THESIS / THÈSE

MASTER EN SCIENCES ÉCONOMIQUES

L'ordonnancement non unitaire, problème heuristique

Kervyn de Meerendré, Daniel

Award date:
1969

Awarding institution:
Universite de Namur

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX-NAMUR

FACULTE DES SCIENCES ÉCONOMIQUES ET SOCIALES

Année Académique 1968-1969

L'ORDONNANCEMENT NON UNITAIRE, PROBLEME HEURISTIQUE

Daniel KERVYN de MEERENDRÉ

Mémoire présenté en vue de l'obtention
du Grade de Licencié en Sciences Economiques
et Sociales (Gestion d'entreprise)

Jury du Mémoire :

MM. F BODART

L. DERWA

A mes parents,
A ma fiancée,

"Ce qu'aiment les hommes, ce que tu aimes, ce n'est pas connaître, ce n'est pas savoir : c'est osciller entre deux vérités ou deux mensonges"(Giraudoux)

"Les étoiles sont si grandes, la terre si petite,
restez comme vous êtes"
(Marshall Mc Luhan)

AVANT - PROPOS

L'idée qui a suscité ce travail revient au professeur F.BODART qui m'y a dirigé avec beaucoup de compétence et de compréhension ; cette étude n'aurait pu être poursuivie et menée à son terme sans son aide et ses encouragements ; qu'il veuille bien trouver ici l'expression de toute ma gratitude.

Le professeur L.DERWA l'y a secondé avec dévouement et sympathie ; sans son sens critique et ses suggestions, je n'aurais pu terminer ce mémoire ; je l'en remercie très sincèrement.

Mes remerciements vont également à tous ceux qui ont bien voulu me faire part de leurs remarques. Je pense particulièrement aux professeurs M.BOURGEOIS, des H.E.C. à Paris et à P.A.VERHEYEN de l'Université Catholique de Tilburg.

Je voudrais enfin remercier le personnel du centre d'informatique, qui a permis, par son aide et sa patience, l'élaboration et l'exécution de mes programmes. Que tous ceux que j'omets trouvent ici l'expression de ma reconnaissance.

D. KERVYN de Meerendré

TABLE DES MATIERES :

L'ORDONNANCEMENT NON UNITAIRE, PROBLEME HEURISTIQUE

INTRODUCTION

Intéressé par l'aspect pratique et concret des méthodes de recherche opérationnelle, nous avons voulu les appliquer à un problème particulier d'entreprise : celui de l'ordonnancement non unitaire.

Les questions que nous nous posions et auxquelles nous avons tenté de répondre sont les suivantes :

- Le problème d'ordonnancement est-il un problème économique ?

  La première partie de ce mémoire essaie de répondre à cette question en la subdivisant :

  - où ce problème se situe-t-il dans l'entreprise ?
  - qu'entend-on, dans une entreprise, par "problème d'ordonnancement ?
  - la résolution de ce problème a-t-il une signification économique ? Dans une optique cybernétique de l'entreprise, cette signification nous apparut évidente par les imbrications d'un aspect ou d'un problème avec tous ceux qui l'entourent.

- Comment le problème est-il résolu ? Cette solution est-elle satisfaisante ?

  La réponse fait l'objet de la seconde partie. Comme la plupart des ouvrages traitant de l'ordonnancement sont l'oeuvre de mathématiciens, qui se sont dégagés du contexte économique du problème qu'ils étudiaient, nous avons essayé de les critiquer en nous situant au niveau de l'entreprise et au niveau de l'utilisation pratique des résultats obtenus (en supposant que ces résultats existent).

- Comment sortir de l'impasse ?

  La découverte d'un fossé particulièrement profond entre des solutions particulières et mathématiques et entre une solution générale et économique nous a entraîné à reprendre le problème dans son ensemble, en le considérant comme un exemple de processus heuristiques de décision. La troisième partie a pour objectif de lier étroitement l'aspect économique de la première partie et l'aspect technique de la seconde.

A la fin de ce mémoire, nous aurons parcouru une boucle : en partant de l'entreprise en général et après avoir passé les solutions "algorithmiques" de l'ordonnancement en revue et après les avoir rejetées parce qu'elles ne respectent pas l'idée de "l'entreprise-système", nous aurons abouti à une solution tenant compte et des aspects particuliers du problème et de l'entreprise considérée comme un tout.

TITRE I :
L'ORDONNANCEMENT DANS L'ENTREPRISE

Cette première partie a pour objet d'étudier la manière dont l'ordonnancement s'insère dans l'entreprise et son importance tant académique qu' économique.

L'entreprise se définissant en effet comme un système - ensemble d'éléments fortement interdépendants les uns des autres -, il est impossible de situer l'ordonnancement hors de son contexte.

Dans un premier chapitre, nous donnerons une définition de l'entreprise et nous examinerons rapidement les deux théories auxquelles cette définition a donné lieu. Elles aboutissent à un schéma d'entreprise, insistant sur les interdépendances entre les éléments qui composent l'entreprise et sur la nécessité de la prise de décision.

Dans un second chapitre, : la prise de décision pose le problème général de la structure de l'entreprise; une fois définie, elle situera la place du département de "planning et contrôle de production".

Dans un troisième chapitre, ce département sera disséqué : lorsque nous aurons déterminé les fonctions à remplir par ses responsables, nous en déduirons sa structure particulière, à l'intérieur de laquelle nous trouverons l'ordonnancement.

Dans un quatrième chapitre, nous définirons l'ordonnancement lui-même, ses différents types et les problèmes qu'il entraîne.

Le dernier chapitre insistera sur l'importance économique revêtue par sa résolution, du point de vue de l'ordonnancement lui-même et du point de vue des implications de la solution sur son environnement.

Nous aurons alors parcouru un double mouvement :

- un mouvement de convergence : de l'entreprise vers l'ordonnancement : où situer l'ordonnancement ? (chapitres 1 à 3),
- point central : définition de l'ordonnancement (chapitre 4),
- un mouvement de divergence : de l'ordonnancement vers l'entreprise : comment la résolution du problème d'ordonnancement réagit-elle (mouvement de "feedback") sur les autres facettes de l'entreprise ? (chapitre 5)

Chapitre 1 :

L'entreprise, unité de production

Section 1 : Définition et finalité de l'entreprise

Une entreprise est une entité économique de production.

La "production" est conçue comme un processus quelconque, une manière de transformer un ensemble d'éléments d'inputs en un ensemble spécifique d'outputs.

Le vocable "économique" souligne la finalité de l'entreprise et suggère l'utilisation judicieuse des ressources, c'est-à-dire la recherche du meilleur parti à tirer de moyens rares; comme ces ressources sont autant matérielles qu'humaines, le terme "économique" a des accents autant techniques que sociaux.

Au vu de cette définition, les problèmes qui se posent à l'entreprise ne sont pas purement internes. L'entreprise vit dans un milieu complexe et mouvant. Elle est rattachée au monde extérieur, du moins par ses deux extrémités : d'une part, par l'appel qu'elle fait aux facteurs de production, d'autre part, par son souci de mettre des produits et des services au point et de leur découvrir des débouchés. Se trouvant "prisonnière" d'un vaste réseau d'interdépendances, il ne s'agit pas pour l'entreprise de subir les heurts du monde extérieur sans essayer de voir clair dans ce qui se passe : s'il est important de savoir ce qui se passerait si les conditions restaient "normales", il est bien plus important de savoir, lorsque l'inattendu se produit, quelles sont les conséquences de son apparition.

Section 2 : Théories de l'entreprise [1]

La section précédente mérite quelques développements parce qu'elle témoigne d'un renversement de situation historique en ce qui concerne la signification de l'expression : "utilisation judicieuse".

§ 1. La théorie classique (que l'on situe d'Adam Smith à Cournot : 1838)

"... l'entreprise demeure une entité bien vague; quant à l'entrepreneur, il ne cesse d'être vague que pour devenir assez douteux ! Dès qu'une possibilité de profit se révèle, le capital et le travail, comme aspirés par le vide, affluent de toutes parts et s'organisent d'eux-mêmes apparemment sans effort et sans aide, de façon à créer le processus de production. Quand les profits et les salaires baissent, le capital et le travail s'écartent de cette activité peu rentable et se dirigent voracement vers de nouveaux pâturages. L'entreprise est donc considérée comme une sorte d'agrégat de capital et de travail, et non pas comme une organisation ..." [2].

La théorie classique se basait sur les hypothèses de maximisation du profit au sens strict et de concurrence parfaite. Dans cette théorie, les contraintes externes limitaient le comportement de l'entreprise et réduisaient les décisions de l'entrepreneur au seul choix des activités qu'il pouvait entreprendre. Le marché éliminait lui-même les entreprises qui n'étaient pas organisées de façon optimale; il n'y avait pas de théorie de l'information; elle n'avait d'ailleurs aucune utilité dans une économie parfaitement concurrentielle où tout est supposé connu de tous : les prix des facteurs et des produits étant donnés, l'entrepreneur devait simplement organiser son affaire de façon à maximiser son profit.

---

[1] BOULDING K.E. et SPIVEY W.A. : "La programmation linéaire et la théorie de l'entreprise", Dunod 1964, chapitres 1,6,7.

[2] BOULDING K.E. et SPIVEY W.A. : op. cit. p. 1

J.P. Miller [1] résumait en disant que la théorie classique admettait implicitement les hypothèses suivantes :

1. l'hypothèse de stationnarité selon laquelle les besoins, les ressources et les connaissances sont données une fois pour toutes;
2. l'hypothèse d'indépendance selon laquelle les besoins, les ressources et les connaissances ne dépendent pas les unes des autres, ni des actions des entreprises;
3. l'hypothèse de motivation selon laquelle le but des entreprises est la maximisation des bénéfices nets;
4. l'hypothèse d'information selon laquelle il existe un "système organisé" qui répartit les renseignements utiles à l'entreprise, que ces renseignements soient techniques, humains, commerciaux etc...;
5. l'hypothèse d'organisation selon laquelle il existe une organisation permettant aux différentes décisions prises, d'être cohérentes et compatibles avec le but de maximisation du profit que s'est fixée l'entreprise.

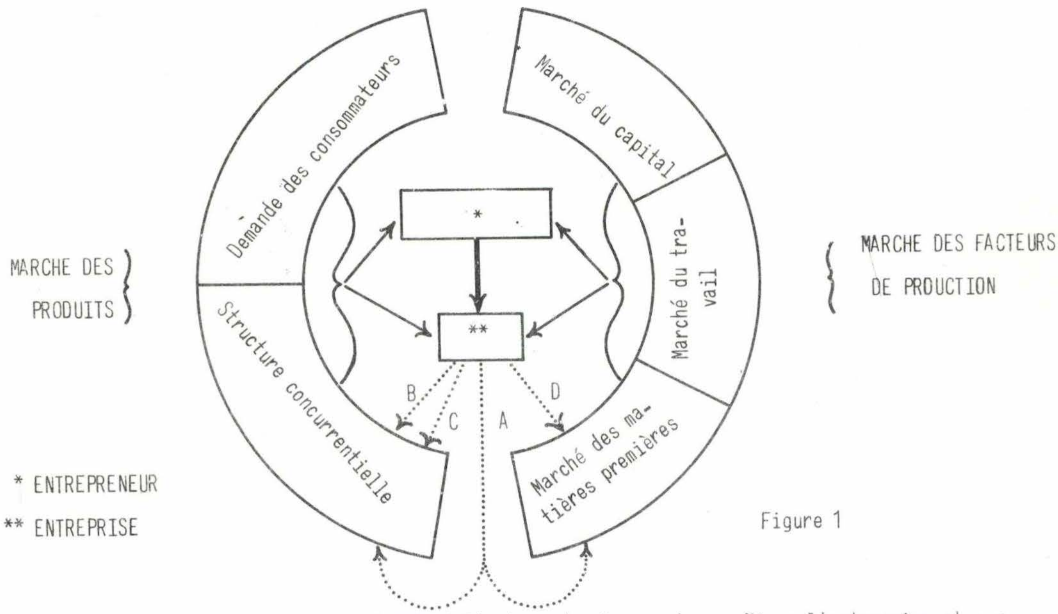


Figure 1

Les lignes en trait plein représentent l'action des forces du marchésur l'entreprise et sur l'entrepreneur. Les lignes en pointillé indiquent les limites de l'activité de l'entreprise. La ligne A est le chemin correspondant à une bonne décision d'innovation. Les lignes B,C et D représentent des essais infructueux d'innovation.

