{ si } O_{i,j,k} \in F_j(\theta) \\ 0 \text{ sinon} \end{cases}$$

pour poser $J(j,\theta) = \sum_i \sum_k J_{i,k}(j,\theta)$, $J(j,\theta)$ représente la charge du groupe j de machine sur l'intervalle $[\theta,\theta+1[$.

Alors on peut en déduire l'équivalence

$$\text{Condition } 4 \Longleftrightarrow J(j,\theta) \leq M(j,\theta) \quad \begin{cases} \forall j = 1,\ldots,m \\ \forall \theta \text{ entier} \end{cases}$$

- Condition 5

Nous n'envisageons pour l'instant que des temps de transit entre opérations nuls. Nous le faisons simplement dans le but de simplifier les notations.

Alors 5 $\Longleftrightarrow$

$$\forall i \text{ et } \forall (j,k)/(O_{i,j,k} \notin E_{i1} \text{ et } \forall (r,s)/O_{i,r,s} < O_{i,j,k}) :$$

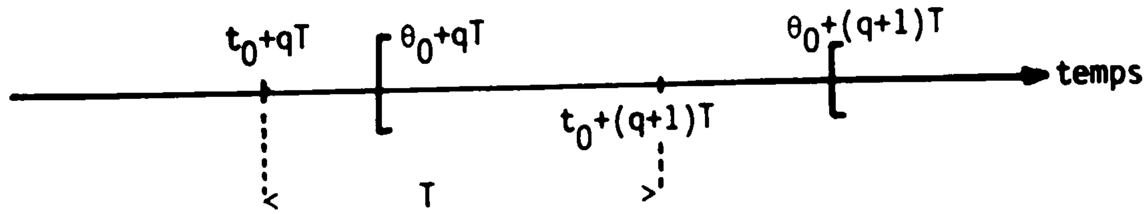
$$C_{i,r,s} + f_{i,r,s} \leq C_{i,j,k}$$

III.2.2. Période et unité de temps

Nous avons vu précédemment que l'arrivée des tâches dans l'atelier implique un ordonnancement effectué périodiquement. La résolution du problème global sur l'intervalle $[T_\alpha,T_\beta]$ peut donc se décomposer en la résolution de p sous-problèmes sur des intervalles d'amplitude T. T est aussi la période avec laquelle les sous-problèmes sont résolus.

A des dates $t_0, t_0+T, \ldots, t_0+qT, \ldots, t_0+(p-1)T$, il faut fixer définitivement l'ordonnancement des opérations sur les intervalles respectifs (avec $\theta_0 > t_0$)

$$[\theta_0, \theta_0+T[, [\theta_0+T, \theta_0+2T[, \ldots, [\theta_0+qT, \theta_0+(q+1)T[, \ldots, [\theta_0+(p-1)T, \theta_0+pT[.$$



A la date t_0+qT les tâches concernées par l'ordonnancement de $[\theta_0+qT, \theta_0+(q+1)T[$ sont telles que

a) $a_i < t_0+qT$: la tâche i est connue et non entièrement ordonnancée.

b) $\theta_0+qT \leq c_i < \theta_0+(q+1)T$ ou bien elles ont des opérations déjà ordonnancées définitivement sur un intervalle antérieur.

La seule hypothèse que nous faisons sur la période T est qu'elle est entière. Il suffit pour cela de choisir une unité de temps convenable. Nous supposerons, en outre, que toutes les données du problème peuvent s'exprimer par des entiers dans cette unité de temps. Ces hypothèses n'entraînent en aucun cas une perte de généralité de notre modèle.

Le choix de la période T dépend essentiellement du fonctionnement de l'atelier. En particulier ce choix est lié aux types de tâches, à la durée moyenne de ces tâches, aux délais accordés. Il est aussi fonction de la fréquence d'arrivée des tâches dans l'atelier. A chaque cas s'impose une période qu'il n'est pas possible ni nécessaire de préciser davantage puisqu'elle dépend de chaque cas et sera avant tout dictée par le bon sens.

III.2.3. Objectifs et principes du modèle

Le principe du modèle a été défini en tenant compte que l'on ignore s'il existe une solution admissible (A) au problème et donc a fortiori une solution optimale. Néanmoins il est possible d'énoncer quelques conditions nécessaires d'admissibilité qui ne sont malheureusement pas suffisantes ce qui diminue leur intérêt.

Nous nous limitons donc à la recherche d'une solution convenable respectant la contrainte 4 sur les moyens de production. Certains délais peuvent ne pas être tenus : la contrainte 2 pourra être violée pour certaines tâches. De plus, à l'aide de prises de décisions logiques, nous essaierons d'atteindre les objectifs tels que montant réduit des en-cours et bonne utilisation des machines. Le but du modèle est ainsi de tendre vers un ordonnancement admissible.

Les p sous-problèmes ne sont pas indépendants. Cette relation est particulièrement perçue au niveau du respect des délais. En effet, lors de la résolution du $q+1^{ième}$ sous-problème interviennent les tâches déjà connues (arrivées avant $t_0+(q-1)T$) et celles qui sont arrivées sur l'intervalle $[t_0+(q-1)T, t_0+qT[$. Or la résolution des q sous-problèmes précédents a pu tenir compte des tâches déjà connues, par contre elle ignorait tout des arrivées futures. La manière dont a été conduite la résolution de ces q sous-problèmes influera donc sur la solution, en particulier au niveau de la tenue des délais.

Pour cette raison nous avons choisi de décomposer la résolution d'un sous-problème en deux étapes appelées "chargement" et "régulation". A la date t_0+qT on ordonnance définitivement des opérations sur l'intervalle $[\theta_0+qT, \theta_0+(q+1)T[$ (chargement et régulation). Les périodes suivantes sont ordonnancées provisoirement (chargement uniquement). Nous avions envisagé de nous limiter à l'ordonnancement définitif de la période immédiate. Mais cela entraînait des difficultés pour sélectionner, parmi l'ensemble des opérations possibles, celles à ordonnancer sur cet intervalle. Le problème de la connaissance à l'avance du respect des dates de fin au plus tard se posait à nouveau.

A l'évidence l'ordonnancement idéal du point de vue respect des délais et montant des en-cours (mais ne respectant pas nécessairement la contrainte (4)) est celui pour lequel toutes les opérations $O_{i,j,k}$ vérifient :

$$C_{i,j,k} = C^{+}_{i,j,k} - f_{i,j,k}$$

La méthode de chargement et la régulation que nous proposons tiennent compte de cette remarque en ordonnançant les tâches à partir de leur cadrage minimal au plus tard.

Le chargement des machines utilise un système de priorités dynamiques suivant un algorithme heuristique inspiré de [5].

III.2.4. Etude du chargement

Le mode de chargement des opérations sur les machines va être impliqué de manière prépondérante par le choix du cadrage initial de chaque opération de chaque tâche.

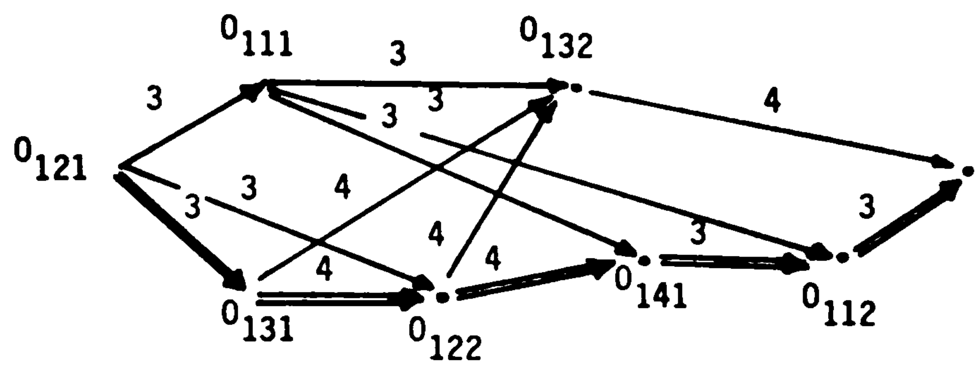
III.2.4.1. Discussion sur le cadrage initial

Pour chaque tâche i on connaît, à partir de l'ordre technologique des opérations, le graphe associé non cadré dans le temps. La durée minimale de la tâche peut être obtenue par un PERT-potentiels. Alors toute opération n'appartenant pas au chemin critique peut être cadrée au plus tôt ou au plus tard.

Exemple :

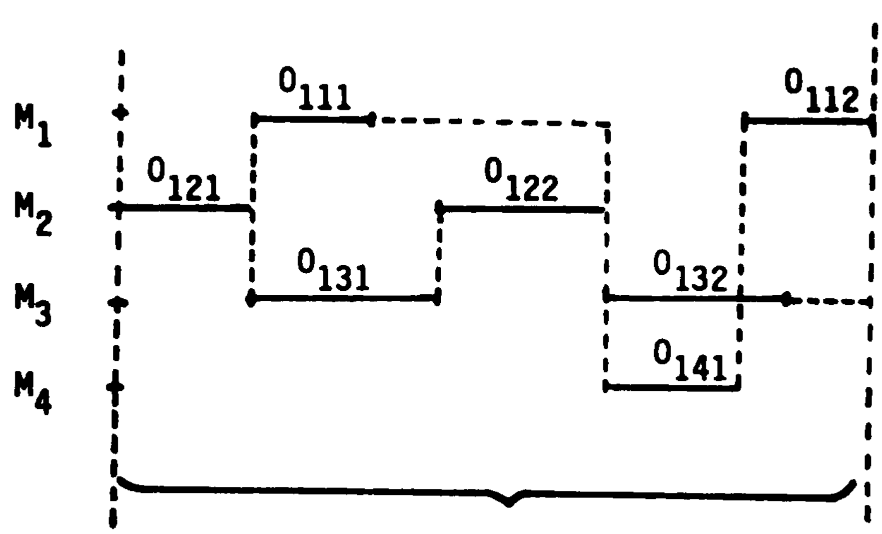
4 machines distinctes/1 tâche/7 opérations.

Le graphe associé est le suivant :



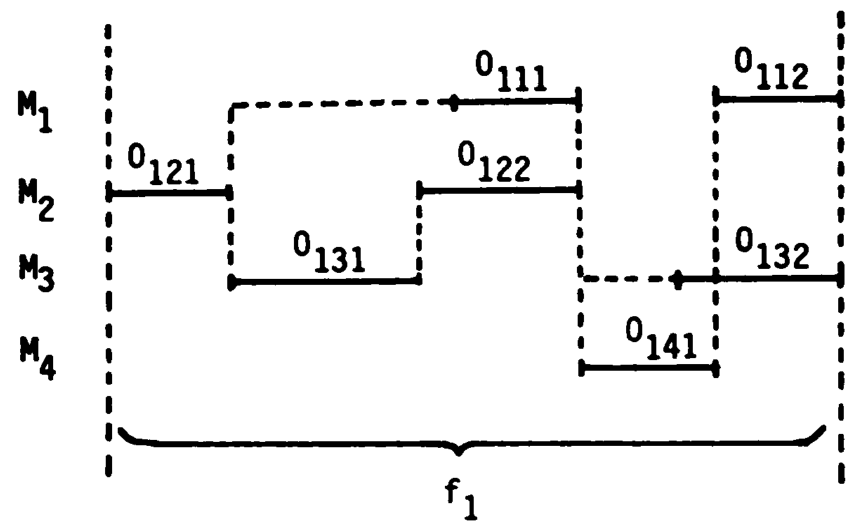
Le chemin critique figurant en trait double, les opérations O_{111} et O_{132} peuvent être au plus tôt ou au plus tard, ce qui donne les diagrammes suivants :

Diagramme au plus tôt :



f_1 = durée minimale de la tâche 1

Diagramme au plus tard :



Si f_i est la durée (minimale) de fabrication de la tâche i, la marge totale de la tâche i est : $d_i - c_i - f_i$.

Pour une opération $O_{i,j,k}$ nous distinguerons :

<u>La marge amont</u>, durée totale dont on peut l'avancer sans déborder en amont de la date de début au plus tôt c_i de la tâche i.

<u>La marge aval</u>, durée totale dont on peut la retarder sans dépasser la date de fin au plus tard d_i de la tâche i.

<u>La distance amont</u> égale à $\text{Inf}_{r,s}$ (durée séparant $O_{i,j,k}$ et $O_{i,r,s}$) avec r et s tels que $O_{i,r,s} < O_{i,j,k}$.

<u>La distance aval</u> égale à $\text{Inf}_{r,s}$ (durée séparant $O_{i,r,s}$ et $O_{i,j,k}$) avec r et s tels que $O_{i,j,k} < O_{i,r,s}$.

Les marges nous permettront de déterminer $C^-_{i,j,k}$ et $C^+_{i,j,k}$ lorsqu'on chargera l'opération $O_{i,j,k}$. Les distances serviront à fixer l'intervalle sur lequel il est possible d'ordonnancer une opération sans influer sur l'ordonnancement des opérations de la même tâche qui l'encadrent.

Il faut cadrer ce diagramme minimal dans le temps. Deux possibilités existent pour ce choix de la répartition des marges. Ce sont la marge totale en aval et la marge totale en amont qui correspondent respectivement aux cadrages au plus tôt et au plus tard. La troisième possibilité, marge totale uniformément répartie entre les diverses opérations de la tâche, paraît sans intérêt puisque son seul effet est de restreindre les intervalles $[C^-_{i,j,k},C^+_{i,j,k}]$.

Il nous a paru préférable de choisir un cadrage minimum initial au plus tard. Cela permet de prendre en compte plus facilement les délais. De plus, le montant d'en-cours initial est minimal. Un autre avantage de ce choix se situe du point de vue de la charge des machines. Celle-ci sera globalement moins élevée pour le très court terme, d'où la possibilité de charger plus facilement des opérations arrivées plus tard avec des marges faibles. Par contre, on risque une sous-utilisation des machines. Pour y pallier, nous avons prévu une "régulation" de la charge pour la période ordonnancée définitivement.

III.2.4.2. Mode de chargement

D'après le choix précédent, (partir pour chaque tâche de son cadrage au plus tard, avec de plus toutes les opérations cadrées au plus tard) il paraît judicieux de réaliser le chargement d'une tâche en commençant par les opérations sans successeurs, et en poursuivant par les opérations dont tous successeurs sont ordonnancés.

Considérons l'ordonnancement effectué à la date t_0+pT. Soit E l'ensemble des opérations concernées par le chargement à cette époque là : ce sont les opérations arrivées depuis l'ordonnancement précédent.

$$E = \{O_{i,j,k}/t_0+(p-1)T \leq a_i < t_0+pT\}$$

Le chargement consiste alors à fixer provisoirement pour toute opération $O_{i,j,k} \in E$ une date de début $C_{i,j,k}$ telles que les conditions d'admissibilité (A) soient vérifiées sauf éventuellement 2 .

Soit $E_0 = \{O_{i,j,k} \in E / O_{i,j,k}$ sans successeurs non ordonnancés$\}$

$E_0 = \bigcup_j E_{0,j}$ avec $E_{0,j}$ le sous-ensemble concernant le groupe j de machines. Achaque itération on fixe un $C_{i,j,k}$ d'une opération $O_{i,j,k}$ appartenant à E_0. L'ensemble E_0 est alors modifié ainsi :

$$E_0 = E_0 - \{O_{i,j,k}\} + \{\text{pred } (O_{i,j,k})\}$$

où pred $(O_{i,j,k})$ désigne l'ensemble des prédécesseurs immédiats de $O_{i,j,k}$ pour lesquels tous les successeurs sont déjà chargés.

Le chargement se termine lorsque $E_0 = \emptyset$.

Remarque 1

Lors d'une itération, un ou plusieurs autres $C_{i,j,k}$ peuvent être modifiés. Nous en donnons l'explication dans la suite.

Remarque 2

La modification de E_0 aurait pu se traduire par

$$E_0 = E_0 - \{O_{i,j,k}\} + \{\text{pred}(O_{i,j,k})\} + \{\text{dep}(O_{i,j,k})\}$$

où dep$(O_{i,j,k})$ représente l'ensemble des opérations que l'on a choisi de déplacer parce que sans cela $O_{i,j,k}$ ne peut être chargée sur l'intervalle $[C^-_{i,j,k}, C^+_{i,j,k} + f_{i,j,k}]$.

Voyons maintenant comment est choisie dans E_0 l'opération $O_{i,j,k}$ pour laquelle nous fixons provisoirement une date de début. Soit

$E_{0,j}$ le sous-ensemble de E_0 formé des opérations à charger sur le groupe de machines j. Nous allons raisonner sur $E_{0,j}$. Ensuite nous étendrons le choix à E_0. Cela se fera en choisissant l'ordre de prise en compte des divers groupes.

La solution initiale consistant à ordonnancer $E_{0,j}$ au plus tard donne la charge du groupe j par $\sum_{\theta} J(j,\theta)$. Nous pouvons déjà, à partir de cette charge, dire s'il y aura des opérations en retard (condition nécessaire de faisabilité sans retard).

Soit j fixé, pour tout i et tout k tel que $O_{i,j,k} \in E_{0,j}$ il faut fixer $C_{i,j,k}$ vérifiant $C^-_{i,j,k} \leq C_{i,j,k} \leq C^+_{i,j,k}$. Supprimons l'opération $O_{i,j,k}$ de cet ordonnancement initial. Si cette opération était chargée en dernier (la dernière de $E_{0,j}$) il sera impossible de trouver un ordonnancement admissible (A) si les conditions (1) ou (2) suivantes sont vérifiées :

$$M_j(C^+_{i,j,k}-C^-_{i,j,k}) < \sum_{\theta = C^-_{i,j,k}}^{C^+_{i,j,k}+f_{i,j,k}} J(j,\theta)-(M_j-1)\, f_{i,j,k} \qquad (1)$$

$$\sum_{\theta = C^-_{i,j,k}}^{C^+_{i,j,k}+f_{i,j,k}} M(j,\theta) < \sum_{\theta = C^-_{i,j,k}}^{C^+_{i,j,k}+f_{i,j,k}} J(j,\theta)+f_{i,j,k} \qquad (2)$$

(1) est le cas où les M_j machines du groupe j sont continuellement disponibles sur $[C^-_{i,j,k}, C^+_{i,j,k} + f_{i,j,k}]$ et (2) correspond au cas où la composition du groupe j varie dans le temps $(M(j,\theta))$.

Exemples

a) Une machine continuellement disponible ($M_j = 1$)

$C^+_{i,j,k}+f_{i,j,k}$

$C^-_{i,j,k}$ $\qquad$ $C^+_{i,j,k}$ $\qquad$ $f_{i,j,k} = 2$

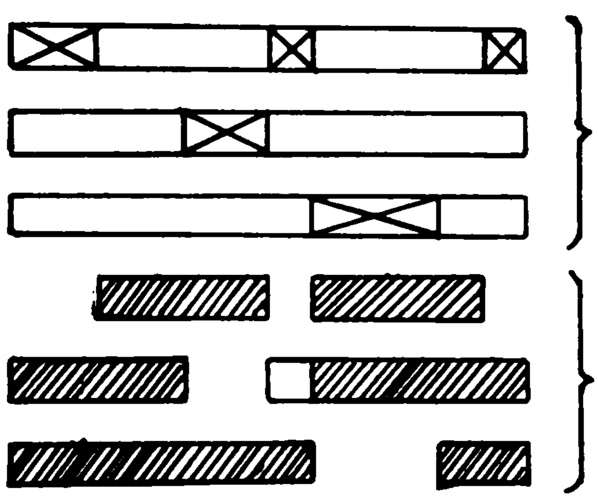
$C^+_{i,j,k} - C^-_{i,j,k} = 10$ $\qquad$ $\sum_{\theta = C^-_{i,j,k}}^{C^+_{i,j,k}+f_{i,j,k}} J(j,\theta) = 11$

(1) est vérifiée $\Longrightarrow$ ordonnancement non admissible avec $O_{i,j,k}$

b) Trois machines non continuellement disponibles

$C^+_{i,j,k}+f_{i,j,k}$

$C^-_{i,j,k}$ $\qquad$ $C^+_{i,j,k}$ $\qquad$ $f_{i,j,k} = 2$



$\sum_{\theta = C^-_{i,j,k}}^{C^+_{i,j,k} + f_{i,j,k}} M(j,\theta) = 27$

$\sum_{\theta = C^-_{i,j,k}}^{C^+_{i,j,k} + f_{i,j,k}} J(j,\theta) = 26$

(2) est vérifiée $\Longrightarrow$ ordonnancement non admissible avec $O_{i,j,k}$

La charge $\sum_{C^-}^{C^+ +f} J(j,\theta)$ est modifiée après le chargement de chaque opération.

Cependant les quantités précédentes ne permettent pas de savoir si l'ordonnancement partiel est admissible. C'est pourquoi nous allons introduire une notion supplémentaire : la "liberté de manoeuvre".

Pour pouvoir tenir compte à la fois de la durée de l'opération et s'assurer que l'opération peut respecter la contrainte de charge des machines 4 , nous évaluerons la "liberté de manoeuvre" $\ell_{i,j,k}$ $O_{i,j,k}$ ainsi :

$$\ell_{i,j,k} = \sum_{\alpha = C^-_{i,j,k}}^{C^+_{i,j,k}} e_\alpha \text{ avec } e_\alpha = \begin{cases} 1 \text{ si } J(j,\theta) < M(j,\theta) \\ \quad \forall\theta \in [\alpha, \alpha+f_{i,j,k}] \\ 0 \text{ sinon} \end{cases}$$

$\ell_{i,j,k}$ peut être interprétée comme le nombre d'emplacements possibles pour l'opération $O_{i,j,k}$.

Alors $O_{i,j,k} \ll O_{\ell,j,p}$ si et seulement si $\ell_{i,j,k} < \ell_{\ell,j,p}$.

<u>Sur le groupe j, on ordonnancera donc d'abord $O_{i,j,k}$ telle que la liberté de manoeuvre $\ell_{i,j,k}$ associée soit minimum.</u>

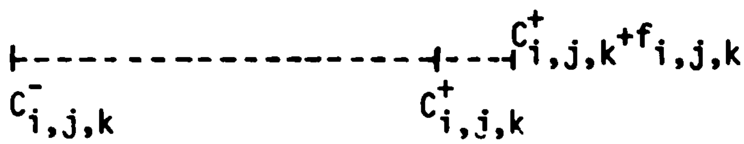
<u>Cas où $\ell_{i,j,k} > 0$</u>

L'ordonnancement partiel est admissible au sens (A). Si $\ell_{i,j,k} > 1$, plusieurs emplacements sont possibles pour $O_{i,j,k}$: on choisit celui qui donne à $C_{i,j,k}$ la valeur la plus grande (chargement au plus tard).

Les deux exemples précédents montrent un ordonnancement initial admissible sans $O_{i,j,k}$ et non admissible avec $O_{i,j,k}$.

Il faut remarquer en effet que les conditions (1) ou (2) ne sont pas des conditions suffisantes d'impossibilité.

Exemple



$f_{i,j,k} = 2$

1 machine continuellement disponible

$C^+_{i,j,k} - C^-_{i,j,k} = 10$

$$\sum_{\theta = C^-_{i,j,k}}^{C^+_{i,j,k}+f_{i,j,k}} J(j,\theta) = 9$$

La condition (1) n'est pas vérifiée et pourtant l'ordonnancement est non admissible.

Dans $E_{0,j}$ nous allons, en tenant compte de ce qui précède, fixer provisoirement les $C_{i,j,k}$ des opérations dans l'ordre suivant :

Définition

L'opération $O_{i,j,k}$ sera dite prise en compte avant l'opération $O_{\ell,j,p}$ et nous noterons $O_{i,j,k} << O_{\ell,j,p}$ si et seulement si

$$\sum_{\theta = C^-_{i,j,k}}^{C^+_{i,j,k}+f_{i,j,k}} M(j,\theta) - \sum_{\theta = C^-_{i,j,k}}^{C^+_{i,j,k}+f_{i,j,k}} J(j,\theta) - f_{i,j,k}$$

$$< \sum_{\theta = C^-_{\ell,j,p}}^{C^+_{\theta,j,p}+f_{\ell,j,p}} M(j,\theta) - \sum_{\theta = C^-_{\ell,j,p}}^{C^+_{\ell,j,p}+f_{\ell,j,p}} J(j,\theta) - f_{\ell,j,p}$$

Cas où $\ell_{i,j,k} = 0$

Cela correspond au fait qu'il est impossible de charger $O_{i,j,k}$ en respectant la contrainte des délais 2 . Deux cas peuvent se présenter :

- S'il est possible d'avancer d'autres opérations déjà chargées (l'ordonnancement systématique au plus tard pouvant éventuellement le permettre) : on envisage tous les déplacements possibles vers l'amont des opérations chargées sur $[C^-_{i,j,k}, C^+_{i,j,k} + f_{i,j,k}]$ en positionnant $O_{i,j,k}$ successivement à tous les emplacements possibles jusqu'à l'obtention d'une solution. La solution retenue est celle qui entraîne une augmentation minimum des en-cours. Il faut remarquer que ces déplacements vers l'amont ne concernent pas seulement des opérations du groupe j.

- S'il n'y a pas de solution par déplacements, l'opération $O_{i,j,k}$ sera en retard (ceci implique que la tâche concernée sera en retard). Alors on ordonnance $O_{i,j,k}$ au plus tôt après $d_{i,j,k}$, date fin au plus tard de cette opération.

Ayant précisé le choix dans $E_{0,j}$ pour j fixé, il reste à spécifier l'ordre suivant lequel les différents groupes de machines sont envisagés. Nous avons choisi de procéder suivant l'ordre décroissant de la charge, du groupe le plus contraignant au groupe qui l'est le moins. Nous considérons la charge locale de chaque groupe, c'est-à-dire

pour toute $O_{i,j,k} \in E_0$ la charge du groupe j

sur l'intervalle $[C^-_{i,j,k}, C^+_{i,j,k} + f_{i,j,k}[$

Cela revient à calculer les $\ell_{i,j,k}$ pour toute $O_{i,j,k} \in E_0$ (donc pour tout j) et à choisir de charger le groupe j de machine avec $O_{i,j,k} \in E_0$ telle que $\ell_{i,j,k}$ est minimum.

$F_p \cup E_p = \{O_{i,j,k}/C_{i,j,k} \in I_p\}$ tel que l'ordonnancement de I_p soit admissible et maximise l'utilisation des machines $U(I_p)$.

Nous utilisons, pour résoudre le problème R_p, une méthode de pénalisation inspirée de [4] mais utilisée de façon sensiblement différente. En effet, nous nous imposons d'aboutir à une solution admissible (condition 4 vérifiée), et ce faisant de maximiser l'utilisation des moyens de production.

III.2.5.2. Principe de la régulation

Il s'agit de résoudre le problème R_p précédemment défini. Les opérations appartenant à E_p sont supposées ordonnancées définitivement. Il faut alors combler les "trous" restants avec certaines des opérations candidates ($\in$ A) de telle sorte que U soit maximale. Nous allons voir que l'algorithme est relativement peu combinatoire car les contraintes sont assez sévères.

$$A = \{O_{i,j,k}/D_{i,j,k} \in I_q \text{ avec } q > p, \text{ et } C^-_{i,j,k} < \theta_0+(p+1)T\}$$

Posons alors

$$A_1 = \{O_{i,j,k}/O_{i,j,k} \text{ sans prédécesseur dans A}\}$$

$$A_2 = \{O_{i,j,k}/O_{i,j,k} \text{ sans précédesseur dans } A - A_1\} \text{ etc...}$$

Initialisation

On ordonnance les opérations appartenant à A_1, puis à A_2, etc..., cadrées au plus tôt, sur les intervalles libres de I_p sans tenir compte de la contrainte sur la limitation des moyens (4). L'ordre envisagé ici est celui des θ croissants. On entend par intervalle libre de I_p :

tout intervalle $[\theta_1,\theta_2[$ tel que pour j fixé

$$J(j,\theta) < M(j,\theta) \qquad \forall\theta \in [\theta_1,\theta_2[$$

ceci après le chargement des opérations de E_p et avant celui des opérations de A.

Il faut remarquer que seulement un sous-ensemble A_0 de A est envisagé puisque les durées $f_{i,j,k}$ des opérations interviennent comme contraintes lors de cet ordonnancement initial.

<u>Résolution de R_p</u>

Posons $n(j,\theta) = \sup\,(J(j,\theta) - M(j,\theta),0)$

et soit $$N = \sum_j \sum_{\theta \in I_p} n(j,\theta)$$

Alors la condition 4 d'admissibilité est équivalente à

$N = 0$ pour le problème R_p.

Pour chaque opération $O_{i,j,k}$ de A_0 on essaie toutes les positions possibles sur $[C^-_{i,j,k},\theta_0+(p+1)T[$. On garde la position pour laquelle N est minimum. En cas d'indétermination on garde la position au plus tôt.

On obtient alors une nouvelle solution et donc une nouvelle valeur de N. Si deux itérations consécutives donnent la même valeur pour N on choisit d'éliminer une des opérations c'est-à-dire qu'elle revient à sa place initiale avec $C_{i,j,k} > \theta_0+(p+1)T$. L'opération éliminée est celle qui induit la diminution minimale de l'utilisation. (Voir exemple dans le chapitre IV).

On itère ainsi jusqu'à obtenir N = 0, i.e. la contrainte 4 est vérifiée. L'ensemble des opérations restantes (celles qui n'ont pas été éliminées) est l'ensemble cherché F_p.

III.2.6. Algorithme général

L'algorithme général de notre modèle est donc le suivant :

Ordonnancement dynamique

$p \leftarrow 0$
Répéter
    Réaliser l'ordonnancement à la date t_0+pT ;
    $p \leftarrow p+1$
jusque (fin d'arrivée des tâches)

Ordonnancement à la date t_0+pT

$E_0 \leftarrow E_2(t_0+pT)$
Tant que ($E_0 \neq \emptyset$) faire
    début
        Calculer les $\ell_{i,j,k}$ pour tout $O_{i,j,k} \in E_0$;
        Ordonnancer $O_{i,j,k}$ correspondant à $\mathrm{Min}(\ell_{i,j,k})$;
        $E_0 \leftarrow E_0 - \{O_{i,j,k}\} + \{\mathrm{pred}(O_{i,j,k})\}$
    fin
Régulation de la période $[\theta_0+pT, \theta_0+(p+1)T[$

La régulation sera illustrée sur un exemple simple mais caractéristique dans le chapitre IV.

Notre modèle d'ordonnancement dynamique exposé dans ce chapitre, permet donc d'avoir une approche plus réaliste des problèmes d'ordonnancement tels qu'ils se posent concrètement. Les trois caractéristiques principales sont les suivantes : la connaissance des tâches qui évolue dans le temps, la structure non linéaire des tâches et les délais affectés aux différentes tâches.

Chapitre IV

Applications et Résultats

Dans ce chapitre, nous ne donnons que très brièvement les principes de l'implémentation de notre modèle, en renvoyant le lecteur à l'annexe de la thèse où est présentée la totalité de notre programme.

Dans un second paragraphe nous donnons des exemples de résolution par notre méthode avec des résultats commentés.

Le principe de régulation est illustré par un exemple simple et, dans une dernière partie, nous discutons la possibilité de lever certaines des hypothèses restrictives de notre modèle.

IV. 1. IMPLEMENTATION DU MODELE

Notre modèle d'ordonnancement dynamique qui est exposé dans le chapitre III a été implémenté sur l'ordinateur IRIS-80 CII-HB du C.I.C.T. La programmation a été réalisée en langage PL/1 en utilisant la caractéristique particulièrement importante des variables dynamiques pour la structuration des données. Ce choix ne garantit pas la meilleure efficacité du programme du point de vue temps d'exécution. Il nous a paru cependant préférable d'utiliser les listes chainées afin de rendre aussi générale que possible la représentation de la structure des tâches non linéaires.

Les caractéristiques des problèmes traités sont les suivantes : la période d'ordonnancement choisie est le jour. Nous avons pris l'heure comme unité de temps. Les machines sont supposées pouvant travailler sur toute la période (i.e. 24 heures par jour). Dans l'état actuel du programme nous nous limitons donc à envisager les cas où $M(j,\theta) = 1$ pour tout j et tout θ.

Toutes les données sont générées aléatoirement. Ce sont, rappelons le :

- l'ordre technologique des tâches représenté par un graphe quelconque

- les dates d'arrivée des tâches dans l'atelier exprimées en jours

- les délais accordés aux tâches exprimés aussi en jours
- les durées des opérations en nombre entier d'heures.

Pour les détails de l'implémentation nous renvoyons le lecteur à l'annexe de cette thèse où se trouve réuni l'ensemble des modules de notre programme.

IV.2. EXEMPLES RESOLUS ET RESULTATS OBTENUS

Nous avons généré une série de problèmes pour tester notre modèle. Tous concernent 10 périodes successives (10 jours). Des tâches arrivent dans l'atelier à chaque période (chaque jour). La date de début au plus tôt de ces tâches est supposée être le début de la période suivante (le lendemain).

La charge initiale des machines a été fixée pour les cinq premières périodes. Afin de tenir compte d'un certain "régime permanent" nous avons supposé que des tâches avaient été chargées avant le début d'ordonnancement des 10 jours successifs. Pour chaque machine, l'initialisation est la suivante :

Période	charge initiale (en heures)
1	10
2	8
3	6
4	4
5	2

Le programme correspondant au modèle a été testé avec succès sur des ateliers comportant de 2 à 9 machines. Le tableau 1 donne un extrait des problèmes testés sur 10 périodes. Pour chacun nous précisons le nombre total des tâches à ordonnancer, le nombre total d'opérations ainsi que la charge moyenne des machines.

TABLEAU 1

numéro du problème	nombre de machines	nombre de tâches	nombre d'opérations	charge moyenne des machines	charge moyenne de la machine la moins chargée	charge moyenne de la machine la plus chargée
1	2	43	155	16,50	16,14	16,86
2	2	43	133	15,32	15,29	15,36
3	3	28	100	18,69	16,29	20,29
4	3	18	153	17,90	16,29	20,14
5	3	31	409	19,48	21,71	23,00
6	4	60	343	18,57	17,36	19,50
7	4	60	343	24,70	23,14	25,79
8	5	39	259	22,28	21,31	23,85
9	5	45	332	18,24	16,36	20,51
10	7	44	340	13,30	11,29	14,57
11	7	44	340	16,77	14,21	18,71
12	7	51	385	19,21	18,75	21,06
13	8	44	169	15,07	13,86	17,86
14	8	58	255	14,28	12,89	16,32
15	9	36	372	15,14	13,36	17,71
16	9	55	323	13,47	11,57	15,29

TABLEAU 2

numéro du problème	Durée moyenne par période	nombre de tâches en retard	Utilisation moyenne des machines	retard moyen
1	16,14	0	18,50	0
2	14,76	0	18,56	0
3	7,14	2	19,96	13,50
4	7,56	3	17,15	8,67
5	31,50	1	21,33	12
6	49,80	3	20,58	6,33
7	65,52	11	21,89	26,18
8	26,28	3	19,64	25
9	31,06	0	17,96	0
10	58,44	0	13,40	0
11	51,60	1	17,10	5
12	67,26	4	19,05	14,57
13	40,42	0	17,56	0
14	48,87	0	15,98	0
15	39,48	0	17,28	0
16	78,80	0	15,89	0

Le tableau 2 montre les résultats obtenus pour ces 16 problèmes. Les temps nécessaires à l'obtention d'une solution sont exprimés en secondes d'U.C. Ils sont relatifs à une période. Comme on le voit sur ce tableau 2, ils dépendent essentiellement du nombre de machines dans l'atelier, du nombre d'opérations à traiter et de la charge des machines sur la période considérée. Il faut noter que ces résultats concernent l'ordonnancement définitif (chargement et régulation) de 10 jours consécutifs pour chaque problème ainsi que le seul chargement pour les jours suivants. L'utilisation des machines (tableau 2) s'applique aux 10 premiers jours. C'est pourquoi il apparait une disparité avec la charge moyenne donnée dans le tableau 1 : cette dernière tient compte des délais des tâches.