---

[1] MILLER J.P. : "Contributions of Industrial and Human Relations to Economists Theory of the Firm"; Industrial Relations Research Association Proceedings, vol. 11, p. 135-136

"La situation d'une entreprise est schématisée par la figure 1, tandis que la figure 2 illustre le processus d'élaboration des décisions, telles que les décrit la théorie traditionnelle.

Décisions	Origine
1. Que produire ?	1. Entrepreneur
2. Comment produire ?	2. Marché des facteurs de Production
3. Combien produire ?	3. Marché des produits
4. A quel prix ?	4. Marché des produits

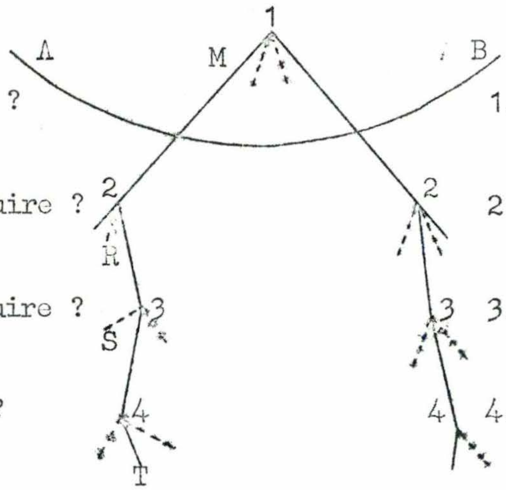


Figure 2

Fig. 2 : les lignes en pointillé représentent les éventualités que n'autorisent pas les contraintes du marché. A gauche, les décisions à prendre, à droite, l'identité des personnes prenant les décisions. Seules, les décisions situées au-dessus de la ligne AB sont prises par les membres de l'entreprise. Pour comprendre la façon dont sont prises les autres décisions, c'est le fonctionnement du marché et non pas l'organisation de l'entreprise qu'il faut étudier ; l'entrepreneur est libre de faire le choix de l'industrie dans laquelle il veut se lancer (décision 1, direction M), mais ensuite, les décisions 2,3,4 lui sont imposées par le marché (direction R,S,T). L'entrepreneur pourrait cependant décider indépendamment d'une innovation (ligne A sur la fig. 1) ; les lignes B,C,D représentent les initiatives étouffées par le marché.

L'innovation présentait dans cette théorie un grand intérêt parce que c'était la seule décision qui permettait à l'entreprise d'agir sur les contraintes du marché, de la placer temporairement, lorsque l'innovation réussissait, dans une position de monopole et de lui donner une certaine liberté de choix en la dégageant momentanément du contrôle sévère du marché. Cependant, les innovations réussies étaient rares, ce qui explique que les économistes se préoccupaient essentiellement du mécanisme du marché". [1]

§ 2. La théorie néoclassique [2]

Elle se caractérise par la négation ou l'adaptation des hypothèses classiques :

- maximisation du profit : objectif unique et finalité de l'entreprise dans la théorie classique, la théorie néoclassique lui donne une définition tellement large que toutes les décisions peuvent être justifiées du point de vue de la maximisation du profit. Le développement de l'analyse marginale donne du profit une définition abstraite mais précise : pour Alfred Marshall, le profit provenait du déséquilibre du marché et devait disparaître quand les forces de concurrence se seraient équilibrées. C'est donc PARCE QUE le marché et la concurrence sont imparfaits que le profit, basé sur l'idée d'un avantage relatif, peut être maximisé.

- concurrence imparfaite : cette hypothèse n'entraîne pas seulement l'existence du profit ; elle fait également disparaître le profit comme objectif unique. Voyons quels pourraient être certains de ces objectifs :

a. En concurrence parfaite, les décisions d'ordre commercial se réduisaient au simple choix de vendre ou de ne pas vendre au prix du marché ; ce prix était établi indépendamment de l'entreprise. La différenciation des produits, l'accroissement du nombre des concurrents, la disparition de la notion de prix "donné" augmentent la possibilité de choix de l'entrepreneur : la situation relative des concurrents résulte de manoeuvres stratégiques.

---

[1] BOULDING K.E. et SPIVEY W.A. : op. cit. p. 210-213

[2] D'après Boulding, elle prend naissance chez Cournot qui, en 1838, écrivit ses "Recherches sur les principes mathématiques de la Théorie des Richesses", mais elle n'a vraiment reçu droit de cité qu'à partir de ses développements qu'on trouve chez Jevons, Edgeworth, Marshall, Harrod, Joan, Robinson et Chamberlin.

b. La réputation de l'entreprise : la concurrence étant imparfaite, les décisions de l'entrepreneur ne résultent plus des "images" que se font les différents groupes qui gravitent autour de l'entreprise (clients, fournisseurs, actionnaires, pouvoirs publics ..), mais de ce que les diregeants PENSENT de ces images. De passive, la réaction des dirigeants devient active: par des actions judicieuses, ils vont essayer de connaître ces "images", déterminer celle qu'ils voudraient voir adopter, décider enfin des moyens à mettre en oeuvre pour obtenir ce changement.

c. Recherche du pouvoir : Jean Marchal a démontré que l'on pouvait inverser le sens de la causalité qui existe entre le profit et la puissance. Pour lui, s'il est vrai que, dans un système capitaliste, le profit est une source de puissance, il n'en est pas moins vrai que c'est l'exercice du pouvoir de l'entreprise sur son milieu qui est à l'origine du profit. C'est le pouvoir effectif d'une entreprise qui lui permet de modifier l'équilibre préexistant. Selon la théorie néoclassique les mouvements de déséquilibre des marchés sont à l'origine des profits temporaires et font l'objet de pressions conscientes de la part des entreprises de façon à obtenir un avantage relatif et un certain monopole.

d. Si tous les économistes sont d'accord pour affirmer que la production doit être évaluée d'après son "efficacité", leurs opinions divergent quand il s'agit de savoir ce que l'on entend par "efficacité" : on parle d'efficacité technique, économique, technologique, sociale etc... D'après Papandréou [1], "l'efficacité est une notion rationnelle, car elle conduit à maximiser un indice d'utilité. Elle implique soit la maximisation de certains objectifs par l'emploi de moyens donnés, soit la minimisation moyens mis en oeuvre pour arriver à un but donné ... Une entreprise peut être efficace sanspour cela rechercher le profit maximal". On est loin du profit maximal, objectif unique de la théorie classique.

---

[1] PAPANDREOU A.G. : "Problems in the Theory of the Firm ; a Survey of Contemporary Economics", vol. II, p. 134, Richard D. Irwin, Inc., Homewood, Illinois 1952.

Des hypothèses implicites de la théorie classique, la nouvelle théorie conserve les hypothèses de motivation, d'information et d'organisation, mais elle en modifie la nature. A l'hypothèse de stationnarité, elle substitue une hypothèse de croissance et elle remplace l'hypothèse d'indépendance par une hypothèse d'influence. Pour Miller [1] les hypothèses qu'elle adopte implicitement sont les suivantes :

1. l'hypothèse de motivation selon laquelle le but de l'entreprise est d'arriver à des résultats satisfaisants ;
2. l'hypothèse d'information selon laquelle les informations parviennent de façon confuse, déformées et avec beaucoup de "bruit". Dans ces conditions, l'acquisition et la diffusion de l'information constituent des problèmes que l'entreprise doit résoudre au sein de son organisation ;
3. l'hypothèse d'organisation selon laquelle le processus des décisions est déterminé par la structure de l'entreprise, structure qui conditionne également la diffusion de l'information ;
4. l'hypothèse de croissance selon laquelle les besoins, les ressources, l'état de la technologie et des connaissances sont susceptibles de changement ;
5. l'hypothèse d'influence selon laquelle les différents paramètres économiques ne sont pas indépendants les uns des autres mais peuvent être modifiés sous l'influence de l'action des entreprises.

Ces hypothèses diffèrent des hypothèses classiques, mais conservent des points communs avec elles : il existe dans les deux théories des régulateurs permettant d'assurer l'équilibre de l'entreprise. Dans la théorie classique, une modification des prix des facteurs entraînait automatiquement des substitutions ; c'est le seul ajustement que tolère le contrôle extrême exercé par le marché. Dans

---

[1] MILLER J.P. : op. cit. p. 135 - 136

la théorie néoclassique, les systèmes de régulation sont internes à l'entreprise : ils feraient partie de sa structure, de sorte qu'une modification des prix des facteurs de production entraîne des décisions internes qui rétablissent l'équilibre. L'entreprise ne recherche plus un maximum, mais un résultat satisfaisant, une maximisation de sa satisfaction. Cette notion correspond à l'idée qu'il existe une latitude entre une situation de survie et une situation maximale : dans cette marge, il appartient à l'entrepreneur de décider.

La nouvelle situation est représentée par les figures 3 et 4 :

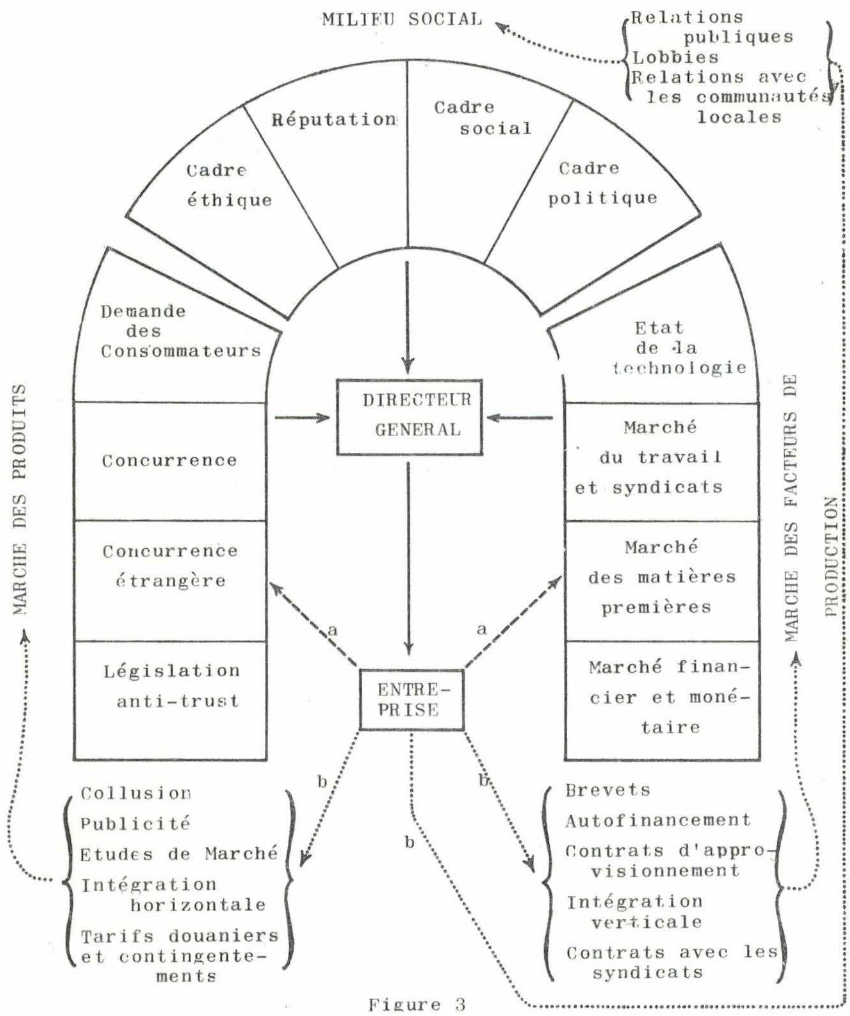


Figure 3

a. Les lignes discontinues montrent que le marché exerce un certain contrôle, mais que l'entreprise conserve une certaine liberté d'action.
b. Les lignes en pointillé montrent quelles sont les activités dans lesquelles l'entreprise s'engage à la suite de décisions internes. Ces activités ont pour but de réduire l'incertitude à laquelle l'entreprise est soumise.