Afin de voir l'influence de la charge des machines sur les temps d'obtention d'une solution nous avons généré deux séries de problèmes comportant 5 et 8 machines. Les données font l'objet du tableau 3. Dans ces deux cas nous n'avons fait varier que la durée des opérations, les autres données restant inchangées. Les résultats obtenus sont consignés dans le tableau 4. Nous y avons fait figurer le nombre de tâches traitées définitivement, c'est-à-dire les tâches dont les opérations sont toutes ordonnancées sur les 10 premières périodes. La variation de ce nombre pour les différentes problèmes peut justifier en partie les variations des temps d'exécution autres que celles impliquées par la charge des machines. En effet, le chargement est moins coûteux en temps que la régulation.

TABLEAU 3

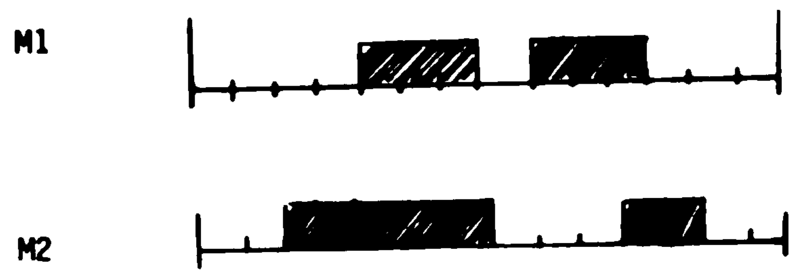
numéro du problème	nombre de machines	nombre de tâches	nombre d'opérations	charge moyenne des machines	charge moyenne de la machine la moins chargée	charge moyenne de la machine la plus chargée
8	5	39	259	22,28	21,31	23,85
17	5	39	259	22,75	21,69	24,69
18	5	39	259	18,77	17,92	20,31
19	5	39	259	14,95	13,31	16,15
13	8	44	169	15,07	13,86	17,86
20	8	44	169	16,33	15,00	19,79
21	8	44	169	12,94	12,00	14,79
22	8	44	169	17,68	16,00	21,50

TABLEAU 4

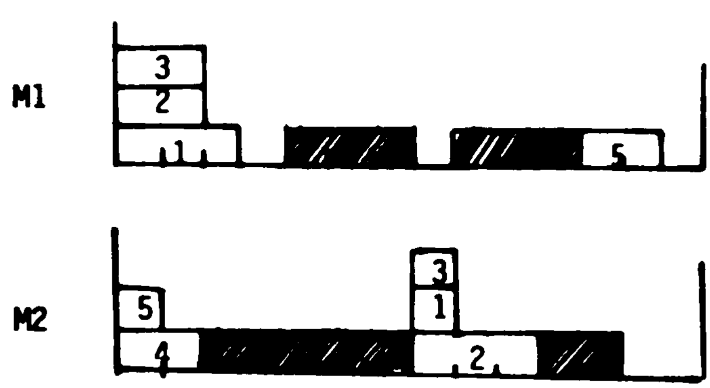
numéro du problème	durée moyenne par période	nombre de tâches en retard	utilisation moyenne des machines	retard moyen	nombre de tâches traitées définitivement
8	26,28	3	19,64	25	21
17	28,26	7	19,91	10,29	19
18	24,96	2	18,60	8	22
19	27,06	0	15,33	0	23
13	40,42	0	17,56	0	29
20	36,72	0	18,21	0	27
21	31,56	0	14,21	0	29
22	33,00	4	19,53	12,50	24

IV.3. EXEMPLE SIMPLE DE REGULATION

Pour illustrer notre modèle nous donnons ci-dessous un exemple simple du principe de la régulation. Cet exemple concerne deux machines M1 et M2 et une période de 14 unités de temps. Après chargement nous obtenons une charge égale à 6 pour M1 et 7 pour M2 suivant le diagramme :



Les opérations chargées sur cette période immédiate sont supposées l'être définitivement. Afin de maximiser l'utilisation des machines il faut chercher à avancer des opérations chargées provisoirement sur des périodes postérieures. Les opérations sont représentées sur le diagramme qui suit :



(les numéros figurant sur ce diagramme sont ceux des tâches)

N = 7

Les contraintes technologiques sont les suivantes :

$$O_{1,1} < O_{1,2}$$

$$O_{2,1} < O_{2,2}$$

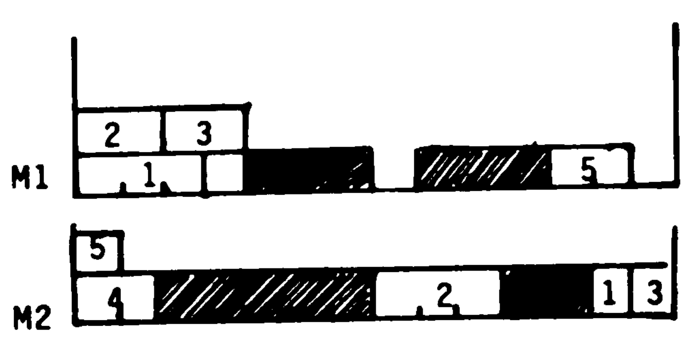
$$O_{3,1} < O_{3,2}$$

$$O_{5,2} < O_{5,1}$$

Il est évident que ce diagramme ne représente pas un ordonnancement admissible puisque la charge des machines n'est pas respectée. Ici N = 7 (cf. chapitre III page 87).

L'algorithme de régulation, exposé dans le précédent chapitre, s'applique ici de la manière suivante . (Nous appelons "écrétage" le calcul de la nouvelle valeur de N en essayant toutes les positions possibles pour chaque opération et en gardant celle qui donne à N la valeur minimale. "Elimination" est le choix de l'opération qui va revenir à sa place initiale).

Ecrétage (1)



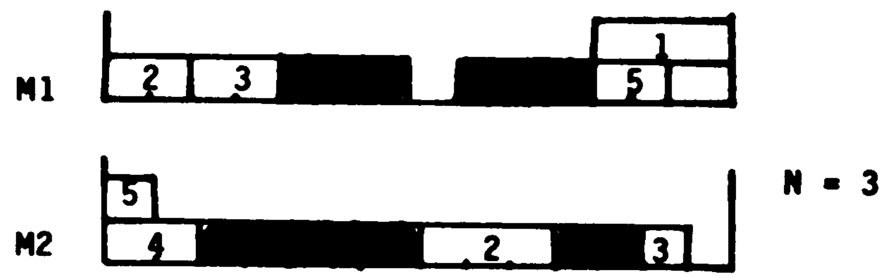
N = 4

Elimination (1)

Opérations sans successeurs	$O_{1,2}$	$O_{3,2}$	$O_{2,2}$	$O_{5,1}$	$O_{4,2}$
Variation de l'utilisation = ΔU	0	-1	-2	-1	-1

L'opération $O_{1,2}$ est éliminée

Ecretage (2)

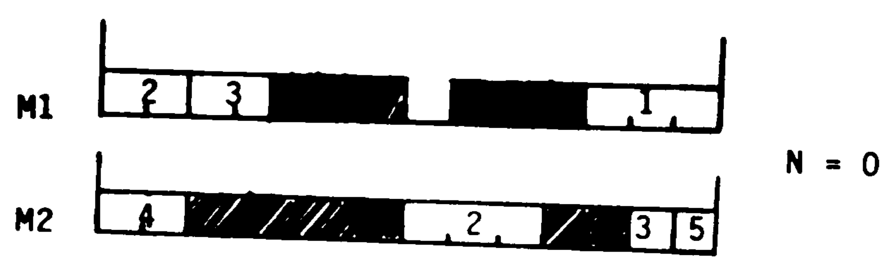


Elimination (2)

Opérations sans successeurs	$O_{1,1}$	$O_{5,1}$	$O_{2,2}$	$O_{3,2}$	$O_{4,2}$
ΔU	-1	0 ou(+1)	-2	-1	-1

L'opération $O_{5,1}$ est éliminée

Ecretage (3)



La régulation est terminée puisque N=0 ce qui signifie que la contrainte de charge des machines est respectée. L'utilisation des machines M1 et M2 est alors respectivement 13 et 14 pour cette période ordonnancée définitivement. La solution trouvée par notre méthode est celle représentée sur ce dernier diagramme.

IV.4. DISCUSSION SUR LES HYPOTHESES

Dans le chapitre III paragraphe 1.3 nous avons classé les hypothèses de notre modèle en trois groupes :

(GH1) hypothèses liées à la réalité de l'atelier

(GH2) hypothèses qui peuvent être levées sans difficulté

(GH3) hypothèses qui sont éminémment restrictives.

Il n'y a pas lieu de remettre en cause les hypothèses du groupe (GH1). Par contre, nous allons examiner la possibilité de supprimer les hypothèses de deux autres groupes.

1. Dans (GH2)

On suppose que les machines peuvent travailler en continu et que les temps de transit dans l'atelier sont nuls. La suppression de ces hypothèses n'entrainerait pas de modifications fondamentales de notre modèle.

La première conduit à un simple aménagement dans l'algorithme puisque l'hypothèse 1 de (GH1) suppose qu'une opération commencée sur une machine ne peut être interrompue par une autre et doit être menée à son terme. Cela peut facilement être pris en compte par les $M(j,\theta)$ donnant la composition du groupe de machines j en fonction du temps θ.

La deuxième hypothèse concernant les temps de transit conduirait de même à une modification mineure pour les prendre en compte. Ces temps de transit sont en effet des données et ne sont pas affectés par les séquences des opérations des diverses tâches sur les machines.

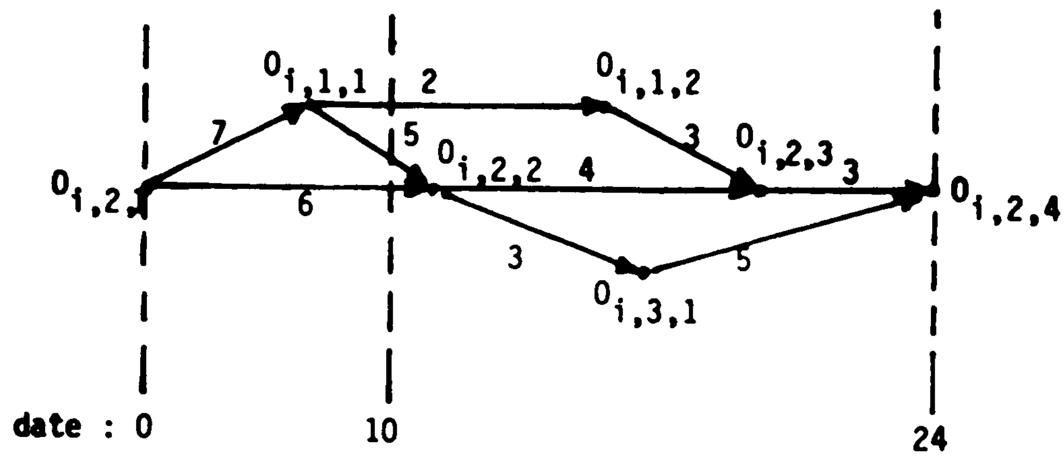
2. Dans (GH3)

(1) Les aléas :

Nous avons supposé qu'il n'y avait pas d'aléas de fabrication pour les opérations ordonnancées définitivement sur la période immédiate.

Ceci est évidemment restrictif puisque une des machines peut tomber en panne et donc ne pas pouvoir effectuer les opérations qui lui étaient affectées. Cela peut entrainer ainsi des répercussions sur les autres machines à cause des contraintes de succession.

Exemple : pour la tâche i



Si la machine 2 tombe en panne à la date 10 jusqu'à la date 24 alors $0_{i,2,2}$ ne pourra être effectuée. La machine 3 ne pourra pas réaliser $0_{i,3,1}$ et sera inactive pendant la durée de $0_{i,3,1}$. Par contre, la machine 1 ne sera pas affectée par la panne de la machine 2.

Donc, lorsqu'un aléa de fabrication intervient, il faut refaire l'ordonnancement sinon le reste de la période immédiate ainsi que les périodes suivantes risquent d'être perturbées. C'est alors la périodicité de l'ordonnancement qui est remise en cause et on aboutit à réaliser l'ordonnancement en temps réel. Une panne, ou un aléa de n'importe quelle nature, implique de refaire l'ordonnancement de la période immédiate et le chargement des périodes suivantes en prenant compte la modification du parc des machines (ce qui est possible grâce aux $M(j,\theta)$).

(2) Les temps d'adaptation

Le temps d'adaptation des machines sont fonction de la séquence des opérations. Pour une machine donnée, un temps inter-opératoire peut être nécessaire pour réaliser un réglage ou une reconversion : il dépend de l'opération qui se termine et de celle qui va commencer sur cette même machine. Dans le cadre de notre modèle, cette hypothèse de temps d'adaptation non nuls ne peut être levée car il s'agit alors d'effectuer l'ordonnancement vis-à-vis d'un critère supplémentaire. Ce critère des temps d'adaptation des machines n'est pas indépendant de ceux pris en compte puique notre but partiel est de maximiser l'utilisation des machines. Son implémentation demanderait cependant une profonde transformation de notre modèle.

Toutes ces considérations justifient pleinement le choix de répartir nos hypothèses en trois groupes définis dans le chapitre III.

CONCLUSION

Dans l'élaboration de notre modèle d'ordonnancement dynamique notre but principal a été de faire des choix tels que le problème résolu conserve son réalisme et s'approche le plus possible des situations concrètes.

Notre système comporte donc des hypothèses assez larges. Il permet de trouver une solution acceptable vis-à-vis de l'utilisation des machines et de la tenue des délais. Le mode d'assignation des dates de début aux opérations ne garantit cependant pas une stricte minimisation des retards.

Si nous supposons que les coûts de fabrication ne dépendent pas de l'ordre d'exécution des opérations alors tout ordonnancement admissible (A) entraîne le même coût fixe d'exécution des tâches. Il n'y a donc pas lieu de distinguer entre plusieurs ordonnancements admissibles.

Dans le cas de notre modèle, la solution obtenue n'est pas nécessairement admissible puisque des délais peuvent ne pas être respectés. Dans la réalité des coûts, tels que pénalités prévues en cas de retard, associés aux différentes tâches ou des coûts d'utilisation pour les machines peuvent intervenir. Nous nous limitons à considérer des coûts identiques pour toutes les tâches et pour toutes les machines. L'interprétation économique est alors la suivante pour notre modèle : toutes les tâches sont d'égale importance et il en est de même pour les machines.

Cette méthode se prête bien à une application en temps réel. La possibilité de faire d'autres essais avec modifications de certains paramètres comme les délais conduit à considérer la fixation de ceux-ci comme conséquence de l'ordonnancement.

Plus précisément les résultats de l'ordonnancement, période par période, peuvent fournir des éléments d'aide à la décision pour :

BIGLIOGRAPHIE

[1] CONWAY, MAXWELL, MILLER
Theory of scheduling - Adison Wesley - 1967.

[2] ASHOUR
Sequencing Theory - Springer Verlag - 1972.

[3] A.H.G. RINNOY KAN
Machine Scheduling problems - Martinus Nijhoff - 1976.

[4] NEPOMIASTCHY
Résolution d'un problème d'ordonnancement à ressources variables
RAIRO - Août 1978.

[5] MINOUX
Algorithmes gloutons - E.D.F. Bulletin - 1977.

[6] CARLIER
Ordonnancement à contraintes disjonctives
RAIRO - Novembre 1978.

[7] LITTLE
A proof for the Queuing Formula $L = \lambda W$
Operation Research n°3 - Mai 1961.

[8] ERSCHLER
Analyse sous contraintes et aide à la décision pour certains problèmes d'ordonnancement
Thèse d'état U.P.S. - Novembre 1976.

[9] ELMAGHRABY
The machine sequencing problem
Reviews and Extensions - N.R.L.Q. n°2 1968.

[10] BALAS
Machine Sequencing via disjunctive graphs : an implicit enumeration algorithm - Operations Research n°6 - 1969.

[11] COUZINET
Etude de l'existence de solutions pour certains problèmes d'ordonnancement - Thèse U.P.S. - 1979.

[12] FONTAN
Notion de dominance et son application à l'étude de certains problèmes d'ordonnancement - Thèse d'Etat U.P.S. - 1980.

[13] FLORIAN, BRATLEY, ROBILLARD
On sequencing with Earliest Starts and Due dates with applications to computing bounds for the $n/m/G/F_{max}$ problems
N.R.L.Q. 20 - 1973.

[14] IGNALL, SCHRAGE
Application to the Branch and Bound technique to some flow-shop scheduling problems - Operations Research - 1965.

[15] JOHNSON
Optimal two-and three-Stage Production Schedules with setup times included - N.L.R.Q. 1 - 1954.