Dans la figure 3, le dirigeant bénéficie d'une grande latitude de choix dans ses décisions, latitude que lui donne son pouvoir partiel de monopole. Cependant, une plus grande incertitude pèse sur ses décisions : d'autres entreprises de la même industrie ou de la même région peuvent en effet disposer d'un certain pouvoir de monopole. Il peut par exemple y avoir non pas un salaire et un prix; il peut aussi ne pas y avoir une seule technologie, mais plusieurs. Le dirigeant prendra ses décisions de façon à minimiser les incertitudes les plus importantes, car l'entreprise opère dans un milieu qui n'est plus contrôlé ni protégé par le marché. La latitude de choix rend les questions d'organisation et d'information très importantes.

Décisions	Origine
1. Que produire ?	1. Directeur Général
2. Comments produire ?	2. Directeur de la Production
3. Combien produire ?	3. Directeur des Ventes
4. A quel prix ?	4. Directeur Général Adjoint

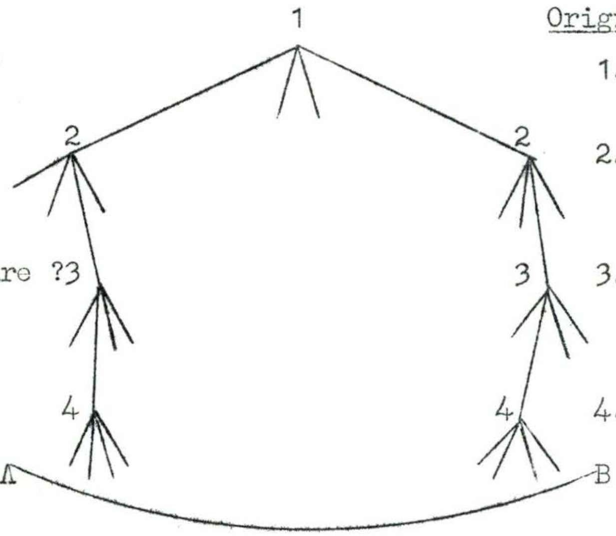


Figure 4

Dans la figure 4, les décisions à prendre sont identiques à celles de la théorie classique, mais ce ne sont plus les mêmes personnes qui les prennent : ce ne sont plus les forces extérieures du marché, mais les membres du groupe de direction. Toutes les décisions sont au-dessus de la ligne AB. :

"On peut imaginer que cette ligne est la partie inférieure d'un store vénitien. Selon la théorie classique, l'économiste qui connaît le marché peut prévoir les décisions : sur la figure 2, le store est levé et l'on voit à travers la vitre. Sur la figure 4, le store est baissé ; la connaissance du marché ne suffit plus à l'économiste pour comprendre et prévoir les décisions. Il lui faut maintenant connaître également ce qui se passe derrière le store, c'est-à-dire la structure interne de l'entreprise ainsi que les différents processus dont elle est le siège" [1]. Le marché continue à exercer certaines contraintes, mais ne contrôle pas d'une façon stricte.

§ 3. Conclusions

Dans la première section, nous disions que l'entreprise n'était pas un navire se balançant au gré des vagues. Le vocable "utilisation judicieuse" est suffisamment général pour englober les deux théories, mais remarquons qu'il évoque :

- soit un problème d'affectation : comment combiner les différentes ressources de manière optimale ? C'est à cette question que tentait de répondre la théorie classique, dont le concept-clé résidait dans la fonction de production.
- soit un probléme d'organisation que l'on pourrait aussi qualifier de problème d'information ou de décision. Nous n'avons plus affaire à une entreprise passive essayant tant bien que mal de se créer une petite place dans la société et se caractérisant par sa soumission au milieu ; nous avons affaire à une entreprise tentant de transformer la société dans laquelle elle s'installe et d'agir sur elle. Elle est une force nouvelle qui s'impose et remodèle le milieu. L'essentiel dans une théorie de l'entreprise est l'origine du contrôle qui s'exerce sur elle. Ce contrôle est plus souvent interne qu'externe. Le marché, qui constitue une toile de fond, établit des limites et des contraintes. Il existe néanmoins une gamme de décisions sur laquelle le marché a peu d'influence. De là découlent les rôles essentiels tenus :

  a. par l'information à propos duquel la théorie néoclassique a mis l'accent sur le mouvement de "feedback" existant entre l'information, la décision et sa réalisation,

---

[1] BOULDING K.E. et SPIVEY W.A. : op. cit. p. 213.

b. par l'organisation interne (et non plus l'organisation du marché) : l'entreprise est l'oeuvre d'un groupe et non plus d'un seul homme,

c. par le processus d'élaboration des décisions qui se définit par la succession d'un certain nombre de démarches destinées à identifier le problème et à lui trouver une solution optimale.

C'est surtout sur les deux derniers points que nous allons insister dans le chapitre suivant, mais n'oublions pas que leur étude n'a aucun sens sans informations.

Chapitre 2 :

L'étude de l'entreprise

Section 1 : Organisation et structure d'entreprise

"L'entreprise est un ensemble d'éléments ; encore faut-il que cet ensemble soit organisé ; un ensemble est organisé lorque ses éléments sont choisis, agencés et coordonnés en vue d'atteindre un but ou de remplir une fonction. Mais l'essentiel semble bien être non pas la description de cet état, mais l'ACTION visant à mettre l'ensemble des éléments à même de remplir leurs fonctions, à coordonner et à contrôler l'ensemble de celles-ci... L'organisation est donc fondamentalement un processus d'analyses et de méthodes orientées vers l'action et insérées dans le temps" [1].

Les problèmes posés par l'étude de l'organisation sont de deux ordres : économiques et techniques d'une part, psychologiques d'autre part. Dans ce cadre, six types de doivent être assumées [2] :

- connaissance des faits : prévisions et diagnostic
- choix des buts : politiques (orientations qualitatives) et objectifs (buts chiffrés et datés)
- organisation des moyens : programmes et budgets
- définition de la structure des responsabilités d'exécution : communication des objectifs et des programmes ; délégation et contrôle correspondant ; coordination des équipes et groupes de travail
- conduite des hommes : choix, perfectionnement, motivation
- contrôle : mesure ou évaluation de l'exécution, actions correctives.

La ou les politiques de l'entreprise sont "l'ensemble des principes généraux et particuliers sur lesquels on s'appuie et des méthodes que l'on

---

[1] COLLARD R. : "Organisation de l'entreprise", Namur 1967.

[2] GELINIER O. : "Fonctions et tâches de la Direction générale" édition "Hommes et Techniques", Paris 1963

utilise pour atteindre les objectifs que l'on a préalablement définis et acceptés" [1].
Les plans concrets de réalisation permettant d'atteindre ces objectifs s'appellent "programmes".

Les objectifs et politiques sont mis en oeuvre par des hommes et des services qui remplissent des fonctions dont l' agencement constitue la structure. Elle ne peut être imposée à priori de l'extérieur car elle dépend notamment du type d'entreprise auquel on à affaire. En combinant trois facteurs (la dimension de l'entreprise, le type d'industrie et le type de production) on aboutit à un nombre infini de types d'entreprises et, par le fait même, de structures (Eilon [2] en dénombre huit). Cette structure, qui définit les services, délimite leur pouvoir de décision et établit les relations entre eux, lie en un tout cohérent les départements essentiels et interdépendants d'une entreprise : départements commerciaux, légaux, financiers, du personnel, de recherche et de production.

Section 2 : Les décisions relatives à la constitution et au fonctionnement d'une entreprise

On peut décomposer l'ensemble des décisions à prendre lors de la constitution et de la réalisation d'une entreprise en un certain nombre d'étapes. Cette classification insiste surtout sur la période de temps que ces décisions embrassent. Il ne faudrait pourtant pas tracer des frontières bien définies entre les trois éléments que nous allons évoquer, d'une part parce qu'il n'y a aucune discontinuité entre elles, d'autre part parce que les décisions portant sur un élément de l'ensemble ont des effets sur tous les autres éléments de cet ensemble. Pour ces raisons, nous n'entrerons pas dans les problèmes de définition et de délimitation qui, à notre avis, sont par nature insolubles. Notons simplement que les conséquences de ces décisions seront d'autant plus importantes que le terme envisagé est long et que les objectifs seront d'autant plus généraux et moins sujets à un traitement analytique que l'horizon

[1] COLLARD R. : op. cit.

[2] EILON S. : "Elements of Production Planning and Control", MacMillan Company, New York 1962 p 11 - 14

temporel est éloigné. Toutes ces décisions s'inscrivent concrètement dans des "programmes" :

- les décisions relatives à l'INVESTISSEMENT relèvent du long et du très long terme. A la base, quelques idées dont certaines sont retenues pour être étudiées de façon plus approfondie. Ces décisions doivent répondre aux questions suivantes : Pourquoi créer une nouvelle entreprise ? Dans quel secteur doit-elle se situer ? Quelle sera sa place dans la structure économique nationale et internationale ? Quelles sont ses perspectives d'avenir ? Quelles sont les possibilités techniques d'exploitation ? Comme un produit techniquement valable n'en est pas pour autant économiquement rentable, une étude de marché, une analyse des coûts et de rentabilité doit être entreprise. Si toutes ces questions ont reçu des réponses satisfaisantes ET si ces réponses sont compatibles avec les objectifs et/ou les contraintes sociales et politiques, on prend la décision d'investir (en supposant que les moyens financiers nécessaires pour réaliser cet investissement ont été trouvés).

- les décisions relatives au "PREPLANNING" relèvent du moyen et du court terme (un à dix ans) ; elles sont prises en fonction des décisions prises au stade antérieur et doivent aboutir à des objectifs plus précis. Auparavant, on a grosso modo répondu à la question : que va-t-on faire ? ou, que peut-on faire ? Il s'agit maintenant de savoir : comment théoriquement réaliser la décision d'investir ? Les décisions prises à ce second stade sont très importantes, car elles doivent tenir compte de toutes les contraintes extérieures ainsi que de leur compatibilité : les exigences du marché, qu'il soit réel ou potentiel, qu'il s'agisse du marché des produits ou des facteurs ; les exigences techniques : matériel, transport, terrains, bâtiments, le "plant layout" (disposition à l'intérieur de l'entreprise des différents équipements).

  A la fin de cette étape, une structure grossière de l'entreprise est construite en ce sens qu'il est possible de la décomposer en départements (financiers, commerciaux, du personnel, de recherche et de production), de situer leur place dans l'entreprise et de spécifier les liaisons qui devront exister entre eux.

- les décisions relatives au PLANNING relèvent du court et du très court terme. C'est en fonction de la production immédiate, des commandes à satisfaire, de la conjoncture économique que ces décisions doivent être prises. Dans les étapes précédentes, les décisions étaient sujettes à des modifications, à des corrections adaptives, mais non à des changements radicaux, ce qui est le cas du planning : le dynamisme d'une entreprise s'estime, entre autres, par sa rapidité d'adaptation et de conversion interne, dues à de nouveaux procédés de fabrication, à des améliorations techniques ou à des changements de gamme de production. Dans cette étape, il s'agit d'étudier comment mettre pratiquement la décision d'investir à exécution.

  C'est ce que nous allons examiner dans le chapitre suivant.

Chapitre 3 :

Le département "Planning et contrôle de production"

Pour être parfaitement logique, nous aurions dû appeler ce département "département de production" et le subdiviser en deux parties : planning et contrôle.
La première partie aurait pour fonction de préparer le travail à effectuer, la seconde de le réaliser et de le contrôler. Cette subdivision aurait mieux mis le doigt, d'une part sur l'aspect "décision" (planning), d'autre part sur l'aspect "réalisation" (contrôle). Cette distinction se heurte cependant à des difficultés pratiques, non pas parce qu'il est difficile de savoir si une chose fait partie du "planning" ou du "contrôle", mais parce que chacun de ces deux services n'a de sens qu'en fonction de l'autre : la réalisation suit le planning, mais les indications données par le contrôle sont envoyées au planning pour y être corrigées et adaptées ; nous aurons l'occasion de revenir sur ce mouvement de "feedback". Nous utiliserons donc l'expression "département

de planning et de contrôle de production".
Après avoir défini ces deux termes dans une première section, nous expliciterons les fonctions de ce département dans une seconde section, en distinguant le planning du contrôle. Lorsque, du niveau théorique, nous aurons passé au niveau pratique (troisième section), nous déduirons une structure de ce département où les deux éléments seront imbriqués l'un dans l'autre.