[16] JACKSON
An Extension of Johnson's results on Job-Lot Scheduling
N.L.R.Q. 3 - 1956.

[17] MANNE
On the Job-Shop Scheduling Problem
Operations Research - 2 - 1960.

[18] FORTET
Applications de l'algèbre de Boole en R.O.
Revue Française de R.O. - 1960.

[19] ROY
Prise en compte des contraintes disjonctives dans les méthodes de chemin critique
A.F.I.R.O. n°38 - 1966.

[20] ERSCHLER, FONTAN, ROUBELLAT
Potentiels sur un graphe non conjonctif et analyse d'un problème d'ordonnancement à moyens limités
R.A.I.R.O. - Vol. 13, n°4 - 1979.

[21] WAGNER, GIGLIO
Approximate solutions to the three - Machines Scheduling Problem
Operations Research - n°2 - 1964.

[22] GUPTA
An Heuristic algorithme for the flow shop scheduling Problem
R.A.I.R.O. - Juin 1976.

[23] ACKERS, FRIEDMAN
A graphical approach to production scheduling Problems
Operations Research - Avril 1956.

SZWARC
Solution of the Ackers-Fridman Scheduling Problem
Operations Research - Novembre 1960.

[24] NEMHAUSER, HARDGRAVE
A geometric model and graphical Algorithm for a Sequencing Problem - Operations Research - Novembre 1963.

[25] GERE
Heuristic in Job shop scheduling - Management Science - n°3
Novembre 1966.

[26] MAXWELL
On Sequencing n jobs on one machine to minimize the number of late jobs - Management Science - n°5 - Janvier 1970.

[27] FLORIAN, TREPANT, MC MAHON
An implicit enumeration algorithm for the machine Scheduling Problem - Management Science - n°12 - Août 1971.

[28] ELMAGHRABY
The role of Modeling in I E Design
Journal of Industrial Engineering - Vol. 14 - n°6 - 1968.

[29] MITTEN
Branch and Bounds methods : General Formulations and properties
Operations Research - Janvier 1970.

[30] ASHOUR
An experimental Investigation and comparative evaluation of Flow-Shop scheduling techniques
Operations Research - Mai 1970.

[31] GIFFLER, THOMPSON
Algorithms for solving Productions Scheduling Problems
Operations Research - Juillet 1960.

[32] CAMPBELL, DUDEK, SMITH
A Heuristic Algorithm for the n Job, m Machine Sequencing Problem - Management Science - n°10 - Juin 1970.

[33] PIERCE, LASKY
Improved Combinatorial Programming Algorithms for a class of all-zero-one Integer Programming Problems
Management Science - n°5 - Janvier 1973.

[34] GOMORY
An Algorithm for Integer Solution to linear Programs
IBM - Technical Report - 1 - Novembre 1958.

[35] BOWMAN
The schedule-sequencing Problem - Operations Research - 1959.

ANNEXE A

NOTATIONS UTILISÉES

a_i	Date d'arrivée de la tâche i dans l'atelier.
A	Critère de l'avance des tâches.
A_i	Avance de la tâche i.
$\bar{A}$	Avance moyenne de l'ensemble des tâches en avance.
c	Date de début au plus tôt (commune à toutes les tâches).
c_i	Date de début au plus tôt de la tâche i.
C_i	Date de début effective de la tâche i.
$C_{i,j,k}$	Date de début effective de l'opération $O_{i,j,k}$.
$C^-_{i,j,k}$	Date de début au plus tôt de l'opération $O_{i,j,k}$.
$C^+_{i,j,k}$	Date de début au plus tard de l'opération $O_{i,j,k}$.
d_i	Date de fin au plus tard de la tâche i.
D_i	Date de fin effective de la tâche i.
$\bar{D}$	Date de fin moyenne des tâches.
$D_{i,j,k}$	Date de fin effective de l'opération $O_{i,j,k}$.
E	Critère de l'écart entre les dates de fin au plus tard et les dates de fin effectives des tâches.
E_i	Ecart entre D_i et d_i pour la tâche i.
f_i	Durée minimale de fabrication de la tâche i si elle était seule.

$\overline{T}$	Durée moyenne minimale de fabrication d'une tâche si elle était seule.
$f_{i,j,k}$	Durée de l'opération $O_{i,j,k}$.
F	Atelier à flot unique (flow-shop).
G	Atelier à flot quelconque.
i	Désigne la tâche générique.
I	Inactivité des machines.
j	Désigne le groupe de machines générique.
$J(j,\theta)$	Charge du groupe j de machines sur l'intervalle $[\theta,\theta+1[$.
k	Désigne la $k^{\text{ième}}$ opération d'une même tâche sur le groupe de machine concerné.
$\ell_{i,j,k}$	Liberté de manoeuvre pour le chargement de l'opération $O_{i,j,k}$.
m	Nombre de groupes de machines.
$M(j,\theta)$	Donne la composition du groupe de machines j en fonction du temps θ.
n	Nombre de tâches.
$O_{i,j,k}$	$k^{\text{ième}}$ opération de la tâche i sur le groupe de machines j.
p_i	Temps alloué à la tâche i dans l'atelier.
$\overline{p}$	Moyenne des p_i.

P_i	Temps total réel de passage dans l'atelier de la tâche i.
P	Critère du temps réel de passage des tâches dans l'atelier.
$\overline{P}$	Moyenne des P_i.
R_i	Retard de la tâche i.
R	Critère du retard des tâches.
$\overline{R}$	Moyenne des retards des tâches (en retard).
t	Désigne le temps.
t_0	Date initiale du lère ordonnancement effectué.
$t(O_{i,j,k},O_{i,r,s})$	Temps de transit entre la fin de $O_{i,j,k}$ et le début de $O_{i,r,s}$.
T	Période de l'ordonnancement.
U	critère de l'utilisation des machines.
$\overline{U}$	Taux moyen d'utilisation des machines.
W_i	Temps d'attente de la tâche i dans l'atelier.
W	Critère du temps d'attente.
$\overline{W}$	Temps d'attente moyen des tâches dans l'atelier.
θ	Désigne le temps.
θ_0	Date initiale de la lère période à ordonnancer.

λ Taux d'arrivée des tâches dans l'atelier.

$<$ Appliqué à deux opérations signifie "précède immédiatement".

$<<$ Appliqué à 2 opérations signifie "est prise en compte avant".

ANNEXE B

RESOLUTION GRAPHIQUE DU PROBLÈME N/M/F/P

Dans le chapitre II paragraphe 3.2., nous avons cité la méthode de résolution graphique de ACKERS [23] pour un problème 2/m/F/P. Nous voulons simplement évoquer ici cette méthode graphique par une généralisation possible au cas n/m/F/P.

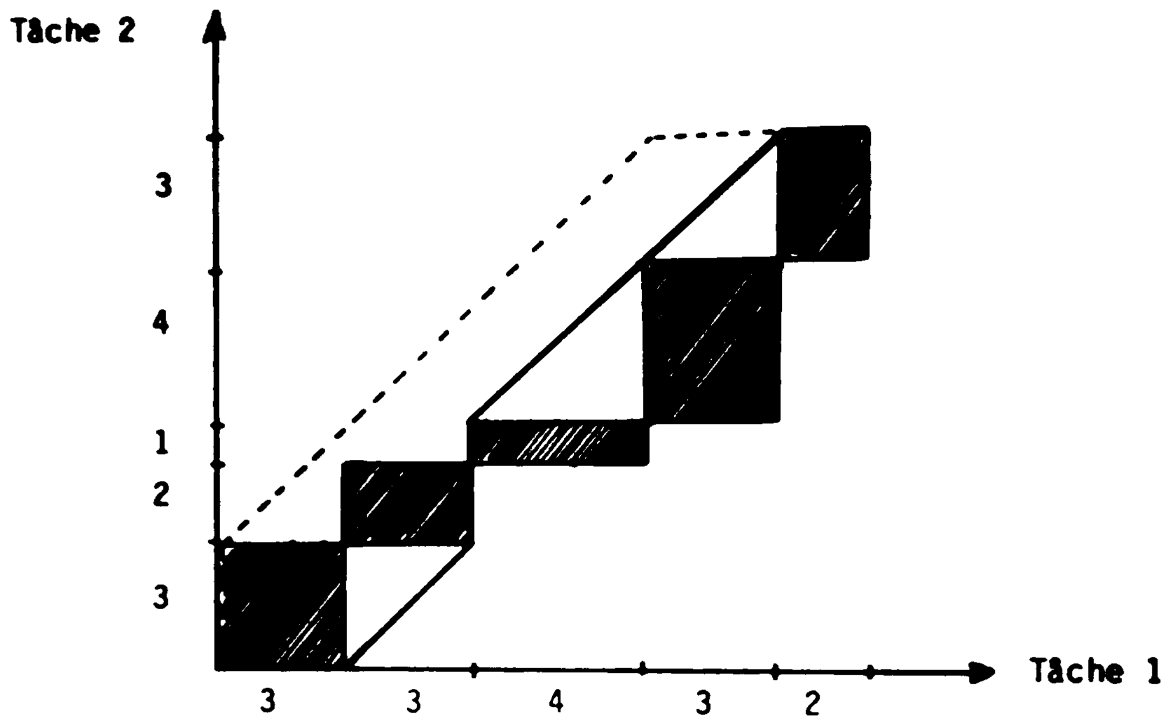
La démarche générale suggérée pour résoudre un problème n/m/F/P est la suivante :

il s'agit de trouver une solution graphique à n-1 problèmes. Le $(i+1)^{\text{ième}}$ problème comporte i tâches que l'on a partitionnées ainsi : sur un axe les (i-1) tâches du $(i-2)^{\text{ième}}$ problème déjà résolu et sur l'autre axe la $i^{\text{ème}}$ tâche à insérer dans l'ordonnancement existant.

Il faut remarquer qu'ainsi formulée la résolution ne peut donner qu'une "bonne solution" et non pas la solution optimale vis à vis de P.

L'exemple proposé est un problème 3/5/F/P. La résolution commence par le traitement d'un problème 2/5/F/P (figure 1). Puis la troisième tâche est insérée dans l'ordonnancement des deux autres tâches. La figure 2 montre la solution graphique obtenue à partir de la solution 1. Sur la figure 3, la même démarche appliquée à la solution 2 conduit à un autre ordonnancement. Enfin la solution de départ pour la résolution correspond à la figure 4 est la solution 2 où les opérations sont cadrées au plus tard (c'est-à-dire un autre chemin optimal non porté sur la figure 1). Pour ce problème 3/5/F/P les solutions optimales vis à vis de P sont $P^* = 20$ (figures 2 et 4).

Figure 1

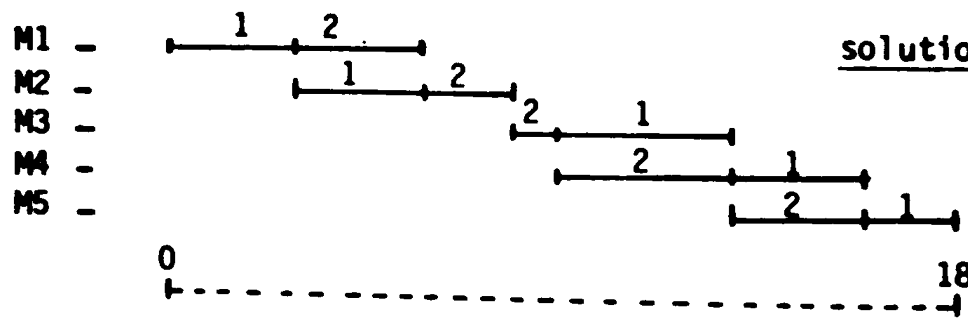


Deux chemins de longueur minimale = 18 (non exclusivement)

Vis à vis du critère P, il y a donc deux solutions optimales équivalentes.

1) Correspondant au chemin en trait plein, la solution est

solution 1



2) Correspondant au chemin en trait pointillé la solution est

solution 2

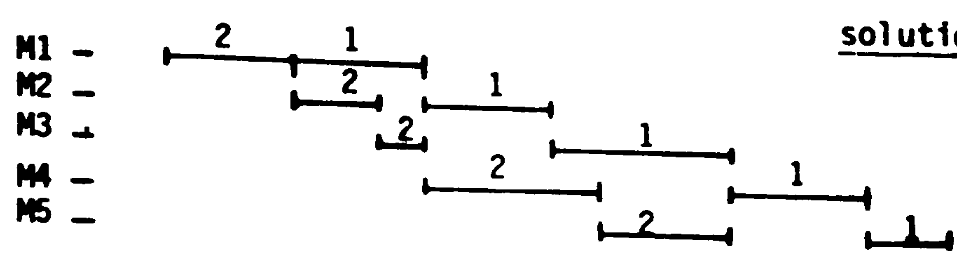
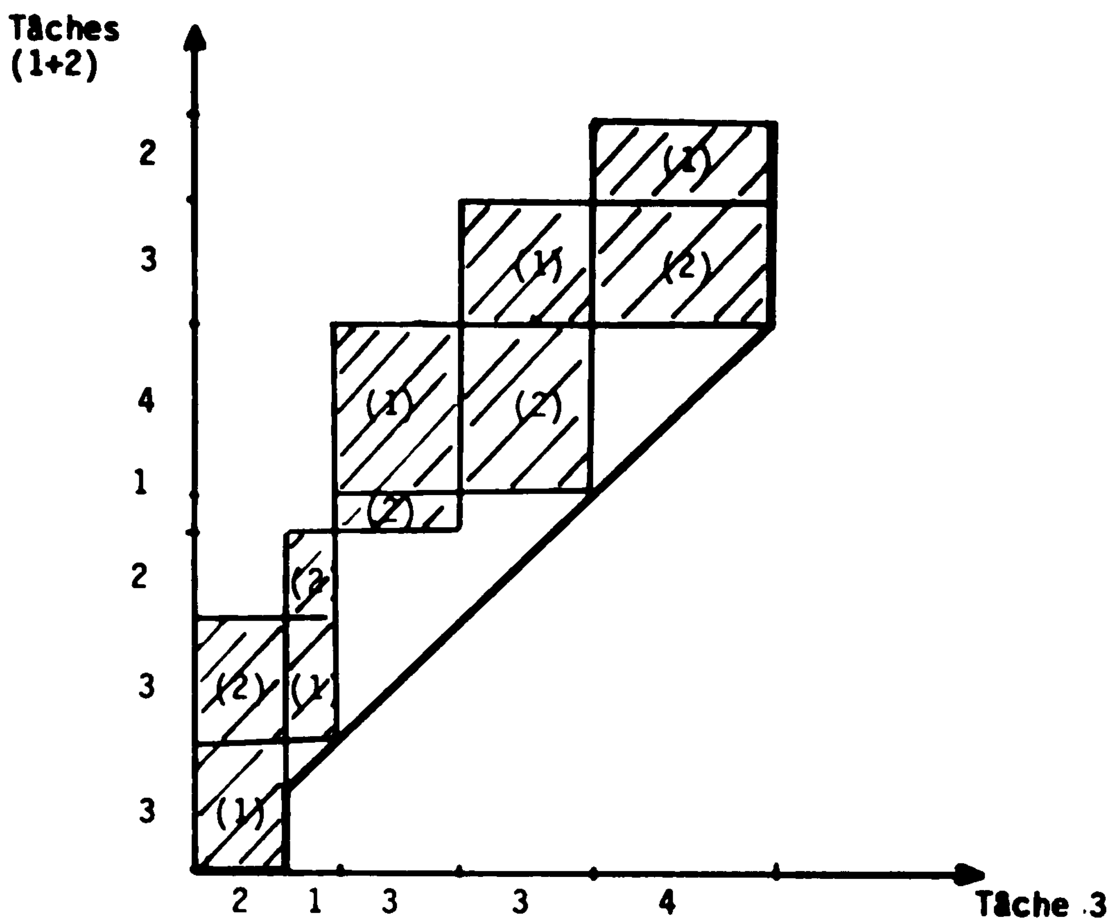
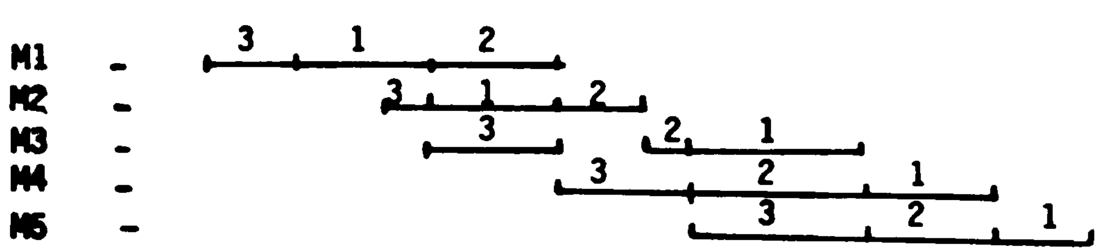