Section 1 : Définition du planning et du contrôle

"The highest efficiency in production is obtained by manufacturing the required quantity of product, of the required quality at the required time, by the best and cheapest method"[1]. Pour réaliser cet objectif, on emploie le planning et le contrôle, outils coordonnateurs de toutes les activités productrices.

Le planning, dans une perspective à court et moyen terme, établit sur base de toutes les données disponibles un schéma de production, compte tenu des contraintes technologiques, commerciales et économique.

Le pendant indispensable du planning est le contrôle. Ce terme a, dans notre vocabulaire, une consonnance péjorative de vérification, de critique et d'inspection. Dans le vocabulaire anglais par contre, le mot "control" exprime une notion beaucoup plus active, désignant le fait qu'une activité n'échappe pas à la direction qu'on lui assigne. Pourquoi un contrôle ? Les processus qui font l'entreprise sont déterminés par un grand nombre de facteurs dont les uns peuvent être manipulés par les responsables à titre d'action corrective de contrôle, les autres variant hors de tout contrôle ; ces perturbations "incontrôlables à priori" nécessitent l'existence d'un mécanisme qui collecte les informations relatives à la réalisation du processus, détecte les perturbations et enclenche un système d'ajustement et de modifica-

[1] ALFORD L.P. et BANGS J.R.: "Production Handbook", Ronald Press Co 1952, cité par EILON S., op. cit. p. 1

tion du planning. Ainsi s'établit un mouvement de "feedback" entre le contrôle et le planning.

Section 2 : Les fonctions du département "planning et contrôle de production

Se limiter à quelques considérations sur les fonctions de ce département est une gageure. Suivant le secteur, le type de produits, la grandeur de l'entreprise, son importance économique et les exigences externes et internes, chaque entreprise a sa propre conception des fonctions et de la structure de ce département. Nous voudrions donner une classification et une structure très générale pour éviter de tomber sur des cas particuliers.

Si le planning et le contrôle de production ont trait à la "direction and coordination of the firm's material and physical facilities toward the attainment of prespecified goals, in the most efficient available way" [1], les fonctions qui en découlent peuvent être divisées en neuf catégories dont les six premières relèvent du planning et les trois dernières du contrôle :

1. Approvisionnement : cette fonction consiste à mettre à la disposition de la production, des matières premières et des produits semi-finis. Ces matières doivent être disponibles, en quantité et qualité déterminées, au moment et à l'emplacement voulus, au coût le plus bas compatible avec la qualité technique requise ; à ces problèmes s'ajoutent ceux des sources d'approvisionnement, des délais de livraison, de la standardisation et de la réduction des variétés des matières, de leur inspection, sans compter les problèmes issus de la sous-traitance. Cette fonction s'occupe donc essentiellement de la gestion des stocks des matières nécessaires à la production, des sources d'approvisionnement et du transport. Remarquons que, pour son bon fonctionnement, une liaison étroite doit s'établir entre ce service et les autres départements, commerciaux et financiers en particulier.

---

[1] EILON S. op. cit. p. 2-4

2. Méthodes de production : l'objectif de cette fonction est d'analyser les différentes méthodes de production possibles et d'en retenir une qui soit compatible avec les circonstances et les moyens de production. Déterminer si une production sera continue ou par lots est du ressort de cette fonction. Elle doit aussi déterminer les différentes combinaisons de ressources permettant de réaliser le produit. L'objectif poursuivi est la mise au point des conditions et des méthodes permettant de fabriquer un produit de la qualité requise, au coût minimum, de le vendre au prix du marché et d'en retirer un profit raisonnable. A ce stade, il s'agit encore de méthodes générales, qui tiennent plus compte des circonstances et de l'équipement que des commandes particulières.

3. Equipement et main-d'oeuvre : les méthodes de production sont analysées en fonction d'un équipement et d'une main-d'oeuvre donnés. Ceux-ci sont étudiés d'un point de vue technique : disponibilité, productivité, capacité, (les normes "standards" ne sont pas étudiées à ce niveau-ci), manière de les utiliser. Pour l'équipement, il faut encore citer la politique de remplacement et d'entretien. Pour la main-d'oeuvre, il faut tenir compte de sa mobilité interne, de ses qualifications et de la politique de perfectionnement. Les caractéristiques techniques, l'emploi, le rangement, l'assignation du petit outillage relèvent également de ce service.

4. "Routing" : lorsque les méthodes générales de production sont définies, chaque étape de production est décomposée en opérations élémentaires et chacune d'elles est étudiée techniquement dans ses moindres détails ; c'est à partir de ce moment que les commandes particulières pourront être planifiées. Le routing, en effet, impose la manière dont la production doit s'écouler dans les centres de production ; cette étude se fait en fonction de la disposition des lieux, des possibilités de stockage intermédiaire et de l'utilisation de l'outillage. Elle aboutit à la création de fiches de travail ou de feuilles de route répartissant le travail à effectuer.

5. L'estimation : lorsque les commandes de production et les feuilles de route sont disponibles, ainsi que les spécifications concernant les vitesses, l'utilisation et la capacité de l'équipement, l'étape suivante consiste à assigner à chaque opération une durée standard. L'élément humain y est de loin le plus important, par l'imbrication d'éléments sociaux (travail en équipe), divergence de caractère (influant sur la rapidité, la souplesse, la qualité du travail), éléments psychologiques, cadre de travail etc... Tout cela fait partie d'une "étude des mouvements". On aboutit à assigner à chaque opération élémentaire une durée "normale".

6. L'ordonnancement : les machines doivent être chargées suivant leur capacité et leur habileté à réaliser une tâche déterminée. Pour assurer des courbes de charge régulières, les responsables de l'équipement, de la main-d'oeuvre, du routing et des estimations, doivent assurer l'utilisation optimale de l'équipement et de la main-d'oeuvre et, par voie de conséquence, l'efficacité du processus de production. Ils doivent veiller à l'arrivée, en temps voulu, des différentes composantes du produit à assembler et à la présence des matières nécessaires. Ils doivent tenir compte des durées de préparation-machine nécessaires lors du changement, sur un même équipement, de produits à traiter ; ils doivent assigner chaque opération à une machine particulière et, lorsque plusieurs machines sont susceptibles de traiter cette opération répartir leur traitement dans le temps ; ils doivent enfin s'assurer que les opérations composant un travail commencent et s'achèvent en temps voulu.

7. Fabrication : jusqu'à présent, rien n'a été réellement produit : les fonctions précédentes (de planning) organisaient, planifiaient les travaux à exécuter. Cette fonction-ci, par contre, concerne l'exécution des fonctions de planning et consiste en la "routine of setting productive activities in motion through release of orders and instructions and in accordance with previously planned times and sequences

as embodied in route sheets and loading schedules"[1]. Elle autorise le début des opérations de production et suit le déroulement du processus en s'assurant que le mouvement matériel s'enchaîne suivant les indications données par l'ordonnancement. En un mot, elle effectue le contrôle, entraînant un mouvement de "feedback" vers la fonction d'ordonnancement.

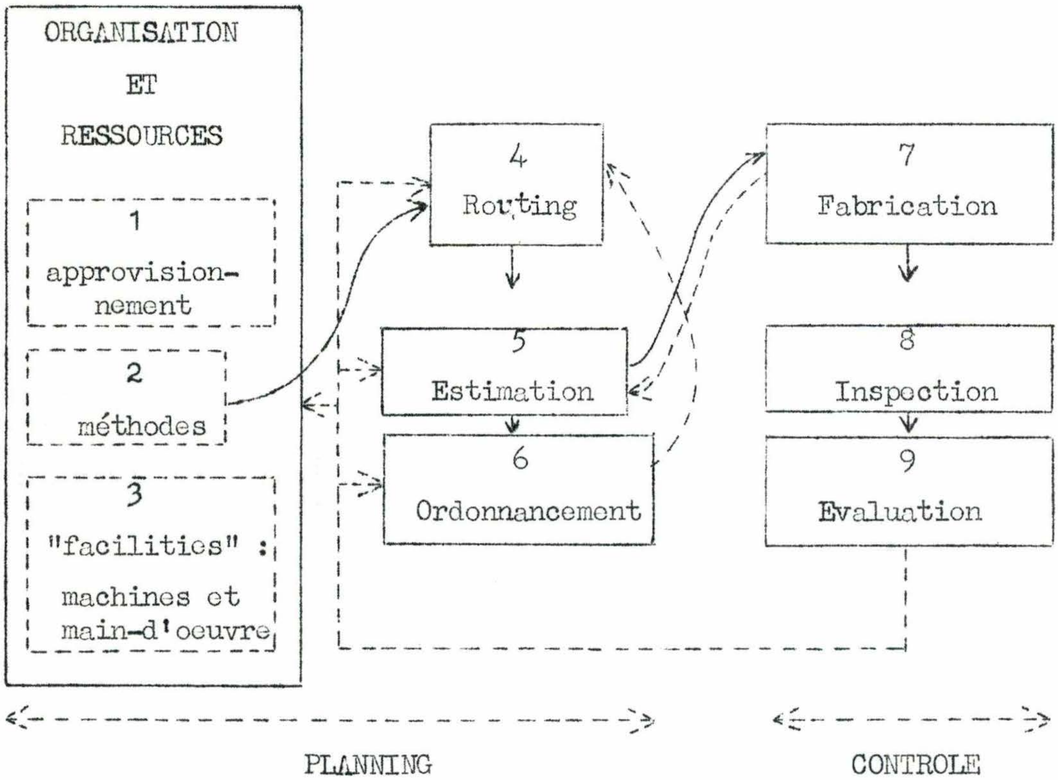
8. Inspection : bien que le contrôle de qualité soit souvent considéré comme une fonction qui n'incombe pas au département "planning et contrôle de production", nous l'incluons dans celui-ci, parce que ses remarques et ses critiques sont essentielles tant pour la production en cours que pour le planning de la production ultérieure. A ce niveau également, s'instaure un mouvement de feedback vers la fonction "estimation et méthodes de production" en vue d'améliorer les méthodes et d'indiquer leurs implications économiques en fonction de différents niveaux de qualité.

9. Evaluation : cette fonction forme le lien entre le contrôle et le planning ultérieur ; souvent négligée, elle est pourtant d'une importance capitale, car, en collectant toutes les données et les informations recueillies tout au long du processus, elle les analyse et en tire des conclusions qui seront renvoyées aux différents stades du planning.
Comme la fonction "fabrication", elle assure un mouvement de feedback, mais, contrairement à cette fonction, des liaisons s'établissent non seulement avec la fonction "ordonnancement", mais aussi avec toutes les autres fonctions. Cela provient de ce que ses préoccupations concernent tant la production des en-cours que l'amélioration des méthodes : dans une optique à long terme, on voudrait que les expériences passées puisent être évaluées pour améliorer le processus productif.

---

[1] ALFORD L.P. et BANGS J.R., cité par EILON S., op. cit. p.3.

Une fois ces différentes fonctions définies, les relations entre elles se présentent de la façon suivante :



———————— : succession des responsabilités fonctionnelles

- - - - - - - - - : mouvement de feedback

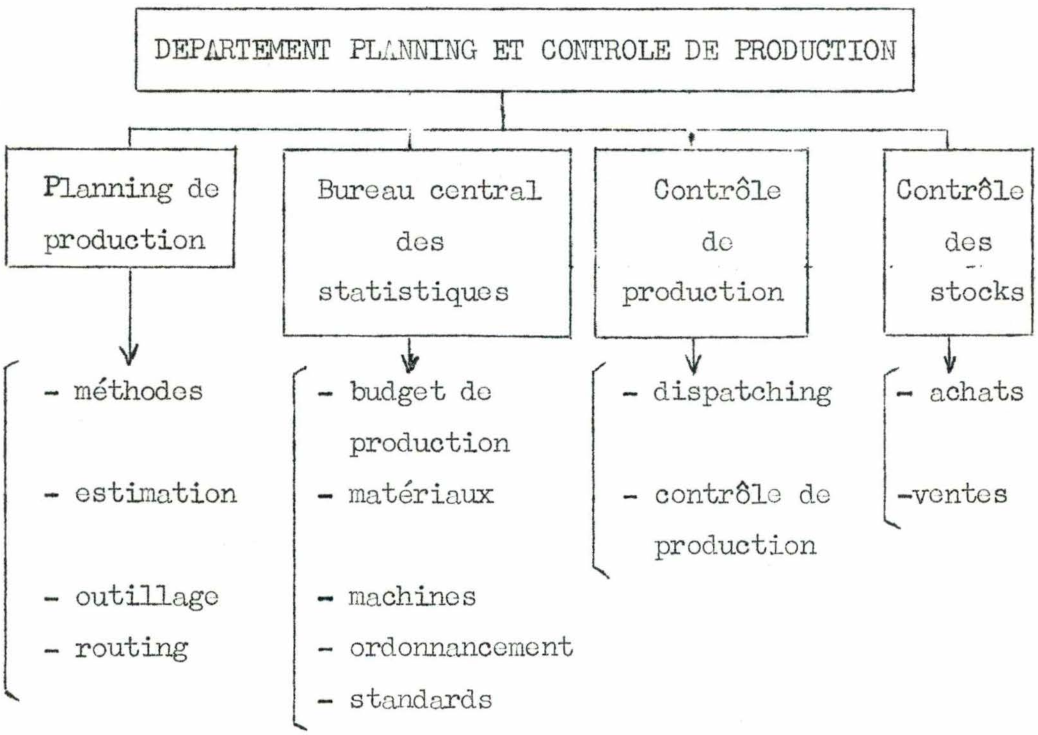
Nous terminerons cette section en insistant sur deux points :

- le feedback, résultat de l'installation d'un mécanisme de contrôle, établit un lien constant entre le planning de production et sa réalisation ;
- comme nous le verrons encore dans la section suivante, il ne faut pas faire de ce département un "monde à part" ; il ne faut pas oublier qu'il est en relation étroite avec tous les autres départements de l'entreprise.

Section 3 : Structure du département "planning et contrôle de production"

Les fonctions du département de production peuvent être structurées de la manière suivante ; les variantes sont nombreuses, mais celle que nous présentons ici offre l'avantage de centraliser en un seul service toutes les données relatives aux ressources, capacités et ordonnancement : à un stade ultérieur, ce bureau de "statistiques" pourrait être remplacé par un ordinateur, organe collecteur d'informations et diffuseur d'instructions à tous les autres services du département.

Ce département se subdivise en quatre sections, chacune comprenant plusieurs services [1].



---

[1] Cette structure est en grande partie semblable à celle établie par EILON S. op. cit. p. 53. Les modifications que nous y avons apportées concernent la répartition des activités à l'intérieur des sections. Un service non décrit remplit la fonction décrite dans la section précédente.

1. Section planning de production :

- service des méthodes ;
- service de l'estimation des temps : étude des temps productifs et improductifs, des temps accordés aux opérateurs pour la réalisation de tâches qui ne nécessitent pas l'emploi de machines, des marges accordées pour des raisons de délais, de fatigue... ;
- outillage : service responsable de l'outillage nécessaire, de ses caractéristiques, de sas affectations, de son emplacement... ;
- service du routing : en relation constante avec le service responsable de l'ordonnancement, il doit traduire en termes simples les plans de production établis dans le service des méthodes. Il décompose le travail à effectuer en opérations, établit les contraintes de sucession entre elles etc... En fonction de ces données et de ces contraintes, le service de l'ordonnancement détermine l'ordre d'exécution des travaux et leur répartition dansle temps en respectant certains critères (délais de livraison, utilisation de l'équipement...)

2. Bureau central des statistiques :

- service du budget de production : les exigences budgétaires y sont examinées en fonction de l'exécution des commandes qui y sont consignées dans les carnets de commande ;
- service des matériaux : les informations concernant les matériaux disponibles y sont centralisées et leur allocation aux centres de production, décidée ;
- service des machines : collecte les données relatives au nombre de machines, à leur vitesse, fréquence de panne, rendement, productivité, mesures de protection... ;
- service de l'ordonnancement ;
- service des standards.

3. Section "contrôle de production" :

- service du dispatching : il est responsable d'un système de communication efficace entre le service d'ordonnancement et les centres de production ; il établit le lien entre l'aspect théorique et l'aspect pratique de la production ;

- service du contrôle de qualité : le service précédemment cité s'occupait des en-cours de fabrication et en assurait le contrôle; celui-ci, par contre, exerce son contrôle sur les produits finis.

4. Section "contrôle des stocks" :

Elle regroupe tous les services s'intéressant à la gestion des stocks tant à l'entrée qu'à la sortie.

Section 4 : Nécessité d'une structure

Cette structure a pour objectif essentiel de répartir le travail et correspond à la m éthodologie qui sera employée pour résoudre les différents types de problèmes qui se posent à la production ; cette méthodologie comprend trois étapes :

- collecte et classification des données ;

- résolution des problèmes, ce qui nécessite des méthodes de prévision du fait de l'existence de variables incontrôlables encore appelées "états de nature" ;

- prise de décision : on attend de l'étape précédente qu'elle choisisse une stratégie optimale, ou du moins proche de l'optimum, parmi toutes les possibilités qui peuvent se présenter. A supposer même que toutes les variables significatives soient inclues dans l'analyse, l'application automatique de ces stratégies optimales peut ne pas être souhaitée ni souhaitable ; d'où la nécessité de cette troisième étape. Même si l'objectif final de la firme était analytiquement bien posé, la programmation dynamique nous apprend que, pour atteindre son optimum, il n'est pas nécessaire que les sous-politiques soient optimales :

Une solution peut être optimale (sous-optimale) du point de vue de la valeur de l'ordonnancement, mais sous-optimale (optimale) ou incompatible si on remplace l'ordonnancement dans son contexte. Il est donc toujours très important de définir PAR RAPPORT A QUOI un optimum est calculé. De plus, il restera toujours des facteurs d'indétermination insoupçonnés et des éléments humains qui empêcheront l'application "straight forward" des méthodes ; cela rend l'intervention humaine dans la prise de décision indispensable. Le voeu pieux de certains qui voient dans un avenir plus ou moins lointain la possibilité d'une entreprise entièrement automatisée, rendant la présence humaine inutile, nous semble un leurre. En effet, les systèmes économiques diffèrent fondamentalement des systèmes déterministes où les liens de cause à effet sont clairement établis ; cela pour deux raisons qui ont trait à leur nature :

- ils dépendent de facteurs dont le degré de varabilité est tel que "any attempt to predict their behavior in deterministic terms has no justifiable basis" [1];
- ils sont structurellement complexes : les variables sont trop nombreuses et trop imbriquées pour pouvoir leur imputer des effets certains sur la conduite du système.

Nous aurons encore l'occasion de revenir sur ces problèmes, notamment dans la discussion concernant l'approche heuristique.

Maintenant que nous avons situé la place de l'ordonnancement dans l'entreprise et donné quelques indications sur son importance, nous entamons son étude de façon plus approfondie.

---

[1] EILON S. : op. cit. p. 143.

Chapitre 4 :

L'ordonnancement

La meilleure présentation générale du problème d'ordonnancement nous semble avoir été faite par B. Roy et ses collaborateurs [1]. Elle est générale par le fait qu'elle n'exclut aucun des problèmes qui se présentent sous ce vocable et qui en possédent les propriétés intrinsèques ; selon son schéma, nous examinerons ce qui leur est commun (section 1) et ce qui les différencie.(section 2).

Section 1 : Elements communs aux problèmes d'ordonnancement

Les problèmes d'ordonnancement sont tous les problèmes qui possèdent les trois catactéristiques suivantes :

a. Il s'agit d'étudier comment réaliser quelque chose :

ce "quelque chose" peut consister dans la construction d'un grand ensemble (navire, usine, habitation), un planning d'atelier, un emploi du temps (par exemple, dans le domaine scolaire ou administratif : organisation des circuits), l'organisation d'une banque ou d' un supermarché, la planification des travaux à traiter par un ordinateur etc ...

b. Ce "quelque chose" est décomposable en tâches : ces tâches sont soit des opérations élémentaires soit des groupes d'opérations (selon le détail du routing). Cette décomposition de tâches en opérations peut être difficile car elle exige des connaissances technologiques et une certaine expérience (nous supposons ce problème résolu), mais, dans tous les cas, ces tâches doivent être définies par rapport à un certain niveau d'homogénéité. Chaque tâche se différencie des autres par :

- le type d'opération à réaliser,

[1] ROY B. : "Les problèmes d'ordonnancement, application et méthodes", Dunod 1964.

- sa durée,
- son époque initiale et son époque terminale,
- chaque tâche suppose qu'on lui affecte certains moyens : matériels, humains ...

Certaines de ces données sont inconnues ou connues selon certaines lois de probabilité.

c. Cette réalisation est soumise à un ensemble de contraintes : l'existence de ces contraintes et d'une fonction-critère à optimiser fait de l'ordonnancement un problème économique. Le mot "contrainte" désigne la formulation analytique d'exigences qu'imposent :

  - la technologie : celle-ci impose une certaine structuration des tâches (une tâche ne peut commencer que lorsque certaines autres sont terminées) ;
  - la main-d'oeuvre : les effectifs sont toujours en nombre limité ;
  - le matériel : une machine ne peut traiter qu'une seule tâche à la fois ;
  - les fournisseurs et les clients (ou, de façon plus générale, l'environnement économique) : les contraintes principales sont les délais de livraison des matières premières ou des produits semi-finis et les exigences de la clientèle en ce qui concerne la date d'achèvement de leurs commandes ;
  - autres éléments : clients, disposition du matériel à l'intérieur des centres de production, structuration des centres d'autorité et de responsabilité, politique de l'entreprise en ce qui concerne les méthodes de contrôle ...

Le problème d'ordonnancement se pose alors de la manière suivante : en respectant le système de contraintes envisagé, il faut choisir pour chaque tâche les moyens qui doivent lui être affectés, l'époque à laquelle elle devra débuter, sa durée d'exécution (dans la mesure où ces éléments sont des inconnues du problème) et tout cela par rapport à une fonction-critère à optimiser ; il peut consister dans la minimisation de la durée totale de traitement, du coût total, du nombre des en-cours, de la longueur des files d'attente, de la maximisation de la production, de l'utilisation optimale des machines, de l'équilibre des courbes de charge etc... Si le ou les critères retenus sont susceptibles de recevoir une formulation analytique, on est conduit à rechercher un optimum, mais lorsque l'expression de ce critère est trop complexe, on se heurte à un dilemme :

- l'expression analytique doit être assez simple pour permettre de mener les calculs à bonne fin, mais, par le fait même, elle perd de sa souplesse et de son réalisme ;
- l'expression analytique, pour traduire toutes les subtilités du jugement humain est si complexe qu'elle en devient inutilisable.

Dans ces conditions, il vaut mieux abandonner l'idée d'optimisation sur le plan mathématique pour adopter la démarche qui consiste à explorer rapidement, méthodiquement et automatiquement l'ensemble des ordonnancements admissibles, en retenir quelques-uns sur base de critères simples, les critiquer et reprendre l'exploration en vue de leur en substituer d'autres qui ne résistent plus aux mêmes critiques.