Figure 2

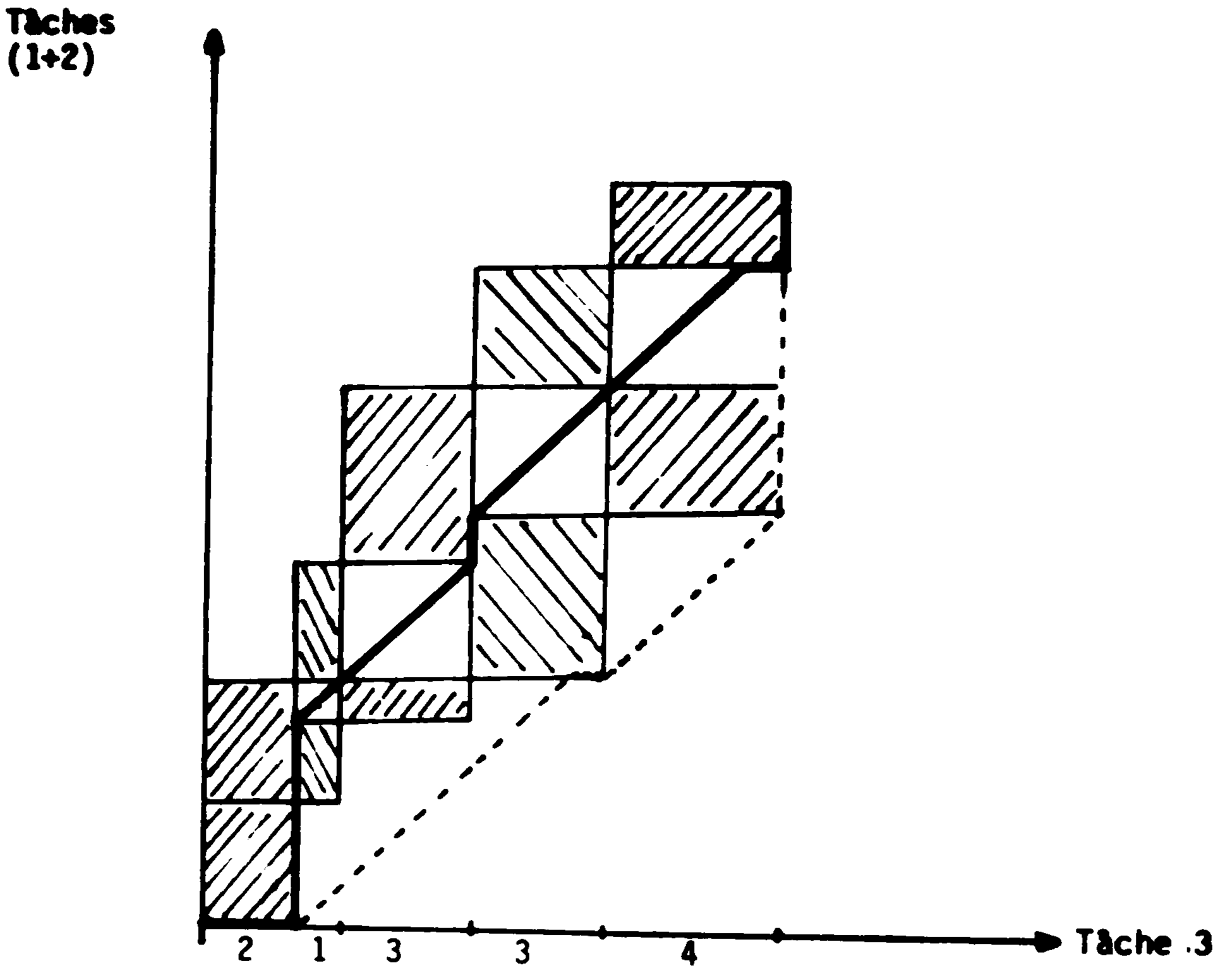


Chemin de longueur minimale = 20

Il correspond à la solution suivante

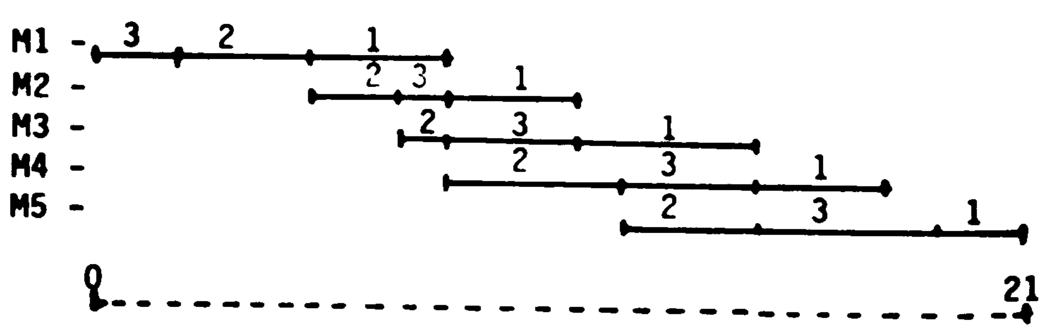


<u>Figure 3</u>



Deux chemins de longueur minimale égale à 21 (non exclusivement)

1) Correspondant au chemin en trait plein, la solution est



2) Correspondant au chemin en trait pointillé la solution est

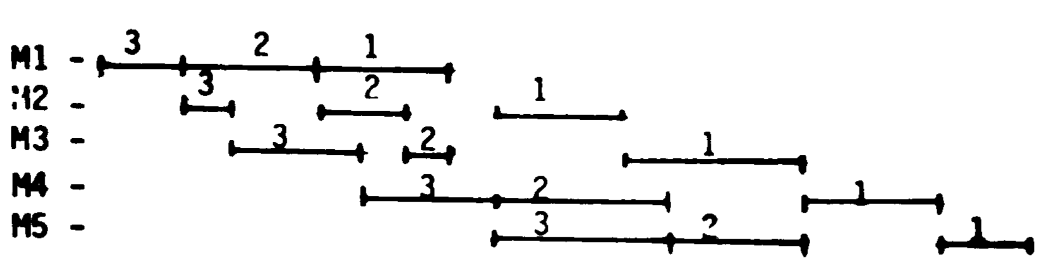
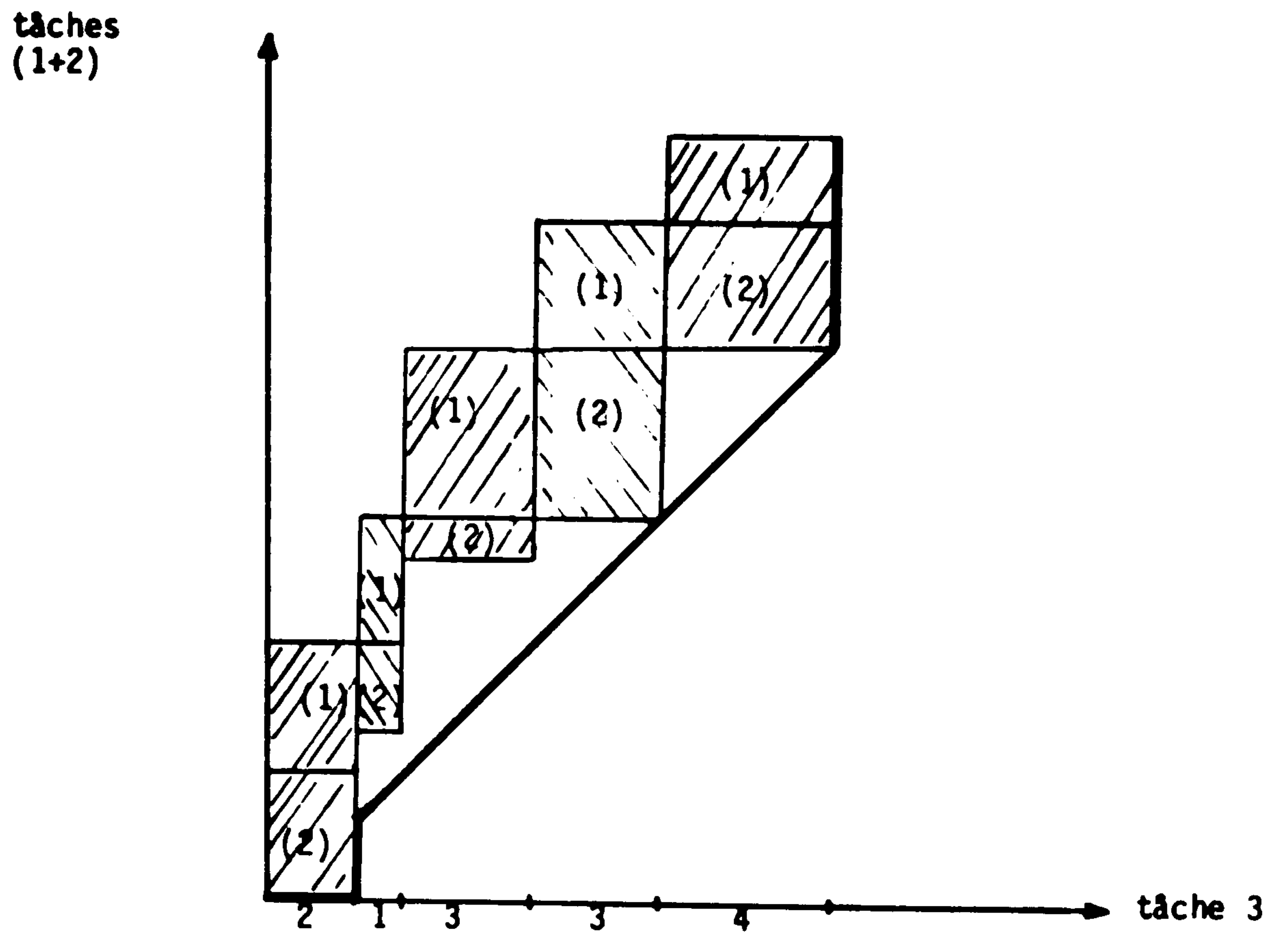
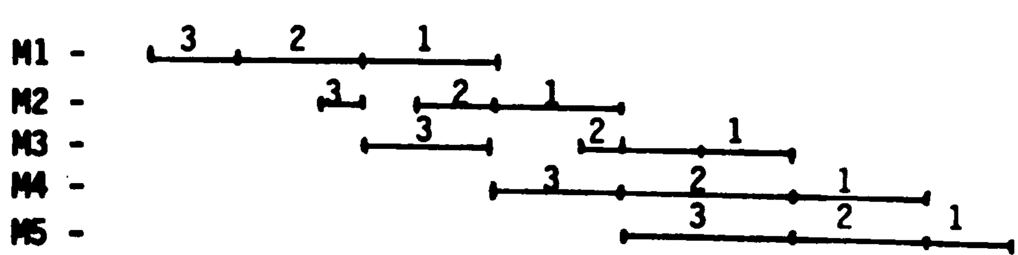