Section 2 : Ce qui différencie les problèmes d'ordonnancement

Les différents types d'ordonnancement correspondent essentiellement à la nature des contraintes significatives ; les principaux types de contraintes sont au nombre de trois : notons, avant de les énumérer, que ces trois types se présentent souvent conjointement, mais, pour des raisons d'analyse, on ignore celles qui ne sont pas trop contraignantes.

a. Les contraintes temporelles : il s'agit des contraintes de localisation temporelle et de succession logique ; elle sont groupées par Roy dans l'expression "contraintes de potentiel". Les premières imposent à une tâche d'être fixée dans le temps (on situe la date de début ou sa date d'achèvement) ; elles trouvent leur origine dans des raisons climatiques (la construction d'immeubles par exemple), dans des délais de livraison accordés ou imposés par les fournisseurs ou dans des exigences commerciales. Les secondes contraintes limitent, essentiellement pour des raisons technologiques, l'intervalle de temps qui s'écoule entre deux tâches. Elles exigent, par exemple, qu'une tâche ne soit pas commencée avant qu'une autre ne soit à un stade suffisant d'avancement.

b. Les contraintes disjonctives : elles généralisent les contraintes de succession logique et traduisent la disjonction des intervalles de temps affectés à la durée d'exécution des tâches ; que ces intervalles de temps n'aient pas de durée commune apparaît lorsque, réclamant l'utilisation d'un matériel unique, celui-ci ne peut traiter qu'une seule tâche à la fois (la distinction entre ce type de contraintes et les précédentes apparaîtra plus clairement dans la section suivante).

c. Les contraintes cumulatives : il faut souvent associer à une tâche une fonction traitant les besoins en ressources dont elle nécessite l'emploi (principalement la main-d'oeuvre). A la différence des deux premiers types de contraintes qui sont mutuellement exclusives (disons plutôt que les secondes généralisent les premières), le troisième type se superpose en général aux contraintes de type temporel ; elles peuvent, en effet, intervenir de deux manières, selon le centre de préoccupation : lorsque la préoccupation essentielle réside dans l'équilibre des besoins de main-d'oeuvre, les contraintes cumulatives prédominent et sont les seules envisagées ; beaucoup plus fréquemment, elles se trouvent combinées avec les premières parce que l'équilibre des courbes de charge doit être établi en fonction d'une planification efficace des opérations [1].

Section 3 : L'ordonnancement unitaire et non unitaire

Nous avons dit dans la section précédante que les deux premiers types de contraintes étaient mutuellement exclusifs : il ne faudrait pas prendre ce qualifi-

---

[1] L'association des contraintes cumulatives et disjonctives n'est pas envisagée parce que la fonction des besoins en ressources est associée à des tâches et non à des machines.

dans un sens trop strict. Les contraintes disjonctives groupent l'ensemble des contraintes temporelles et celles relatives à l'équipement. Les contraintes temporelles constituent un sous-ensemble des contraintes disjonctives. S'il est vrai que l'étude des problèmes soumis à ces dernières résoudraient automatiquement les problèmes à contraintes temporelles, on a l'habitude de les distinguer, parce que les centres de préoccupation diffèrent : les problèmes soumis à des contraintes temporelles sont étudiés par l'ordonnancement unitaire et ceux soumis à des contraintes disjonctives par l'ordonnancement non unitaire.

§1. L'ordonnancement unitaire

Dans un projet unitaire, toutes les tâches forment un ensemble : dans ce contexte, on peut insérer les tâches relatives à la contruction d'un immeuble ou d'un navire ou les tâches à réaliser entre le moment de la "conception" d' une entreprise jusqu'à son fonctionnement (voir supra chap. 2). Contrairement aux problèmes associés à des contraintes disjonctives, la réalisation du "quelque chose" est suffisamment longue pour que l'on envisage qu'un projet à la fois.

Malgré la différence de dénomination donnée aux méthodes de réalisation de ce problème, dénomination provenant du travail original pour lequel la méthode a été développée [1], le principe de résolution est identique et repose sur la recherche du chemin critique. On établit un graphe potentiel-étape [2]: les sommets représentent les étapes, c'est-à-dire, le point de rencontre entre deux ou plusieurs arcs, qui eux représentent les différentes tâches ; leur enchaînement résulte des contraintes de succession logique ; on associe à chaque arc la durée (ou le coût) de la tâche correspondante. Le graphe complété, on procède à la recherche du chemin critique : après avoir daté la première étape,

---

[1] P.E.R.T. : Program Evaluat on Research Task et ses variantes : le PERT cost où on associe le contrôle budgétaire, le PERT charge où on associe les contraintes cumulatives ; C.P.M. : Critical Path Method ; P.R.I.S.M. : Program Reability Information System for Management ; P.E.P. : Program Evaluation Procedure.

[2] Roy préfère établir un graphe potentiel-tâche.

ce qui fixe le début possible des tâches issues de ce sommet, on procède à la datation de toutes les autres étapes en respectant les durées des tâches rencontrées sur les différents chemins possibles ; si un choix se présente, on prend la valeur maximum. Le calendrier ainsi formé s'appelle le "minorant". Si on opère de la même manière en partant du dernier sommet (auquel on donne la date à laquelle on est arrivé lors de la construction du minorant) et si on remonte le graphe vers le premier sommet, la date associée à ce premier sommet doit être identique à celle établie au départ du minorant. Ce second calendrier s'appelle le "majorant". Les deux valeurs attribuées à chaque sommet sont les dates au plus tôt et au plus tard auxquelles les tâches qui ont ce sommet pour origine doivent commencer.

Deux cas peuvent se présenter :

- les valeurs associées à un même sommet sont identiques : cela signifie que, sous peine de retarder la date d'achèvement du projet, les tâches issues de ce sommet doivent débuter au plus tard à la date indiquée ; elles ne peuvent pas non plus commencer plus tôt à cause des contraintes de succession ; aucune tolérance n'est donc permise. L'ensemble des arcs reliant des sommets dont les dates sont identiques forme le ou les chemins critiques [1]. Les tâches qui appartiennent à ce chemin sont donc particulièrement à surveiller, car l'ensemble du projet sera retardé d'autant d'unités de temps que la somme des unités de temps de retard de ces tâches. Inutile d'ajouter que cette planification des tâches est optimale : par construction du graphe, le projet ne pourra être terminé avant la date associée au dernier sommet.

- les valeurs associées à un même sommet ne sont pas identiques : cela signifie que ces tâches ne sont pas critiques, qu'une marge est tolérée ; le montant de cette marge équivaut à la différence entre les deux dates et il s'ensuit qu'une tâche issue de ces sommets peut commencer dans les limites de ces deux valeurs sans retarder l'achèvement total du projet.

---

[1] Par construction, les valeurs associées au premier et au dernier sommet sont identiques : ils constituent donc le début et la fin du chemin critique.

§2. L'ordonnancement non unitaire

Nous ne ferons ici que situer le problème. L'analyse détaillée des éléments qui le composent sera exposée dans la seconde partie.

L'ordonnancement non unitaire se réfère surtout aux problèmes de production industrielle [1]. La plupart des processus implique plusieurs transformations de matières ou de produits semi-finis en produits finis. Les différentes opérations doivent être traitées par un certain nombre de "machines"[2]. La décomposition des "supports"[3] en opérations élémentaires, l'assignation de ces opérations à des machines spécifiques, le calcul de la durée de traitement associée à chaque opération (qui dépend aussi de la machine utilisée) et la détermination des stades de fabrication (les contraintes de succession logique) ont déjà dû être établies antérieurement. Le problème d'ordonnancement consiste à déterminer l'ordre, la séquence des différents supports sur chaque machine de façon à optimiser une fonction-critère prédéterminée.

Si on ne considérait que les contraintes par lesquelles les opérations doivent être exécutées dans un certain ordre, rien ne différencierait ce problème de l'ordonnancement unitaire. L'indétermination de l'ordre de passage des commandes sur les machines (sources de disjonction) qui doivent traiter certaines de leurs composantes engendre des degrés de liberté supplémentaires. Cet ordre doit précisément être choisi de manière à satisfaire au mieux les priorités ou les délais imposés, à minimiser le coût et les emplacements des stocks intermédiaires ou la durée totale d'un ensemble d'opérations. Ce problème peut se compliquer par des possibilités supplémentaires de choix :

---

[1] Insistons sur le mot "surtout", car la réalisation du "quelque chose" ne limite pas l'ordonnancement au cadre de l'entreprise de production (cfr. section 1)

[2] A défaut de terme plus adéquat, nous avons traduit le mot anglo-saxon "facility" par "machine". Dans notre terminologie, "machine" désignera tout moyen par lequel quelque chose peut être fait ; il peut s'agir d'une machine, d'un homme, d'une équipe de travail ou d'un outil.

[3] Un support désigne ce qui doit être traité par la "machine". Dans notre optique de production, nous utiliserons le mot "commande".

plusieurs machines capables de réaliser le même travail, travaux d'assemblage etc ... (voir seconde partie).

Dans le cas unitaire, il n'y a qu'un projet à réaliser et chaque tâche est unique ; on suppose que tous les moyens de production lui sont automatiquement accordés ; ici, par contre, plusieurs commandes sont à réaliser en même temps et l'existence d'un nombre limité de machines engendre des conflits d'occupation (on supposera toujours qu'une même machine ne peut traiter qu'une seule commande à la fois). En un mot, ce qui le distingue de l'ordonnancement unitaire c'est que la localisation temporelle n'est pas exactement définie. IL en va de même devant la caisse d'un supermarché ou dans un poste à péage sur une autoroute lorsqu'une file de voitures est en attente, c'est-à-dire en "conflit" pour se voir accorder le même service par un préposé qui ne peut s'occuper que d'un seul client à la fois.(organisation des aérodromes, des guichets de banque, des tables de restaurant). Ces exemples veulent uniquement souligner que l'ordonnancement ne se borne pas uniquement à un problème de planning industriel, bien qu'ils rentrent tous dans la catégorie "problèmes d'entreprise" puisque celle-ci a été définie comme "unité de production de biens ou de services".

Chapitre 5 :

L'importance économique du problème d'ordonnancement

Section 1 : Généralités

Les trois premiers chapitres ont déjà fourni quelques indications sur cette importance, mais nous voudrions, dans ce chapitre, en expliquer brièvement les raisons. Celle qui recouvre toutes les autres provient de ce que l'entreprise forme un tout et qu'il est, par conséquent, illusoire de vouloir dissocier un de ses éléments des autres : les actions et les décisions concernant un des éléments vont avoir des conséquences sur les autres, mais ces mêmes autres éléments ne vont pas se borner à les subir : ils réagissent, ils rétroagissent par un mouvement qui leur est propre sur les éléments qui sont en partie cause de leur action. Ce mouvement action-réaction est qualifié dans la littérature de mouvement de feed-back. Prenons l'exemple simple de l'individu : il subit un grand nombre d'influences qui ne font qu'orienter son action dans une certaine direction, sans que l'on puisse dire qu'elles sont les causes de son action. Son action va, à son tour, réagir sur l'environnement.

L'ordonnancement est un problème d'organisation dans le sens donné à ce terme par la théorie néoclassique (voir chap. 1). C'est PARCE QUE l'entreprise a une action sur le marché et que la concurrence est imparfaite qu'un problème d'ordonnancement existe. On peut aller jusqu'à dire que, dans une économis dirigée, il n'existe pas (disons plutôt qu'il ne serait pas apparu comme problème), parce que l'entreprise y est créée en fonction de ce qu'elle doit faire. Si une force extérieure, quelle qu'elle soit, lui impose une production d'autant, de tel produit, au prix X, à réaliser par Y machines, à une époque T, il n'y aurait plus qu'un problème global d'entreprise, à résoudre enune seule fois et une fois pour toutes. L'ordonnancement n'est pas un problème d'affectation basé sur une fonction technique ou/et économique de production, mais un problème interne d'organisation de la production.

L'ordonnancement, sans signification s'il est étudié par et pour lui-même, vit dans un milieu de relations complexes d'interdépendances. L'objet de ce chapitre est d'énumérer quelques unes de ces liaisons, mais, avant cela, nous voudrions revenir sur une conséquence de ce qui précède : il est faux de croire que l'obtention d'une solution optimale pour chacun des éléments d'un ensemble entraîne nécessairement l'optimisation de l'objectif final de cet ensemble : les problèmes à résoudre varient d'un élément à l'autre, ce qui, inéluctablement, entraîne un manque d'homogénéité dans les jugements de valeur portés par les responsables sur le choix des critères de décision et, par conséquent, des difficultés d'expression analytique (lorsque celle-ci n'est pas rendue impossible par la nature même du problème). Ces défauts d'homogénéité aboutissent souvent à des constatations d'incompatibilité (voir infra).

Section 2 : Liaisons entre l'ordonnancement et d'autres problèmes se posant à l'entreprise

Nous n'étudierons dans ce qui suit que les économies indirectes, c'est-à-dire celles réalisées par un bon ordonnancement et affectant d'autres éléments que l'ordonnancement lui-même. Si nous n'envisageons les économies directes que dans la seconde partie, c'est pour ne pas trop nous écarter de l'objet de la première partie qui consiste à situer l'ordonnancement dans l'ensemble de l'entreprise.

§1. Liaisons avec les problèmes d'équipement

a. L'ordonnancement et l'investissement en équipement : cette relation, ainsi que la liaison entre l'ordonnancement et les stocks nous paraît la plus importante. Dans le chapitre 4, il apparut que la difficulté essentielle de l'ordonnancement (et, en même temps, ce qui le différenciait de l'ordonnancement unitaire) provient de la limitation des équipements [1]. Au point de départ, un investissement en équipement a été consenti et c'dst en fonction de cet équipement que l'ordonnancement des supports est résolu. Mais si, jusqu'à présent, la politique d'investissement (achat et remplacement) apparaît comme déterminée indépendamment de l'ordonnancement, la résolution de ce problème va permettre aux responsables de l'investissement de revoir leur politique : l'analyse de l'ordonnancement va donner des indications utiles sur les goulots d'étranglement existant à certains stades de production; ces goulots peuvent résulter soit d'un nombre insuffisant de machines ou de leur mauvaises performances techniques. Il existe aussi des méthodes d'ordonnancement qui peuvent orienter le choix d'un nouvel équipement (voir annexe I, section 1).

b. L'ordonnancement et l'équipement existant :

---

[1] A la limite (c'est-à-dire grâce à un équipement illimité), chaque support peut être considéré à lui seul comme un projet de type PERT, puisque les seules contraintes qui subsistent sont déterminées par la technologie, enserrant les durées de traitement des opérations dans des limites temporelles rigides (contraintes de potentiel)

Seront considérées comme données, pour les responsables de l'ordonnancement, les décisions relatives à l'utilisation de l'équipement :

- la politique d'entretien : va-t-on suivre l'adage "mieux vaut prévenir que guérir" ou va-t-on se contenter de réparer les pannes lorsqu'elles surviennent ?
- l'estimation des durées de préparation des machines lors du changement de la gamme de production. Ces données sont des contraintes pour l'ordonnancement et la prise en considération des durées de préparation pose un dilemme particulièrement aigu :
- ou bien on minimise cette durée, ce qui accroît le coefficient d'utilisation productive des machines, mais réduit l'étendue de la gamme de production ou entraîne des coûts de retard importants ;
- ou bien, pour échapper aux inconvénients précédents, on procède au "changeover" fréquent ; le coefficient d'utilisation productive et la production en subiront les conséquences.

Un équilibre est donc à trouver entre l'étendue de la gamme, les quantités à produire et les coefficients d'utilisation. Nous reviendrons plus en détail sur ce problème dans la seconde partie.

c. L'ordonnancement et le "plant layout" : avant de régler le problème d'ordonnancement, on a dû disposer les équipements dans les centres de production en tenant compte des moyens de transport, des facilités de manutention, des emplacements de stockage intermédiaires, du type de produit à fabriquer, de la mobilité du personnel ... Ici encore des incompatibilités avec la recherche d'un ordonnancement optimal peuvent survenir.

§2. Liaisons avec les marchés

a. L'ordonnancement et le marché des outputs : les décisions à prendre dans ce cadre sont :

- l'acceptation de la commande : décision prise en fonction des délais exigés, de l'importance du client, du carnet de commande et du niveau de service ;

- l'adaptation de l'appareil de production aux spécifications du client : les quantités à produire de chaque partie de la commande, la date de mise en fabrication de chacune de ces parties et la détermination des dates de livraison ;
- la détermination de la suite des opérations, pour vérifier les dates formulées au stade précédent.

Toutes ces décisions relèvent de l'ordonnancement. Selon le ou les critères employés, les caractéristiques spécifiques des commandes, la capacité et la disponibilité des machines, la résolution de l'ordonnancement permettra de respecter les délais de livraison (et, par conséquent, minimiser les coûts de retard) ou/et d'accepter toutes les commandes qui se présentent. Mais ce serait obliger l'ordonnancement à rester passif. De façon plus active, il peut aboutir à un raccourcissement des délais de livraison ou à la recherche de nouveaux débouchés : un accroissement de l'activité est possible par une meilleure utilisation du temps-machine sans qu'il soit nécessaire de réaliser de nouveaux investissements en équipement.

b. L'ordonnancement et le marché des inputs : le choix des fournisseurs, les quantités demandées, les spécifications de qualité ne relèvent pas directement de l'ordonnancement, mais auront une influence sur lui.

Deux remarques :

- C'est ici qu'intervient le problème des stocks. Il y a trois types de stocks : les stocks des matières premières, des produits finis ou semi-finis et celui des en-cours. Les deux premiers sont gérés par le service des stocks et sont en liaison étroite avec l'ordonnancement ; c'est pour cette raison qu'ils sont étudiés ici. Les stocks des en-cours, par contre, sont gérés par les responsables de l'ordonnancement : une bonne organisation de ces stocks à l'intérieur des ateliers peut être un objectif (ou une contrainte) pour les responsables de l'ordonnancement. C'est la raison pour laquelle nous en dirons un mot dans la seconde partie.

- Nous avons jusqu'à présent parlé presque essentiellement de la production sur commandes. N'oublions pas qu'il existe un type de production sur stocks par l'intermédiaire d'un département de vente ou commercial. Nous verrons dans la seconde partie que cette distinction permet, d'une certaine façon, de classifier les problèmes d'ordonnancement en problèmes statiques et en problèmes dynamiques ; comme nous nous attacherons aux problèmes statiques, nous parlerons surtout de la production sur commandes.

Coincés par la double préoccupation de réduire les coûts de commande et les coûts de stockage, une quantité optimale peut être déterminée, mais, en ne voyant que cela, on néglige les besoins réels de la production, les exigences des clients et les délais accordés aux fournisseurs ; d'où la nécessité d'un examen de la demande, des dates et des volumes de réapprovivionnement, des coûts de commande, de détention, de dépréciation et de rupture ; en fonction de ces facteurs et grace à un contact permanent avec les centres de production, une politique de gestion des stocks est établie. Mais des modifications, si minimes qu'elles soient, dans les quantités produites, dans le routing et dans l'ordonnancement de la production entraînent la revision constante des valeurs données aux paramètres et donc de la politique de gestion des stocks.

§3. Autres liaisons

Au lieu d'exposer toutes les liaisons existant entre l'ordonnancement et tous les autres aspects de l'entreprise, citons simplement les liens avec la main-d'oeuvre (disponibilité, efficacité, productivité et qualification), avec la politique des transports (externes ou internes à l'entreprise), avec la gamme de production (routings, étendue de la gamme, quantité à produire et qualité à satisfaire). Le meilleur ordonnancement peut s'avérer incompatible avec le degré de qualité requis, parce qu'une "bonne" utilisation du temps-machine peut entraîner un vieillissement précoce de l'équipement, des irrégularités de production ou des déchets importants. [1]

Les quelques relations citées se bornaient à des liaisons biunivoques ; ce cadre de pensée est trop étroit. Pour l'exposé de chacune d'elles, nous avons dû faire appel à des éléments extérieurs. Les sous-ensembles concernant des liaisons d'interdépendance entre 3, 4, ... éléments ne nous feront pas sortir de l'impasse ; à la limite, nous aurions abouti à une étude de toutes les relations existant dans cette entité appelée "entreprise" pour cerner l'importance de l'ordonnancement.

CONCLUSIONS DE LA PREMIERE PARTIE

I. Dans la littérature anglo-saxonne, on rencontre les termes "scheduling" et "sequencing" (parfois "dispatching", chez Gere notamment) et, malheureusement, ils ne sont pas toujours employés dans le même sens :

- il y a les auteurs qui n'établissent pas de véritables distinctions :
  - Gere [2] avoue employer les trois expressions indistinctement ;
  - Conway [3] se base sur l'etymologie des termes pour les distinguer : "sequencing" = mise en ordre, en séquence, et "scheduling" = établissement d'un diagramme ; sur cette base, le mot "sequencing" concerne l'ordonnancement d'un nombre indéterminé de commandes sur une seule machine, tandis que "scheduling" celle d'un nombre indéterminé de commandes sur plusieurs machines.
- il y a les auteurs qui établissent une distinction réelle. Parmi ces au-

---

[1] Quelques liaisons sont étudiées dans l'Annexe I

[2] GERE W.S. : "Heuristics in Job Shop Scheduling" ; Management Science, vol. XIII n° 3, nov. 1966, p. 167-186.

[3] CONWAY R.W., MAXWELL W.L., MILLER L.W. : "Theory of Scheduling", Addison Wesley Company 1967.

teurs, on peut citer Eilon [1], Starr [2], Muth et Thompson [3] ; ils ont envisagé l'ordonnancement comme un problème à résoudre conjointement à d'autres. Ces problèmes connexes sont, principalement, ceux qui concernent les quantités à produire, les exigences du marché, les stocks et l'investissement. Tout ce qui concerne l'ensemble de ces questions est groupé sous le vocable "scheduling", alors que le mot "sequencing" reçoit la même signification que le mot "scheduling" chez Conway.

Il nous semble que la différence de ces deux approches est la même que celle existant entre le "théorique" et le "pratique". Le premier groupe suppose donné :

a. les quantités à produire sur chaque machine : les décisions concernant ce point ont dû être prises antérieurement en fonction du marché et des possibilités de stockage ;

b. les disponibilités en équipement : l'ordonnancemant est résolu indépendamment de la politique d'investissement de l'entreprise ;

c. l'assignation de chaque opération à une machine particulière.

En résumé, les deux questions relatives au "Que produire ?" et au "Comment produire ?" sont supposées avoir reçu une réponse satisfaisante. Le deuxième groupe, plus préoccupé d'organisation industrielle, englobe tous ces problèmes ; si, à priori, cette manière de voir l'ordonnancement apparaît plus attrayante parce que plus réaliste, n'oublions pas que ce qu'elle gagne en généralité et en réalisme, elle le perd en précision, particulièrement du côté de la formulation mathématique. Nous nous bornerons donc, dans la suite de cette étude, à considérer l'ordonnancement principalement sous son aspect théorique. Nous aurons au moins explicité dans le

---

[1] op. cit.

[2] STARR M.K. : Production Management : Systems ans Synthesis" Prentice-Hall, Inc; Englewood Cliffs N.J. 1964.

[3] MUTH J.F. et THOMPSON G.B. : "Industrial Scheduling", Prentice Hall, Inc. Englewood Cliffs N.J. 1963

dernier chapitre comment on pouvait envisager l'ordonnancement de façon plus pratique et, pour concrétiser ces liaisons, nous avons cru bon d'ajouter dans l'annexe I quelques exemples de ces liaisons...

II. Si on définit les problèmes économiques comme étant les problèmes traitant de l'affectation ou de l'utilisation judicieuse de ressources rares et si l' objet de la science économique consiste à étudier les principes de solution des problèmes économiques, (en d'autres termes, à étudier "comment tirer le meilleur parti de ressources limitées"), il ne fait aucun doute que l'ordonnancement est un problème économique :

- les ressources limitées : ce terme groupe toutes les contraintes, principalement celles qui ont trait aux disponibilités en équipement, aux ressources en matières premières et en main-d'oeuvre ;
- l'utilisation judicieuse : " l'utilisation" est limitée par les contraintes de routing ; l'adjectif "judicieux" insiste sur l' existence d'un critère de choix permettant de juger de la valeur de "l'utilisation" ainsi que celle des méthodes qui permettent d'aboutir à l'optimisation de ce critère.

III. L'entreprise formant un tout, sa structure doit être aussi perméable que possible : notre mentalité cartésienne nous pousse à classer, cataloguer, détacher les éléments pour les étudier par et pour eux-mêmes. Chaque élément est certainement indispensable mais non suffisant ; il s'ensuit que la qualité essentielle d'une structure réside dans les possibilités de contacts et d'ouverture qu'elle offre.

IV. Ce point nous entraîne à prévenir une critique possible. La présentation de cette première partie pourrait laisser croire que nous estimons le problème d'ordonnancement comme le problème essentiel à résoudre dans une entreprise et que sa résolution facilite celle des autres problèmes. A cause du sujet de cette étude nous avons bien dû donner à l'ordonnancement un "éclairage" particulier. Il en va de même dans une salle de spectacle lorsque les faisceaux lumineux sont dirigés sur ce qui, à un moment donné, apparaît comme le plus important.