Figure 4



Chemin de longueur minimale égale à 20 correspondant à la solution suivante



Annexe C

Listing du Programme

```
STMT LEVEL NEST
 13    1        DCL ORPIC PIC '99999999',
       1            ORCHD CHAR(8) DEF ORPIC,
       1             1 ORPICD DEF ORPIC,
       1                2 NUMJ PIC '9999',
       1                2 NUMM PIC '99',
       1                2 NUMO PIC '99';
 14    1        DCL SUCPIC(5) PIC '99999999',
       1            SUCD(5) CHAR(8) DEF SUCPIC;
 15    1        DCL (I,J,NOP,DA,DD,NJAS) REAL BIN FIXED(31),
       1             NTEC REAL BIN FLOAT(24),
       1            (TRETARD,TAVANCE,GRBAR,GABAR) REAL BIN FLOAT(24),
       1             EOJ BIT(1);
 16    1        DCL (DAO,DJ,NJTOTAL,NJENRETARD,NJENAVANCE,NJOK) REAL BIN FIXED(31);
 17    1        DCL NULL BUILTIN;
 18    1        /*---------------------------------------------------------*/
       1
       1        ON ERROR SNAP SYSTEM;
 19    1        ON ENDFILE(SYSIN) EOJ='1'B;
 21    1        EOJ='0'B;
 22    1        DAO=0;
 23    1        NTEC,TRETARD,TAVANCE=0;
 24    1        NJTOTAL,NJENRETARD,NJENAVANCE,NJOK=0;
 25    1        UTILISATION=1;
 26    1        PUT SKIP EDIT ('ORDONNANCEMENT A LA DATE',DAO) (X(40),A,F(5));
 27    1        /* LECTURE DU NBRE EFFECTIF DE MACHINES */
       1        GET SKIP LIST (NBMACH);
 28    1        /* APPEL INITIALISATION DES CHARGES */
       1        CALL INITCHARG(NBMACH);
 29    1        GET SKIP LIST (NJAS,DA);
 30    1        ARRIVEE: DO WHILE (¬EOJ);
 31    1    1                  IF   DA=DAO+1
       1    1                  THEN;
 33    1    1                  ELSE DO;
 34    1    2                        /*DJ : JOUR A ORD. DEFINITIVEMENT */
       1    2                         DJ=DAO+1;
 35    1    2                         CALL MAJDIST;
 36    1    2                         CALL REGULCHARGES(DJ);
 37    1    2                         PUT SKIP EDIT ('*** STATISTIQUES ***') (X(20),A);
 38    1    2                         PUT SKIP EDIT ('= ORD.T DU JOUR:',DJ) (X(20),A,F(5));
 39    1    2                         CALL RESULTATS(DJ);
 40    1    2                         DAO=DA-1;
 41   -1    2                         PUT PAGE EDIT ('ORDONNANCEMENT A LA DATE',DAO)
       1    2                                   (X(40),A,F(5));
 42    1    2                         DO I=1 TO NBMACH;
 43    1    3                           CHARMACH(I)=SUBSTR(CHARMACH(I),25);
 44    1    3                         END;
 45    1    2                       END;
 46    1    1                  DO J=1 TO NJAS;
 47    1    2                     GET SKIP LIST (NO,DD,NOP);
 48    1    2                     ALLOCATE CELJOB SET(PJOB);
 49    1    2                     PJOB->NUJOB=NO;PJOB->DARAT=DA;
 51    1    2                     PJOB->DATEDUE=DD;PJOB->LIEN=TLJOB;
 53    1    2                     PJOB->INDRET='0'B;PJOB->DATEFIN=0;
 55    1    2                     PJOB->NBOPON=NOP;PJOB->NBOPER=NOP;
 57    1    2                     TLJOB=PJOB; /* INSERTION EN TETE */
 58    1    2                     CALL CREGRAPHJOB(NOP,NO,DA,DD,PJOB);
 59    1    2                  END;
```

```
STMT LEVEL NEST      • • • •   ORDPRODUCTION   • • • •                              PL/1 4.2   LE    26*05*81         3
 40      1     1                  /* CHARGEMENT SELON CRIO */
         1     1                  DO WHILE (TLOAT#=NULL);
 41      1     2                       CALL CALCRIO(PAT,PPAT);
 42      1     2                      CALL TRAITCHARGES(PAT,PPAT);
 43      1     2                      /* PROG DS LA LISTE DES OPERS */
         1     2                  END;
 44      1     1                  GET SKIP LIST (NJAS,DA);
 65      1     1         END ARRIVEE;
 66      1               PUT SKIP LIST ('FIN D ARRIVEE DES JOBS');
 67      1               PUT SKIP EDIT ('*** STAT GLOBAL ***') (A);
 68      1               PUT SKIP EDIT ('NJTOTAL NJOK,NJENRETARD,NJENAVANCE') (A);
 69      1               PUT SKIP EDIT (NJTOTAL,NJOK,NJENRETARD,NJENAVANCE)
         1                             (F(7),F(5),F(11),F(11));
 70      1               PUT SKIP EDIT ('MONTANT-TOTAL-EN-COURS =',MTEC) (A,F(10,2));
 71      1               UTILBAR=UTILISATION/(NBMACH*DJ);
 72      1               PUT SKIP EDIT ('UTILISATION DES MACHINES=',UTILBAR) (A,F(6,2));
 73      1               IF NJENRETARD#=0
         1               THEN GRBAR=TRETARD/NJENRETARD;
 75      1               ELSE GRBAR=0;
 76      1               IF NJENAVANCE#=0
         1               THEN GABAR=TAVANCE/NJENAVANCE;
 78      1               ELSE GABAR=0;
 79      1               PUT SKIP EDIT ('GRBAR =',GRBAR,'GABAR =',GABAR) (A,F(10,2),A,F(10,2));
 80      1               PUT SKIP LIST ('***==============================*****');
 81      1               CALL MAJDIST;
 82      1               DO I=1 TO 10; DJ=DJ+1;
 84      1     1              PUT PAGE LIST ('ORDON . PARTIEL DU JOUR:',DJ);
 85      1     1              CALL RESULTATS(DJ);
 86      1     1         END;
 87      1               PJOB=TLJOB;
 88      1               DO WHILE (PJOB#=NULL);
 89      1     1            PUT SKIP LIST ('JOB NO ',PJOB->NOJOB);
 90      1     1            IF  PJOB->NOJOB#=0000
         1     1            THEN DO I=1 TO PJOB->NBOPER;
 92      1     2               P=TP(PJOB(I));
 93      1     2               PUT SKIP EDIT ('OPCH =',P->OPCH,'DDEFF =',P->DDEFF) (A,A(8),A,F(3));
 94      1     2               PUT SKIP EDIT ('DAMONT =',P->DAMONT,'DAVAL =',P->DAVAL) (A,F(3),A,F(3));
 95      1     2               PUT SKIP EDIT ('MAMONT =',P->MAMONT,'MAVAL =',P->MAVAL) (A,F(3),A,F(3));
 96      1     2                 END;
 97      1     1            ELSE;
 98      1     1            PJOB=PJOB->LIEN;
 99      1     1         END;
100      1
         1
         1               /*=====================================================*/
         1               /* REGULCHARGES : PROC INTERNE A ORDPRODUCTION          */
         1               /*                ----------------------               */
         1               /* PARAMETRES= DJAO: DATE DU JOUR A ORDONNANCER         */
         1               /*                                                     */
         1               /*-----------------------------------------------------*/
         1
         1               REGULCHARGES:PROC(DJAO);
101      2
         2               DCL (TROU(NBMACH,2,24),
         2                    NBTROU(NBMACH),UTILACT(NBMACH),
         2                    IM,K,NOTR,NBOPA,MAXNBTROU,NJOUR)
         2                                                       REAL BIN FIXED(31),
         2                    1 STRUCTAB BASED(PTT),
```

```
STMT LEVEL NEST    * * * *   ORDPRODUCTION   * * * *                    PL/1 4.2   LE   26*05*81        5
            2              /*-------------------------------------------------------------*/
            2
            2              /* POUR CHAQUE TROU ,CREATION D'UNE LISTE D'OPERS            */
            2              /*          POUVANT ETRE AVANCEES                            */
            2
            2              RECHERCHOPER:PROC(NJOUR);
 140        3
            3              DCL (NJOUR,IM,NOTR,L,LL,I,J,IT,AV) REAL BIN FIXED(31),
            3                  (IMPOS,FIN,DER) BIT(1),
            3                  (PS,PP,R,P,Q) POINTER
            3                                               ;
 141        3
            3              /* POS. SUR 1ERE OPER(DE CHAQUE MACH) QUI SERA CONCERNEE */
            3              DO IM=1 TO NBMACH;
 142        3      1          P=TPLIST(IM); DER='0'B;
 144        3      1          DO WHILE ((P¬=NULL)&(¬DER));
 145        3      2            IF P->POPER->DDEFF<=DJAO*24 THEN P=P->SUIV;
 147        3      2                                        ELSE DER='1'B;
 148        3      2          END;
 149        3      1          PCOUR(IM)=P; /*ON EXAMINERA A PARTIR DES PCOUR */
 150        3      1       END;
 151        3
            3              DO IM=1 TO NBMACH;
 152        3      1        P=PCOUR(IM);
 153        3      1        DO WHILE (P¬=NULL);
 154        3      2          P->POPER->NDDEFF=P->POPER->DDEFF;
 155        3      2          P->POPER->NDAMONT=P->POPER->DAMONT;
 156        3      2          P->POPER->NDAVAL=P->POPER->DAVAL;
 157        3      2          P=P->SUIV;
 158        3      2        END;
 159        3      1       END;
 160        3              /* RECHERCHE */
            3              IMPOS='0'B; NBOPA=0;
 162        3              DO WHILE (¬IMPOS);
 163        3      1          IMPOS='1'B;
 164        3      1          DO IM=1 TO NBMACH;
 165        3      2            P=PCOUR(IM);
 166        3      2            IF P=NULL THEN FIN='1'B; ELSE FIN='0'B;
 169        3      2            DO WHILE (¬FIN);
 170        3      3
            3      3              IF (¬(P->FIXE))&(¬(P->POPER->TRAITEE))&(P->POPER->TRAITPRED)
            3      3               THEN
            3      3                DO NOTR=1 TO NBTROU(IM) WHILE (¬(P->POPER->TRAITEE));
 172        3      4                  IF P->POPER->DUROP<=TROU(IM,2,NOTR)
            3      4                   THEN DO;
 174        3      5                     LL=TROU(IM,2,NOTR)-P->POPER->DUROP;
 175        3      5                     DO L=1 TO LL+1 WHILE (¬(P->POPER->TRAITEE));
 176        3      6                       IF P->POPER->DDEFF-(DJAO-1)*24-P->POPER->NDAMONT
            3      6                                  <=TROU(IM,1,NOTR)+L-1
            3      6                        THEN DO; /*POSSIBLE DE DEPLACER */
 178        3      7                          P->POPER->TRAITEE='1'B;
 179        3      7                          P->POPER->SANSSUC='1'B;
 180        3      7                          IMPOS='0'B;
 181        3      7                          ALLOCATE LISTROU SET(Q);
 182        3      7                          NBOPA=NBOPA+1;
 183        3      7                          Q->PTOPER=P->POPER;
 184        3      7                          Q->POSITION=L;
 185        3      7                          Q->SUIVANT=PTT->TPTROU(IM,NOTR);
```

```
CARTE      STMT LEVEL  NEST     * * * *   ORDPRODUCTION  * * * *                         PL/1 4.2   LE    26*05*81
 291        186     3     7                          Q->PLISTOP=P;
 292        187     3     7                          PTT->TPTROU(IM,NOTR)=0;
 293        188     3     7                          AV=P->POPER->DDEFF-(DJAO-1)*24-TROU(IM,1,NOTR)-L+1;
 294        189     3     7                          P->POPER->DPL=AV;
 295        190     3     7                          P->POPER->NDDEFF=P->POPER->DDEFF-AV;
 296        191     3     7                          P->POPER->NDFOPER=P->POPER->DFOPER-AV;
 297        192     3     7                          /* MODIF DISTANCES ET TRAITPRED */
 298                3     7                          P->POPER->NDAMONT=P->POPER->NDAMONT-AV;
 299        193     3     7                          P->POPER->NDAVAL=P->POPER->NDAVAL+AV;
 300        194     3     7                          DO J=1 TO P->POPER->NBSUC;
 301        195     3     8                            PS=P->POPER->TPTSUC(J);
 302        196     3     8                            PS->TRAITPRED='1'B;
-303        197     3     8                            PS->NDAMONT=9999;
 304        198     3     8
 305                3     8                            DO I=1 TO PS->NBPRED;R=PS->TPTPRED(I);
 306        200     3     9                              PS->TRAITPRED=(PS->TRAITPRED)&(R->TRAITPRED);
 307        201     3     9               PS->NDAMONT=MIN(PS->NDAMONT,PS->NDDEFF-R->NDFOPER-1);
 308        202     3     9                            END;
 309        203     3     8                          END;
 310        204     3     7                          DO J=1 TO P->POPER->NBPRED;
 311        205     3     8                            PP=P->POPER->TPTPRED(J);
 312        206     3     8                            PP->SANSSUC='0'B;
 313        207     3     8          PP->NDAVAL=9999;
 314        208     3     8                            DO I=1 TO PP->NBSUC;R=PP->TPTSUC(I);
 315        210     3     9            PP->NDAVAL=MIN(PP->NDAVAL,R->NDDEFF-PP->NDFOPER-1);
 316        211     3     9                            END;
 317        212     3     8                          END;
 318        213     3     7                              END;
 319        214     3     6                   END;
 320        215     3     5                       END;
 321        216     3     4                 ELSE ; /*IMPOSSIBLE POUR CE TROU */
 322        217     3     4             END;
 323        218     3     3             P=P->SUIV;
 324        219     3     3             IF P=NULL THEN FIN='1'B;
 325        221     3     3                          ELSE
 326                3     3                           IF P->POPER->DDEFF>(DJAO+NJOUR)*24
 327                3     3                             THEN FIN='1'B;
 328        223     3     3       END;
 329        224     3     2   END;
 330        225     3     1 END;
 331        22      3
 332                3
 333                3       END RECHERCHOPER;
 334        227     2
 335                2       /*-------------------------------------------------*/
 336                2       /* SOLUTION : PROC INTERNE A REGULCHARGES          */
 337                2       /*            --------------------                 */
 338                2       /*PARAMETRE= NBOPA: NB OPER CANDIDATES             */
 339                2       /*                                                 */
 340                2       /*-------------------------------------------------*/
 341                2
 342                2       SOLUTION: PROC(NBOPA);
 343        228     3
 344                3       DCL
 345                3           (GNM(NBMACH,24),AGNM(24),SNGNM(24),NGNM(24),
 346                3            GNMACH(NBMACH),GN,DEPL,INITGN,VARGN,
 347                3            NOPA,RET,POSI,DELTAU,MINVU,MAXAGNM,MAXNGNM,
 348                3            ECONFIG(NBMACH,NBOPA,24),
```

```
CARTE        STMT LEVEL  NEST      * * * *   DEPRODUCTION   * * * *                        PL/I 4.2   LE    26*05*81        7
349                3                    AGN,SIM,NGN,NOTROU,IM,IT,NT,I,J,IP,II,IP1,L,
350                3                    IDEB,IFIM,PREF,PMAX,PMIN,LMOINS,LPLUS)
351                3                                                  REAL BIN FIXED(31),
352                3                    (P,SUIV,PPS,PS,PR,R,Q,PP,PT)     POINTER,
353                3                    (GNINV,VUM,PREPAS,FINI,SOL,MEILLEUR)     BIT(1);
354          229   3
355                3                CALL CALGN;
356          230   3                IF GN=0 THEN SOL='1'B ;ELSE SOL='0'B;
357          233   3                DO WHILE (¬SOL);
358          234   3     1            /*      ECRETAGE    */
359                3     1           GNINV='0'B; AGN=GN;
360          236   3     1           DO WHILE (¬GNINV);
361          237   3     2             GNINV='1'B;
362          238   3     2             DO IM=1 TO NBMACH;
363          239   3     3               AGNM(*)=GNM(IM,*);
364          240   3     3               DO IT=NBTROU(IM) TO 1 BY -1;
365          241   3     4                  PP=NULL;PT=PTT->TPTROU(IM,IT);
365          243   3     4                DO WHILE (PT¬=NULL);
367          244   3     5                  NGNM=AGNM;
368          245   3     5                  /* CALCUL DE NGNM SANS CETTE OPER;INIT DE VARGN */
369                3     5                  IDEB=PT->POSITION+TROU(IM,1,IT)-1;
370          246   3     5                  DEPL=IDEB;
371          247   3     5                  IFIN=IDEB+PT->PTOPER->DUROP-1;
372          248   3     5                  INITGN=0; SNGNM=NGNM;
373          250   3     5                  DO I=IDEB TO IFIN;
374          251   3     6                    SNGNM(I)=SNGNM(I)-1;
375          252   3     6                    IF SNGNM(I)¬=-1 THEN INITGN=INITGN-1;
376          254   3     6                  END;
377          255   3     5                /* CALCUL DE PMIN ET PMAX POUR CETTE OPER */
378                3     5                 PMIN=IDEB-PT->PTOPER->NDAMONT;
379          256   3     5                 PMAX=IDEB+PT->PTOPER->NDAVAL;
380          257   3     5                           NOTROU=IT;PREF=PT->POSITION+TROU(IM,1,IT)-1;
381          259   3     5                /* ENVISAGE TOUS LES TROUS VERS LA DROITE */
382                3     5                 DO NT =IT TO NBTROU(IM);
383          260   3     6                   FINI='0'B;
384          261   3     6                   IF TROU(IM,2,NT)<PT->PTOPER->DUROP THEN FINI='1'B;
385          263   3     6                   IF ¬FINI THEN
386                3     6                     DO; LMOINS=TROU(IM,1,NT);
387          266   3     7                        IF LMOINS>PMAX THEN FINI='1'B;
388          268   3     7                         ELSE DO;
389          269   3     8                             LPLUS=LMOINS+TROU(IM,2,NT)-PT->PTOPER->DUROP;
390          270   3     8                             IF LPLUS<PMIN THEN FINI='1'B;
391          272   3     8                               END;
392          273   3     7                     END;
393          274   3     6                   IF ¬FINI THEN
394                3     6                     DO; LPLUS=MIN(LPLUS,PMAX);
395          277   3     7                        LMOINS=MAX(LMOINS,PMIN);
396          278   3     7                        DO L=LMOINS TO LPLUS;
397          279   3     8                          VARGN=INITGN; NGNM=SNGNM;
398          281   3     8                          DO I=L TO L+PT->PTOPER->DUROP-1;
399          282   3     9                            NGNM(I)=NGNM(I)+1;
400          283   3     9                            IF NGNM(I)¬=0 THEN VARGN=VARGN+1;
401          285   3     9                          END;
402          286   3     8                          MEILLEUR='0'B;
403          287   3     8                          IF VARGN<0 THEN MEILLEUR='1'B;
404          289   3     8                           ELSE IF VARGN=0 THEN
405                3     8                                 DO; MAXNGNM=0;MAXAGNM=0;
406          293   3     9                                   DO J=1 TO 24;
```

```
          4        END CALGN;
  471     3
          3        /*-------------------------------------------------*/
          3
          3        -END SOLUTION;
  472     2
          2        /*---------------------------------------------------*/
          2        /* MODIFLC: PROC INTERNE A REGULCHARGES               */
          2        /*       --------------------------                   */
          2        /*                                                    */
          2        /*---------------------------------------------------*/
          2
          2        MODIFLC: PROC;
  473     3
          3
          3        DCL   (IM,IT)    REAL BIN FIXED(31),
          3              (POP,PT,P,PLIS,PP,PRED,AVANT)      POINTER;
  474     3
          3        DO IM=1 TO NBMACH;
  475     3    1      UTILNEW(IM)=UTILACT(IM);
  476     3    1      DO IT=1 TO NBTROU(IM);
  477     3    2        /* MODIF DS LISTOPMACH DE OPER PTEE PAR PLIS */
          3    2        PT=PTT->TPTROU(IM,IT);
  478     3    2        DO WHILE(PT¬=NULL);
  479     3    3           PLIS=PT->PLISTOP;
  480     3    3           POP=PT->PTOPER;
  481     3    3           SUBSTR(CHARMACH(IM),POP->DDEFF-DAO*24,POP->DUROP)=(24)'0'B;
  482     3    3           POP->DAMONT=POP->NDAMONT;
  483     3    3           POP->DAVAL=POP->NDAVAL;
  484     3    3           POP->TRAITEE='0'B;
  485     3    3          -IF POP->NBPRED¬=0 THEN POP->TRAITPRED='0'B;
  487     3    3           POP->DDEFF=TROU(IM,1,IT)+PT->POSITION-1+DAO*24;
  488     3    3           POP->DFOPER=POP->DDEFF+POP->DUROP-1;
  489     3    3          /* RECHERCHE ANCIEN EMPLACEMENT  */
          3    3           P=TPLIST(IM);AVANT=NULL;
  491     3    3           DO WHILE(P->POPER->OPCH¬=POP->OPCH);
  492     3    4              AVANT=P;P=P->SUIV;
  494     3    4           END;
  495     3    3           IF AVANT=NULL THEN TPLIST(IM)=PLIS->SUIV;
  497     3    3                        ELSE AVANT->SUIV=PLIS->SUIV;
  498     3    3           /* RECHERCHE NEW EMPLACEMENT */
          3    3           P=TPLIST(IM);PRED=NULL;
  500     3    3           DO WHILE (P->POPER->DDEFF<POP->DDEFF);
  501     3    4              PRED=P; P=P->SUIV;
  503     3    4           END;
  504     3    3           /* INSERTION APRES PRED DS LISTOPMACH  */
          3    3           IF PRED = NULL THEN DO;
  506     3    4                                PLIS->SUIV=TPLIST(IM);
  507     3    4                                TPLIST(IM)=PLIS;
  508     3    4                              END;
  509     3    3                        ELSE DO;
  510     3    4                                PLIS->SUIV=PRED->SUIV;
  511     3    4                                PRED->SUIV=PLIS;
  512     3    4                              END;
  513     3    3           /* MODIF DE CHARMACH        */
          3    3           SUBSTR(CHARMACH(IM),POP->DDEFF-DAO*24,POP->DUROP)=(24)'1'B;
  514     3    3           UTILNEW(IM)=UTILNEW(IM)+POP->DUROP;
  515     3    3           PP=PT;
```

```
STMT LEVEL NEST
 516    3    3            PT=PT->SUIVANT;
 517    3    3            FREE PP->LISTROU;
 518    3    3          END;
 519    3    2        END;
 520    3    1     END;
 521    3          FREE PTT->STRUCTAB;
 522    3
        3          END MODIFLC;
 523    2
        2          /*-------------------------------------------------------*/
        2
        2          END REGULCHARGES;
 524    1
        1          /*=====================================================*/
        1
        1          /*=========================================================*-
        1          /* PROCEDURE INTERNE A ORDPRODUCTION : INITCHARG           */
        1          /*               -----------------------------              */
        1          /* PARAMETRES = NBMACH: NBRE DE MACHINES                   */
        1          /*=========================================================*/
        1
        1          INITCHARG: PROC(NBMACH);
 525    2          DCL NBMACH REAL BIN FIXED(31),
        2               PRED POINTER,
        2               (I,J,K) REAL BIN FIXED(31);
 526    2          /* INIT CHARGES MACHINES */
        2          DO I=1 TO NBMACH; CHARMACH(I)=(240)'0'B; K=1;
 529    2    1        DO J=10 TO 2 BY -2;
 530    2    2           SUBSTR(HISTOMACH(I,K),1,J)=(24)'1'B; K=K+1;
 532    2    2        END;
 533    2    1     END;
 534    2          /* INIT JOB BIDON */
        2          TLJOB=NULL; ALLOCATE CELJOB SET(PJOB);
 536    2          PJOB->NOJOB=00; PJOB->DARAT=1; PJOB->ORDON=99;
 539    2          PJOB->DATEDUE=5; PJOB->DATEFIN=100;
 541    2          PJOB->LIEN=TLJOB; TLJOB=PJOB;
 543    2          /* ALLOCATION DES OPERATIONS BIDONS CORRESPONDANTES */
        2          OPPIC=00000000;
 544    2          DO I=1 TO 5; ALLOCATE CELOPER SET(P);
 546    2    1        OPPIC=OPPIC+1;
 547    2    1        P->OPCH=OPCHD; P->DUROP=12-2*I;
 549    2    1        P->NBSUC=0;P->NBPRED=0;
 551    2    1        P->TPTSUC(1)=P; P->PTJOB=PJOB;
 553    2    1        P->DDEFF=24*(I-1)+1; P->DFOPER=P->DDEFF+P->DUROP-1;
 555    2    1        PJOB->TPOPJOB(I)=P;
 556    2    1     END;
 557    2          /*  INIT DES LISTOPMACH   */
        2          DO I=1 TO NBMACH;
 558    2    1        ALLOCATE LISTOPMACH SET(Q);
 559    2    1        TPLIST(I)=Q; Q->POPER=PJOB->TPOPJOB(1);Q->FIXE='1'B;PRED=Q;
 563    2    1      DO J=2 TO 5; ALLOCATE LISTOPMACH SET(Q);
 565    2    2         Q->POPER=PJOB->TPOPJOB(J);
 566    2    2         Q->FIXE='1'B; PRED->SUIV=Q; PRED=Q;
 569    2    2      END;
 570    2    1     END;
 571    2
        2          END INITCHARG;
 572    1
```

```
        2       REMPLISUC: DO I=1 TO NOPJ;P=Q->TPOPJOB(I);
 610    2   1                   DO J=1 TO P->NBSUC;
 611    2   2                      DO K=1 TO NOPJ;X=Q->TPOPJOB(K);
 613    2   3                      IF SUBSTR(TABCH(I),8*J+1,8)=X->OPCH
        2   3                      THEN P->TPTSUC(J)=X;
 615    2   3                      ELSE;
 616    2   3                      END;
 617    2   2                   END;
 618    2   1       END REMPLISUC;
 619    2           REMPLIPRED: DO I=1 TO NOPJ; P=Q->TPOPJOB(I);
 621    2   1                    P->NBPRED=((INDEX(TABCH(I),' ')-1)/8)-(P->NBSUC+1);
 622    2   1                    DO J= 1 TO P->NBPRED;
 623    2   2                       DO K=1 TO NOPJ;X=Q->TPOPJOB(K);
 625    2   3                       IF SUBSTR(TABCH(I),8*(J+P->NBSUC)+1,8)=X->OPCH
        2   3                       THEN P->TPTPRED(J)=X;
 627    2   3                       ELSE;
 628    2   3                       END;
 629    2   2                    END;
 630    2   1       END REMPLIPRED;
 631    2           /*------------------------------------------------------*/
        2           /* CALCUL DES DATES-DUES DES OPERATIONS: DDUEOPER         */
        2           /*    ET INIT DES MARGES AMONT                          */
        2           IN3='1'B; DURTOT=0;II=0;PREUN=99999;
 635    2           DO J=1 TO NOPJ; P=Q->TPOPJOB(J);
 637    2   1           SNBSUC(J)=P->NBSUC; /*SAUVEGARDE DES NBSUC INITIAUX */
 638    2   1           SNBPRED(J)=P->NBPRED; /* SV. DES NBPRED INITIAUX     */
 639    2   1       END;
 640    2           DO J=1 TO NOPJ; P=Q->TPOPJOB(J);
 642    2   1          IF    P->NBSUC=0
        2   1          THEN DO; P->DDUEOPER=DATEDUE*24;
 645    2   2                   P->NBSUC=-1;
 646    2   2                   /* INSERTION DS LISTE (PTOS,TLOAT)   */
        2   2                   P->PTOS=TLOAT;
 647    2   2                   TLOAT=P;
 648    2   2                   II=II+1; STPOPJOB(II)=P;
 650    2   2               END;
 651    2           END;
 652    2           DO J=1 TO NOPJ; P=Q->TPOPJOB(J);
 654    2          IF   P->NBSUC=-1
        2   1      THEN  IF P->NBPRED¬=0 THEN DO K=1 TO P->NBPRED;
 657    2   2                X=P->TPTPRED(K);
 658    2   2                X->NBSUC=X->NBSUC-1;
 659    2   2             END;
 660    2   1                                 ELSE
        2   1        P->DAMONT,P->MAMONT=P->DDUEOPER-P->DUROP-(DARAT-1)*24;
 ...    2   1      ELSE;
 662    2   1       END;      /* FIN INIT PROCESSUS DE CALCUL   */
 663    2
        2           MAJDDUE: DO WHILE (INB); INB='0'B;
 665    2   1                  DO J=1 TO NOPJ; P=Q->TPOPJOB(J);
 667    2   2                     IF    P->NBSUC=-1
        2   2                     THEN;
 669    2   2                     ELSE DO; IN3='1'B;
 671    2   3                              IF P->NBSUC=0
        2   3                              THEN DO; X=P->TPTSUC(1);
 674    2   4                                      II=II+1;STPOPJOB(II)=P;
 676    2   4                                      P->DDUEOPER=X->DDUEOPER-X->DUROP;
 677    2   4                                      DO K=2 TO SNBSUC(J);
```

```
 678   2   5                                   X=P->TPTSUC(K);
 679   2   5                                  P->DDUFOPER=MIN(P->DDUEOPER,
       2   5                                   X->DDUEOPER-X->DUROP);
 680   2   5                               END;
 681   2   4                               P->NBSUC=-1;
 682   2   4                               IF  P->NBPRED=0
       2   4                               THEN DO; P->MAMONT=
 684   2   5                                     P->DDUEOPER-P->DUROP-(DARAT-1)*24;
 685   2   5                                     P->DAMONT=P->MAMONT;
 686   2   5                                    END;
 687   2   4                               ELSE DO K=1 TO P->NBPRED;
 688   2   5                                       X=P->TPTPRED(K);
 689   2   5                                       X->NBSUC=X->NBSUC-1;
 690   2   5                                    END;
 691   2   4                            END;
 692   2   3                        ELSE;
 693   2   3                      END;
 694   2   2               END;
 695   2   1   END MAJDDUE;
 696   2
       2       DO J=1 TO NOPJ; P=Q->TPOPJOB(J);
 698   2   1      P->NBSUC=SNBSUC(J); /* RESTAURATION DES NBSUC */
 699   2   1   END;
 700   2       DO J=1 TO NOPJ; P=STPOPJOB(J);
 702   2   1         /* DISTANCES AMONT DES SUC */
       2   1          DO K=1 TO P->NBSUC; X=P->TPTSUC(K);
 704   2   2             X->DAMONT=MIN(X->DAMONT,X->DDUEOPER-P->DDUFOPER-X->DUROP);
 705   2   2         END;
 706   2   1   END;
 707   2       /* APPEL DIAGRAMME MINI AU PLUS TARD    */
       2       CALL DIAGTARD(Q);
 708   2       DO I=1 TO NBMACH;
 709   2   1      APREUN=MIN(PREUN,INDEX(DIAGOPT(I),'1'B));
 710   2   1      IF APREUN#=0 THEN PREUN=APREUN;
 712   2   1   END;
 713   2       DURTOT=(DATEDUE-DAO)*24-PREUN+1;
 714   2
       2       /*______---------------------------------------*/
       2       /*      CALCUL DE CMOINS ;(DIAG. AU PLUS TOT)   */
       2       DO J=1 TO NOPJ;P=Q->TPOPJOB(J);
 716   2   1      IF P->NBPRED=0
       2   1      THEN DO; P->NBPRED=-1;
 719   2   2               P->CMOINS=(DARAT-1)*24+1;
 720   2   2           END;
 721   2   1   END;
 722   2       DO J=1 TO NOPJ; P=TPOPJOB(J);
 724   2   1       IF P->NBPRED = -1
       2   1       THEN DO K=1 TO P->NBSUC;
 726   2   2              X=P->TPTSUC(K);
 727   2   2               X->NBPRED=X->NBPRED-1;
 728   2   2            END;
 729   2   1   END;
 730   2       INC='1'B;
 731   2       MAJCMOINS: DO WHILE(INC); INC='0'B;
 733   2   1                DO J=1 TO NOPJ;P=Q->TPOPJOB(J);
 735   2   2                  IF P->NBPRED=-1 THEN;
 737   2   2                  ELSE DO; INC='1'B;
 739   2   3                         IF P->NBPRED=0
```

```
STMT LEVEL NEST
         2    3                THEN DO; X=P->TPTPRED(1);
 742     2    4                   P->CMOINS=X->CMOINS+X->DUROP;
 743     2    4                   DO K=2 TO SNBPRED(J);
 744     2    5                       X=P->TPTPRED(K);
 745     2    5                       P->CMOINS=MAX(P->CMOINS,
         2    5                        X->CMOINS+X->DUROP);
 746     2    5                   END;
 747     2    4                   P->NBPRED=-1;
 748     2    4                   IF P->NBSUC=0 THEN;
 750     2    4                   ELSE DO K=1 TO P->NBSUC;
 751     2    5                           X=P->TPTSUC(K);
 752     2    5                           X->NBPRED=X->NBPRED-1;
 753     2    5                        END;
 754     2    4                 END;
 755     2    3                  ELSE;
 756     2    3              END;
 757     2    2            END;
 758     2    1    END MAJCMOINS;
 759     2        DO J=1 TO NOPJ;
 760     2    1         X=Q->TPOPJOB(J);
 761     2    1         X->NBPRED=SNBPRED(J);/*RESTAURATION DES NBPRED */
 762     2    1   END;
 763     2        DO J=1 TO NOPJ; X=STPOPJOB(J);
 765     2    1      IF X->NBPRED=0 THEN;
 767     2    1       ELSE X->MAMONT=X->DDUEOPER-X->DUROP-X->CMOINS+1;
 768     2    1   END;
 769     2
         2        /*======================================================*/
         2        /* DIAGTARD : PROC INTERNE A CREGRAPHJOB                 */
         2        /*         -------------------------                    */
         2        /* PARAMETRES = PTRJ : POINTEUR DU JOB (CFLJOB)          */
         2        /*======================================================*/
         2
         2        DIAGTARD: PROC(PTRJ);
 770     3        DCL (PTRJ,X) POINTER,
         3             I  REAL BIN FIXED(.,.
         3            NOMACH PIC '99';
 771     3
         3        DIAGOPT=(240)'0'B;
 772     3        DO I=1 TO PTRJ->NBOPER;
 773     3    1         X=PTRJ->TPOPJOB(I);
 774     3    1         OPCHD=X->OPCH;
 775     3    1         NOMACH=OPPICD.NUMM;
 776     3    1         SUBSTR(DIAGOPT(NOMACH),X->DDUEOPER-X->DUROP+1-DAO*24,X->DUROP)=
         3    1                                           (24)'1'B;
 777     3    1   END;
 778     3
         3        END DIAGTARD;
 779     2
         2        /*======================================================*/
         2        END CREGRAPHJOB;
 780     1
         1        /*======================================================*/
         1
         1
         1
         1        /*------------------------------------------------------*/
         1        /* RESULTATS : PROC INTERNE A ORDPRODUCTION              */
```

```
STMT LEVEL NEST
 834   2    2            END;
 835   2    1         ELSE DO; Q=P; P=P->LIEN; END;
 839   2    1       END;
 840   2
       2
       2
       2            /*-----------------------------------------------------------*/
       2            /* STATGLOBAL : PROC INTERNE A RESULTATS                    */
       2            /*              ----------------------                       */
       2            /* PARAMETRES = PTJ: POINTEUR DU JOB                        */
       2            /*-----------------------------------------------------------*/
       2
       2            STATGLOBAL: PROC(PTJ);
 841   3            DCL (PTJ,PO) POINTER;
 842   3            DCL (RDUR,RDD,RDFOP,RDF) REAL BIN FLOAT(24);
 843   3            DCL (MECJOB,MECOP) REAL BIN FLOAT(24);
 844   3            DCL (RETARD,AVANCE,J) REAL BIN FIXED(31);
 845   3
       3            NJTOTAL=NJTOTAL+1;
 846   3            /* STAT SUR DATES DUES */
       3            RETARD=MAX(0,PTJ->DATEFIN-PTJ->DATEDUE*24);
 847   3            TRETARD=TRETARD+RETARD;
 848   3            IF   RETARD>0
       3            THEN NJENRETARD=NJENRETARD+1;
 850   3            ELSE NJOK=NJOK+1;
 851   3            AVANCE=MAX(0,PTJ->DATEDUE*24-PTJ->DATEFIN);
 852   3            TAVANCE=TAVANCE+AVANCE;
 853   3            IF   AVANCE>0
       3            THEN NJENAVANCE=NJENAVANCE+1;
 855   3            ELSE;
 856   3
       3            /* STAT SUR EN-COURS */
       3            MECJOB=0;
 857   3            DO J=1 TO PTJ->NBOPER;
 858   3    1          PO=PTJ->TPOPJOB(J);
 859   3    1          RDUR=PO->DUROP; RDD=PTJ->DATEDUE*24;
 861   3    1          RDFOP=PO->DFOPER; RDF=PTJ->DATEFIN;
 863   3    1          IF PTJ->DATEFIN<=PTJ->DATEDUE*24
       3    1          THEN MECOP=RDUR*(RDD-(RDFOP-RDUR/2.0));
 865   3    1          ELSE MECOP=RDUR*(RDF-(RDFOP-RDUR/2.0));
 866   3    1          MECJOB=MECJOB+MECOP;
 867   3    1       END;
 868   3            PUT SKIP EDIT ('MONTANT-EN-COURS-JOB NO ',PTJ->NOJOB,' = ',MECJOB)
       3                         (X(30),A,A(4),A,F(10,2));
 869   3            MTEC=MTEC+MECJOB;
 870   3            PUT SKIP EDIT ('RETARD =',RETARD,'AVANCE =',AVANCE,'INDRET=',PTJ->INDRET)
       3                         (X(35),A,F(6),A,F(6),A,B(1));
 871   3
       3            END STATGLOBAL;
 872   2
       2            /*-----------------------------------------------------------*/
       2
       2            END RESULTATS;
 873   1
       1            /*-----------------------------------------------------------*/
       1
       1            /*  ========================================================*/
```

```
STMT LEVEL NEST
         1             /*----------------------------------------------------------*/
         1
         1             MAJDIST: PROC;
 874     2
         2             DCL   (IM,I) REAL BIN FIXED(31),
         2                   (P,Q,X)    POINTER;
 875     2
         2             DO IM=1 TO NBMACH;
 876     2     1          Q=TPLIST(IM);
 877     2     1          DO WHILE(Q#=NULL);
 878     2     2             /* M.A.J. DES DISTANCES DE OPER PTFE PAR X  */
         2     2             X=Q->POPER;
 879     2     2             IF X->NBPRED=0 THEN X->DAMONT=X->MAMONT;
 881     2     2               ELSE DO;
 882     2     3                       X->DAMONT=9999;
 883     2     3                       DO I=1 TO X->NBPRED;
 884     2     4                          P=X->TPTPRED(I);
 885     2     4                          X->DAMONT=MIN(X->DAMONT,X->DDEFF-P->DFOPER-1)
 886     2     4                       END;
 887     2     3                    END;
 888     2     2             IF X->NBSUC=0 THEN X->DAVAL=MAX(0,X->DDUEOPER-X->DFOPER)
 890     2     2              ELSE DO;
 891     2     3                      X->DAVAL=9999;
 892     2     3                      DO I=1 TO X->LBSUC;
 893     2     4                         P=X->TPTSUC(I);
 894     2     4                       X->DAVAL=MIN(X->DAVAL,P->DDEFF-X->DFOPER-1);
 895     2     4                      END;
 896     2     3                   END;
 897     2     2             /* EDITIONS  */
         2     2             Q=Q->SUIV;
 898     2     2          END;
 899     2     1       END;
 900     2
         2             END MAJDIST;
 901     1
         1             /*=====================================================*/
         1             /* CALCRIO : PROC INTERNE A ORDPRODUCTION              */
         1             /*                   ------------------------          */
         1             /*                                                     */
         1             /* CALCULE CRITRE CRIO POUR TOUTES LES OPERS SANS SUC  */
         1             /* DE LA LISTE :TLOAT,PTOS; RENVOIE 2 VALEURS DE PTEURS */
         1             /*            --------------------------------         */
         1             /* PARAMETRES: POAC= PTEUR OPER A CHARGER              */
         1             /*             POPAC= PTEUR OPER PRECED                */
         1             /*=====================================================*/
         1
         1             CALCRIO: PROC(POAC,POPAC);
 902     2             DCL (POC,POAC,POPAC,PRECED,X) POINTER,
         2                  UNB BIT(24) INIT((24)'1'B),
         2                 I REAL BIN FIXED(31),
         2                (MINNEMP,NEMP) REAL BIN FIXED(31),
         2                 MACH PIC '99',
         2                 DIAGOPT BIT(240);
         2
 903     2              POC=TLOAT;MINNEMP=99999;PRECED=NULL;
         2              DO WHILE (POC#=NULL); /* TRAITER OPER PTEE PAR POC */
 906     2     1          /* CALCUL DE CPLUS */
 907     2     1           OPCHD=POC->OPCH;
         2     1           MACH=NUMM;
 909     2     1
```

```
STMT LEVEL NEST
 909   2    1       DIAGOPT=(240)'0'B;
 910   2    1       IF POC->NBSUC=0
       2    1       THEN POC->CPLUS=POC->DDUFOPER-POC->DUROP+1;
 912   2    1       ELSE DO;
 913   2    2             POC->CPLUS=99999;
 914   2    2             DO I=1 TO POC->NBSUC; X=POC->TPTSUC(I);
 916   2    3                POC->CPLUS=MIN(POC->CPLUS,X->DDEFF-POC->DUROP);
 917   2    3             END;
 918   2    2             END;
 919   2    1
       2    1     /* CALCUL DU NBRE D EMPLACEMENTS POSSIBLES */
       2    1       NEMP=0;
 920   2    1       SUBSTR(DIAGOPT,POC->CMOINS-DAO*24,POC->DUROP)=SUBSTR(UNB,1,POC->DUROP);
 921   2    1       DO I=POC->CMOINS-DAO*24 TO POC->CPLUS-DAO*24;
 922   2    2          IF BOOL(DIAGOPT,CHARMACH(MACH),'0001'B)
       2    2          THEN;
 924   2    2          ELSE NEMP=NEMP+1;
 925   2    2          DIAGOPT='0'B||DIAGOPT;
 926   2    2       END;
 927   2    1
       2    1     /*    CHOIX      */
       2    1       IF NEMP<MINNEMP THEN DO; MINNEMP=NEMP;
 930   2    2                               POAC=POC;POPAC=PRECED;
 932   2    2                               END;
 933   2    1       ELSE IF NEMP=MINNEMP THEN
       2    1                  IF POC->DUROP>POAC->DUROP
       2    1                     THEN DO; MINNEMP=NEMP;
 937   2    2                             POAC=POC;POPAC=PRECED;
 939   2    2                             END;
 940   2    1
       2    1    /* PROG DS LISTE   */
       2    1     POC->CRIO=NEMP;
 941   2    1     PRECED=POC;
 942   2    1     POC=POC->PTOS;
 943   2    1  END;
 944   2       END CALCRIO;
 945   1
       1       /*-----------------------------------------------------*/
       1
       1       /*=========================================================*/
       1       /* TRAITCHARGES : PROC INTERNE A ORDPRODUCTION            */
       1       /*         --------------------------------               */
       1       /* PARAMETRES = PTROPER: POINTEUR DE OPER A CHARGER       */
       1       /*                PTRPREC: PTEUR DE OPER PREC              */
       1       /*=========================================================*/
       1
       1       TRAITCHARGES: PROC(PTROPER,PTRPREC);
 946   2       DCL DIAGOPT(NBMACH) BIT(240),
       2           SDIAGOPT BIT(240),
       2           HISTOPT(NBMACH,10) BIT(24) DEF DIAGOPT;
 947   2       DCL (PTRPREC,XS,PTROPER,X,PO,SUC) POINTER,
       2           MACH PIC '99',
       2            UNB BIT(24) INIT((24)'1'B),
       2           (AVANCE,DEMOPT,II,JJ) REAL BIN FIXED(31),
       2           (ORDPOS,INSERER) BIT(1);
 948   2
       2       /* RECHERCHE DE LA MACHINE A CHARGER : MACH */
       2       OPCHD=PTROPER->OPCH;
```

```
STMT LEVEL NEST
 949    2        MACH=NUMM;
 950    2        PO=PTROPER;
 951    2                 /* CALCUL DU DIAG AU PLUS TARD  */
        2                  DIAGOPT(MACH)=(240)'0'B;
 952    2               PO->DPL=9999;
 953    2               DO I=1 TO PO->NBSUC; SUC=PO->TPTSUC(I);
 955    2     1         PO->DPL=MIN(PO->DPL,SUC->DDEFF-PO->DDUEOPER-1);
 956    2     1         END;
 957    2               IF PO->DPL=9999 THEN PO->DPL=0;
 959    2               SUBSTR(DIAGOPT(MACH),PO->DDUEOPER-PO->DUROP+1+PO->DPL
        2                 -DAO*24,PO->DUROP)=(24)'1'B;
 960    2              /* REINIT MARGES ET DISTANCES */
        2               PO->MAMONT=PO->MAMONT+PO->DPL;
 961    2               PO->MAVAL=PO->MAVAL-PO->DPL;
 962    2                     /*CHARGEMENT DE L'OPERATION */
        2                      /* TEST:OPER PEUT-ELLE RESTER A SA PLACE? */
        2                      IF BOOL(DIAGOPT(MACH),CHARMACH(MACH),'0001'B)
        2                      THEN DO;/*OPER NE PEUT PAS RESTER A SA PLACE */
 964    2     1                      CALL DEPLACEAMONT(PO,MACH,ORDPOS);
 965    2     1                      IF (¬ORDPOS)
        2     1                      THEN DO; /* ORDON AMONT IMPOSSIBLE */
 967    2     2                            /* ORDONNANCEMENT AVAL */
        2     2                           PO->PTJOB->INDRET='1'B;
 968    2     2                           CALL ORDONAVAL(PO,MACH);
 969    2     2                           END;
 970    2     1                      ELSE;
 971    2     1                     END;
 972    2                      ELSE DO;/* OPER PEUT RESTER A SA PLACE */
 973    2     1                      PO->DDEFF=INDEX(DIAGOPT(MACH),'1'B)+DAO*24;
 974    2     1                      PO->DFOPER=PO->DDEFF+PO->DUROP-1;
 975    2     1                      CHARMACH(MACH)=
        2     1                       BOOL(CHARMACH(MACH),DIAGOPT(MACH),'0110'B);
 976    2     1                      CALL INSERTLISTOP(MACH,PO);
 977    2     1                     END;
 978    2
        2        PO->ORDFAIT='1'B;
 979    2                /* SORTIE DE LA LISTE POUR OPER CHARGEE  */
        2                IF PTRPREC=NULL THEN TLOAT=PTROPER->PTOS;
 981    2                                ELSE PTRPREC->PTOS=PTROPER->PTOS;
 982    2                /* INSERTION DS LISTE DES OPERS PRECED SI ORDFAIT */
        2                IF PO->NBPRED¬=0
        2                    THEN DO II =1 TO PO->NBPRED;
 984    2     1                      X=PO->TPTPRED(II);INSERER='1'B;
 986    2     1                      DO JJ=1 TO X->NBSUC;
 987    2     2                           XS=X->TPTSUC(JJ);
 988    2     2                           IF XS->ORDFAIT
        2     2                           THEN;
 990    2     2                           ELSE INSERER='0'B;
 991    2     2                      END;
 992    2     1                      IF INSERER THEN DO; X->PTOS=TLOAT;
 995    2     2                                          TLOAT=X;
 996    2     2                                      END;
 997    2     1                   END;
 998    2        /*=========================================================*/
        2        /* INSERTLISTOP : PROC INTERNE A TRAITCHARGES              */
        2        /*                ----------------------------             */
        2        /* PARAMETRES = MACH : NUMERO DE LA MACHINE                */
        2        /*              PO:   PTEUR OPER                          */
```

```
STMT LEVEL  NEST
1040   3     2          P0->DAMONT=P0->DAMONT-AVANCE;
1041   3     2          P0->MAVAL=P0->MAVAL+AVANCE;
1042   3     2          P0->DAVAL=P0->DAVAL+AVANCE;
1043   3     2          P0->DPL=P0->DPL-AVANCE;
1044   3     2          P0->DDEFF=INDEX(DIAGOPT(MACH),'1'B)+DA0*24;
1045   3     2          P0->DFOPER=P0->DDEFF+P0->DUROP-1;
1046   3     2          CALL INSERTLISTOP(MACH,P0);
1047   3     2          /* MODIF DES MARGES POUR OPER DIRECT. SUIVANTES */
       3     2          P1=P0;
1048   3     2          IF   P1->NBSUC¬=0
       3     2          THEN DO K=1 TO P1->NBSUC;
1050   3     3                T=P1->TPTSUC(K);
1051   3     3                /*T->MAMONT INCHANGEE */
       3     3                /*T->MAVAL    ---     */
       3     3             END;
1052   3     2          ELSE;
1053   3     2
       3     2          ORDPOS='1'B;
1054   3     2          END;
1055   3     1      END;
1056   3
       3            END DEPLACEAMONT;
1057   2
       2            /*-----------------------------------------------------*/
       2
       2            /*------------------------------------------------------*/
       2            /* ORDONAVAL:  PROC INTERNE A TRAITCHARGES.             */
       2            /*             ---------------------                    */
       2            /* PARAMETRES =  P0: PTEUR OPER A RETARDER              */
       2            /*             MACH: MACHINE CONCERNEE                  */
       2            /*------------------------------------------------------*/
       2
       2            ORDONAVAL: PROC (P0,MACH);
1058   3            DCL
       3                (P0,PRED,P1,T,SUIVM) POINTER,
       3                 DOPPIC PIC '99999999',
       3                 DOPCHD CHAR(8) DEF DOPPIC,
       3                1 DOPPICD DEF DOPPIC,
       3                  2 DNUMJ PIC '9999',
       3                  2 DNUMM PIC '99',
       3                  2 DNUMO PIC '99',
       3                 (L,K,RETARD) REAL BIN FIXED(31),
       3                (ORDPOS,OK,MODIF) BIT(1),
       3                 MACH PIC '99',
       3                 ZERO BIT(100);
       3
1059   3             ZERO=(100)'0'B;
       3             DIAGOPT(MACH)=SDIAGOPT;
1060   3             /* RECH. SI OPER SUIV. SUR MEME MACHINE */
1061   3             SUIVM=NULL;
       3             IF P0->NBSUC¬=0
1062   3             THEN DO K=1 TO P0->NBSUC;
       3                    T=P0->TPTSUC(K); DOPCHD=T->OPCH;
1064   3     1              IF MACH=DNUMM THEN DO;SUIVM=T;SUIVM->IREG='0';END;
1066   3     1                                  ELSE;
1071   3     1          END;
1072   3     1       ELSE;
1073   3             RETARD=0; ORDPOS='0'B;
1074   3
```

```
CARTE   STMT LEVEL  NEST
 1335   1076    3          PO->MAMONT=PO->MAMONT+AVANCE;
 1336   1077    3          E1: DO WHILE (¬ORDPOS);
 1337   1078    3     1          RETARD=RETARD+1;
 1338   1079    3     1          IF SUIVM¬=NULL
 1339           3     1          THEN
 1340           3     1            IF (SUIVM->DDEFF<=PO->DDUEOPER+PO->DPL+RETARD)&(SUIVM->IREG¬='A')
 1341           3     1            THEN DO;
 1342   1082    3     2              SUBSTR(CHARMACH(NUMM),SUIVM->DDEFF-DAO*24,SUIVM->DUROP)
 1343           3     2                   =SUBSTR(ZERO,1,SUIVM->DUROP);
 1344   1083    3     2              SUIVM->IREG='A';OK='0'B;
 1345   1085    3     2              IF SUIVM->NBSUC¬=0
 1346           3     2               THEN DO K=1 TO SUIVM->NBSUC WHILE (¬OK);
 1347   1087    3     3                     T=SUIVM->TPTSUC(K);DOPCHD=T->OPCH;
 1348   1089    3     3                     IF MACH=DNUMM THEN DO;SUIVM=T;OK='1'B;
 1349   1093    3     4                                          SUIVM->IREG='0';
 1350   1094    3     4                                        END;
 1351   1095    3     3                   END;
 1352   1096    3     2               ELSE SUIVM=NULL;
 1353   1097    3     2                 END;
 1354   1098    3     1            ELSE;
 1355   1099    3     1          ELSE;
 1356   1100    3     1          SUBSTR(DIAGOPT(MACH),INDEX(DIAGOPT(MACH),'1'B),PO->DUROP+1)=
 1357           3     1                                   '0'B||(23)'1'B;
 1358   1101    3     1          IF  BOOL(DIAGOPT(MACH),CHARMACH(MACH),'0001'B)
 1359           3     1          THEN;
 1360   1103    3     1          ELSE DO;
 1361   1104    3     2                ORDPOS='1'B;
 1362   1105    3     2                CHARMACH(MACH)=BOOL(CHARMACH(MACH),DIAGOPT(MACH),'0110'B);
 1363   1106    3     2                PO->MAMONT=PO->MAMONT+RETARD;
 1364   1107    3     2                PO->MAVAL=PO->MAVAL-RETARD;
 1365   1108    3     2                PO->DPL=PO->DPL+RETARD;
 1366   1109    3     2                PO->DDEFF=INDEX(DIAGOPT(MACH),'1'B)+DAO*24;
 1367   1110    3     2                PO->DFOPER=PO->DDEFF+PO->DUROP-1;
 1368   1111    3     2                CALL INSERTLISTOP(MACH,PO);
 1369   1112    3     2
 1370           3     2                /* MODIF EVENT. DES OPER SUIVANTES  */
 1371           3     2                  MODIF='0'B;
 1372   1113    3     2                  IF PO->NBSUC¬=0 THEN
 1373           3     2                   DO;
 1374   1115    3     3                    DO K=1 TO PO->NBSUC WHILE(¬MODIF);
 1375   1116    3     4                       T=PO->TPTSUC(K);
 1376   1117    3     4                       IF T->DDEFF<=PO->DFOPER THEN MODIF='1'B;
 1377   1119    3     4                    END;
 1378   1120    3     3                    IF MODIF THEN DO;
 1379   1122    3     4                                    P1=PO;
 1380   1123    3     4                                    CALL MOS(P1);
 1381   1124    3     4                                 END;
 1382   1125    3     3                  END;
 1383   1126    3     2              END;
 1384   1127    3     1    END E1;
 1385   1128    3
 1386           3          /*-----------------------------------------------*/
 1387           3          /*  MOS: PROC INTERNE A ORDONAVAL (RECURSIVE)    */
 1388           3          /*        ------------------                      */
 1389           3          /* PARAM:  PA: POINTEUR OPER A TRAITER            */
 1390           3          /*-----------------------------------------------*/
 1391           3
 1392           3          MOS: PROC(PA) RECURSIVE
```

```
CARTE   STMT LEVEL NEST
1451    1165   5         /* RECH. SI OPER SUIV. SUR MEME MACHINE */
1452           5         SUIVM=NULL;
1453    1166   5         IF POP->NBSUC¬=0
1454           5         THEN DO K=1 TO POP->NBSUC;
1455    1168   5    1            T=POP->TPTSUC(K); DOPCHD=T->OPCH;
1456    1170   5    1            IF MACH=DNUMM THEN DO;SUIVM=T;SUIVM->IREG='0';
1457    1174   5    2                                  END;
1458    1175   5    1                              ELSE;
1459    1176   5    1       END;
1460    1177   5         ELSE;
1461    1178   5         DO WHILE (¬ORDPOS);
1462    1179   5    1       RETARD=RETARD+1;
1463    1180   5    1        IF SUIVM¬=NULL
1464           5    1        THEN
1465           5    1         IF (SUIVM->DDEFF<=POP->DDUEOPER+POP->DPL+RETARD)&(SUIVM->IREG¬='A')
1466           5    1          THEN DO;
1467    1183   5    2           SUBSTR(CHARMACH(NUMM),SUIVM->DDEFF-DAO*24,SUIVM->DUROP)
1468           5    2                =SUBSTR(ZERO,1,SUIVM->DUROP);
1469    1184   5    2           SUIVM->IREG='A'; OK='0'B;
1470    1186   5    2           IF SUIVM->NBSUC¬=0
1471           5    2            THEN DO K=1 TO SUIVM->NBSUC WHILE (¬OK);
1472    1188   5    3                   T=SUIVM->TPTSUC(K);DOPCHD=T->OPCH;
1473    1190   5    3                   IF MACH=DNUMM THEN DO;SUIVM=T;SUIVM->IREG='0';
1474    1194   5    4                                          OK='1'B;
1475    1195   5    4                                     END;
1476    1196   5    3                 END;
1477    1197   5    2            ELSE SUIVM=NULL;
1478    1198   5    2               END;
1479    1199   5    1         ELSE;
1480    1200   5    1        ELSE;
1481    1201   5    1      SUBSTR(DIAGOPT(MACH),INDEX(DIAGOPT(MACH),'1'B),POP->DUROP+1)=
1482           5    1                                    '0'B||(23)'1'B;
1483    1202   5    1      IF BOOL(DIAGOPT(MACH),CHARMACH(MACH),'0001'B)
1484           5    1      THEN;
1485    1204   5    1      ELSE DO;
1486    1205   5    2           CHARMACH(MACH)=BOOL(CHARMACH(MACH),DIAGOPT(MACH),'0110'B);
1487    1206   5    2           POP->MAMONT=POP->MAMONT+RETARD;
1488    1207   5    2           POP->MAVAL=POP->MAVAL-RETARD;
1489    1208   5    2              POP->DPL=POP->DPL+RETARD;
1490    1209   5    2           POP->DDEFF=INDEX(DIAGOPT(MACH),'1'B)+DAO*24;
1491    1210   5    2           POP->DFOPER=POP->DDEFF+POP->DUROP-1;
1492    1211   5    2           CALL INSERTLISTOP(MACH,POP);
1493    1212   5    2           ORDPOS='1'B;
1494    1213   5    2           END;
1495    1214   5    1    END;
1496    1215   5
1497           5         END DECALEAVAL;
1498    1216   4
1499           4         /*--------------------------------------------------------*/
1500           4
1501           4         /*--------------------------------------------------------*/
1502           4         /* SUPPRESLISTOP: PROC INTERNE A   MOS                    */
1503           4         /*             ----------------------                     */
1504           4         /* PARAMETRES =  MACH:  NO MACHINE                        */
1505           4         /*               PTO:  PTEUR OPER A SUPPRIMER DE LA LISTE */
1506           4         /*--------------------------------------------------------*/
1507           4
1508           4         SUPPRESLISTOP: PROC(MACH,PTO);
```

```
STMT LEVEL NEST      * * * *   ORDPRODUCTION  * * * *
1217    5           DCL  MACH PIC '99',
        5                (PTO,QTR,RTR) POINTER;
1218    5
        5           QTR=TPLIST(MACH);
1219    5           RTR=NULL;
1220    5           DO WHILE (QTR->POPER*=PTO);
1221    5     1          RTR=QTR;
1222    5     1          QTR=QTR->SUIV;
1223    5     1     END;
1224    5           IF  RTR*=NULL
        5           THEN RTR->SUIV=QTR->SUIV;
1226    5           ELSE TPLIST(MACH)=QTR->SUIV;
1227    5           FREE QTR->LISTOPMACH;
1228    5
        5           END SUPPRESLISTOP;
1229    4
        4           /*-------------------------------------------------------*/
        4           END MOS;
1230    3
        3           /*-------------------------------------------------------------*/
        3           END ORDONAVAL;
1231    2
        2           /*-------------------------------------------------------------*/
        2
        2
        2           END TRAITCHARGES;
1232    1
        1           /*************************************************************/
        1
        1           END ORDPRODUCTION;
```