V. La transposition à la réalité des résultats analytiques théoriques de l'ordonnancement est extrêmement complexe. Si la formulation mathématique du critère est déjà difficile (voir ci-dessus), il en est de même des contraintes : il faut d'abord recueillir toutes les contraintes significatives ce qui, dans un monde dominé par l'incertitude de "l'action" de certaines variables ,, peut friser le "chimérique". Même recueillies, ces contraintes doivent être formulées, tâche rendue complexe par l'intervention de facteurs humains et de jugements de valeurs impondérables. Le nombre des contraintes, la multiplicité des sources dont elles émanent et le risque d'incohérence qui en découle nécessitent ensuite une analyse de compatibilité. Enfin, la réalité est adaptive, dynamique : certaines contraintes disparaissent, de nouvelles prennent leur place ; "l'effort continu à réaliser vaut-il encore la peine d'être fourni?".

VI. La mise en place d'un système d'hypothèses restrictives (infra) peut résoudre certains problèmes, mais fondamentalement : "Y a-t-il réellement un problème d'ordonnancement dans les entreprises?". Telle était la question que s'est posée W.F.Pounds dans un article intitulé "The Scheduling environnment" [1]. Cet homme se trouvait fort contrarié lors de visites rendues à des entreprises américaines lorsque, après avoir demandé comment ce problème était résolu, on lui répondait "Qu'est-ce que l'ordonnancement ?". Il exposa les principes généraux que nous avons discutés dans le chapitre 4, mais les responsables affirmaient ne pas connaître ce problème. Après un certain temps de réflexion, il trouva que "the job shop scheduling problem is not recognised by most factory schedulers because, FOR THEM, in most cases, no scheduling

---

[1] MUTH J.F. et THOMPSON G.B. : op. cit. chapitre 1 p. 5-12.

problem exists... because the organisation, which surrounds the schedulers reacts to protect them for strongly interdependent sequencing problems". Le problème d'ordonnancement semble donc exister, mais n'est pas toujours reconnu, parce qu'il est trop complexe à résoudre ; de ce fait, ils éliminent une série de contraintes, ce que l'auteur appelle "les réactions de l'organisation". En quoi peuvent consister ces réactions ? :

a. une entreprise enregistre des commandes par différents canaux : lettres, téléphone, telex, contact personnel... Comme une durée standard a été préalablement établie pour l'estimation des dates de livraison, on accepte les commandes si les exigences du client correspondent avec cette estimation ; par contre, si le client exige une livraison plus rapide, une discussion s'engage et un arrangement est trouvé ; dans le cas où le même client exige souvent un service rapide, on refuse sa commande.

b. l'entreprise a, en général, une estimation peu précise du coût d'un service rapide, car, dépendant des travaux en cours, ce coût n'est pas fixe. Une des composantes de ce coût réside dans le problème d'ordonnancement qui survient lors de la présence de commandes à priorités élevées.
Résultat : "The sales department therefore is protecting the scheduling function from a scheduling problem when the department begins to resist requests for fast service".

c. chaque jour, on prépare une feuille de route contenant les commandes pressantes et/ou en retard ; ce document circule entre les départements de ventes et de production et forme la base des décisions qui stabiliseront et simplifieront le problème d'ordonnancement. Si la liste est trop longue, on commence par réduire le nombre des commandes acceptées ; si cela ne suffit pas, le département de production autorise les heures supplémentaires et envisage la démultiplication des chaînes de production par l'achat de nouveaux équipements.

Cette méthode n'est pas critiquable à priori: même si ce système n'est pas un exemple d'optimum, l'entreprise peut maintenir un niveau de service satisfaisant. Conclusion de Pounds : "Computationally difficult problems do not arise because those constraints that would create them are removed when they become active".

Deux critiques :

- il est trop facile de dire : le problème est trop difficile, supprimons tout ce qui le complique : "on fait comme s'il n'existait pas". Cela va à l'encontre de la méthodologie scientifique où, devant un problème complexe, on commence par le simplifier en introduisant des hypothèses ; ensuite on tente de les lever.
- cette manière de voir l'ordonnancement le rend "passif" : dans sa phase ultime, la résolution de ce problème doit être considérée dans le cadre "dynamique" de l'entreprise ; il fixe lui-même des dates de livraison les plus rapprochées possible, tout en les respectant. C'est le côté actif de l'ordonnancement.

VII. Dans une étude scientifique, on cherche une solution optimale si elle existe, des solutions sous-optimales dans le cas contraire ; mais, souvent, on fait l'hypothèse "des choses égales par ailleurs" (ceteris paribus). N'oublions cependant pas que la recherche d'un optimum n'est pas un but en soi : les conséquences de cet optimum peuvent s'avérer incompatibles avec certains objectifs qui débordent du cadre particulier où le problème se situe.

"A l'heure actuelle, on ne peut guère demander à la R.O. que de réaliser des suboptimisations partielles. Une des raisons de cet état de fait est le caractère relativement récent de l'analyse des systèmes et l'ignorance de données qui seraient nécessaires pour la compréhension des systèmes complexes. Une autre raison est que l'étude des systèmes demande beaucoup de temps, alors que l'homme d'affaires désire, en général, des résultats rapides ...

...ce qui le conduit à financer plus volontiers les recherches qui lui semblent le plus rapidement rentables ... (la R.O. ne s'applique) aux organisations complexes que si l'on prend soin de vérifier la compatibilité des différents standards d'optimalité adoptés... Il faut cependant avoir présent à l'esprit que le choix aléatoire de critères de sub-optimisation ne conduit pas forcément à un optimum général ; le contraire est même l'éventualité la plus probable. L'optimisation globale d'un système ....dépend de l'optimisation simultanée et cumulative de nombreux sous-systèmes. En d'autres termes, l'optimisation globale d'un système complexe est équivalente à la réunion de sub-optimisations correctes, dans la mesure où les standards d'optimalité adoptés pour effectuer ces dernières tiennent bien compte du fait que le standard global est fonction de nombreuses variables" [1].

---

[1] BOULDING K.E. et SPIVEY W.A. : p. 180-191.

TITRE II :

L'APPROCHE ALGORITHMIQUE DES PROBLEMES D'ORDONNANCEMENT [1]

Introduction

Après avoir situé la place de l'ordonnancement non unitaire dans l'entreprise, le point de vue que nous allons aborder maintenant est complètement différent et fait essentiellement appel aux différentes techniques de recherche opérationnelle susceptibles de résoudre ce problème. Il ne faut pas perdre de vue que l'objectif de ce travail consiste à tenter de répondre à la question suivante : les méthodes de résolution appliquées à l'ordonnancement sont-elles suffisantes ? Aussi, un survol rapide de ces méthodes a dû être inséré et ce malgré son aridité. Nous avons cependant essayé de réduire au maximum les aspects mathématiques et les développements algorithmiques, pour lesquels on se référera aux annexes.

Dans cette seconde partie, chaque type de problème fait l'objet d'une description rapide, de l'examen des méthodes utilisées pour le résoudre de façon optimale lorsque ces méthodes existent, des cas particuliers pouvant se présenter et des critiques que l'on peut adresser à ces méthodes.

Nous voudrions qu'à la fin de la lecture de la seconde partie, les points suivants soient devenus évidents :

- le problème d'ordonnancement unitaire est complexe tant au niveau théorique qu'au niveau de son utilisation pratique ;
- hormis des cas bien particuliers, se présentant rarement dans la pratique, il n'a pas encore reçu de solution optimale ;
- même lorsqu'ils sont résolus, les méthodes de résolution sont souvent pratiquement inapplicables soit par leur complexité même, soit par les

---

[1] Pour la seconde partie, on se référera presque exclusivement à CONWAY, R.W., MAXWELL, W.L. et MILLER, L.W. op. cit. page 1 à 140.

hypothèses sous-jacentes aux modèles, soit par les moyens de calcul à mettre en oeuvre. Il existe donc un fossé assez profond entre la solution mathématique et la solution économique de ce problème. Probablement parce que les praticiens ne sont pas théoriciens et vice-versa.

- la seconde partie sert en fait d'introduction à la troisième partie en lui donnant sa raison d'être.

Chapitre 1 :

Le modèle général d'ordonnancement non unitaire

Dans ce chapitre, nous analyserons ce modèle en étudiant ses trois composantes :

- les variables et les contraintes du modèle : qu'est-ce qui fait de l'ordonnancement un problème spécifique ? Cette spécificité provient essentiellement de la nature des contraintes dont les liens avec les différentes variables seront systématisées en faisant appel à la théorie des graphes. L'ordonnancement apparaîtra alors comme un problème d'analyse combinatoire (section 1).

- les hypothèses de base du modèle : il s'avère impossible de résoudre ce problème si on ne fait appel à des hypothèses ; la plupart d'entre elles sont tellement restrictives qu'elles limitent l'application des résultats obtenus à des cas bien particuliers (section 2);

- la fonction-objectif du modèle : nous répondrons à la question de savoir sur quelle base nous pouvons juger de la valeur d'un ordonnancement (section 3) ;

Nous terminerons ce chapitre :

- en classant les modèles d'ordonnancement : cette classification sera faite sur base des trois composantes que nous venons d'énumérer (section 4) ;

- en introduisant les chapitres suivants par une <u>classification des méthodes de résolution de ces problèmes</u> (section 5).

<u>Section 1</u> : <u>La formulation du problème par appel à la théorie des graphes</u>

La théorie des graphes apparaît d'une grande utilité en matière d'ordonnancement :

- elle permet de <u>visualiser</u> les problèmes simples en synthétisant les données du problème : les tâches (opérations) sont représentées par des points, appelés "sommets" ; ces sommets sont reliés par des arcs symbolisant les contraintes qui lient les tâches (opérations) les unes aux autres ;
- elle permet de <u>résoudre</u> des problèmes simples : lorsque le nombre de sommets et/ou d'arcs n'est pas trop élevé et que la fonction à optimiser est fonction du temps, il s'avère plus facile de travailler graphiquement que par pur raisonnement mathématique.

Ces deux raisons apparaîtront plus clairement dans la suite - notamment en ce qui concerne l'aspect <u>pratique</u> de la théorie des graphes - lorsque nous parlerons du diagramme de Gantt.

Soit un centre de production qui se voit imposer le traitement de n tâches et dispose de m machines ; chaque tâche i se compose de g_i opérations :

- chacune d'elles s'est vue assigner une machine sur laquelle elle doit être traitée ;
- chacune d'elles occupe sur cette machine, un nombre d'unités de temps : la "durée de traitement" ;
- certaines d'entre elles sont reliées par des contraintes de succession logique.

Reprenons ces points en les explicitant sur base d'un exemple : un atelier se compose de 4 machines et doit réaliser 5 tâches ; chacune de ces tâches se décompose en 4 opérations (ce qui n'est pas une contrainte : plusieurs opérations d'une même tâche peuvent nécessiter le traitement d'une <u>même</u> machine ; le nombre d'opérations par tâche peut être inférieur à celui des machines).
Les caractéristiques des opérations sont les suivantes :

- l'assignation d'<u>une</u> opération j ($j = 1 \dots g_i$) de la tâche i ($i = 1 \dots n$) à la machine k($k = 1 \dots m$) la distingue des autres opérations et se note x_{ijk}. On suppose donc que <u>chaque</u> tâche et que <u>chaque</u> machine a reçu un numéro d'identification tout à fait arbitraire ; nous reviendrons sur la signification de la valeur donnée à l'indice "opération" (j).

- chaque x_{ijk} nécessite une durée de traitement p_{ijk} dont l'ensemble forme une matrice P, comprenant n lignes (tâches) et m colonnes (machines) (dans notre exemple 5 et 4).

$$
[P] = \begin{array}{c|cccc|}
 & \multicolumn{4}{c}{\text{MACHINES}} \\
\text{TACHES} & M1 & M2 & M3 & M4 \\
\hline
T1 : & 5 & 8 & 2 & 1 \\
T2 : & 4 & 1 & 4 & 8 \\
T3 : & 2 & 3 & 2 & 9 \\
T4 : & 6 & 4 & 3 & 4 \\
T5 : & 1 & 3 & 1 & 2 \\
\hline
\end{array}
$$

- dans les deux alinéas précédents, nous avons ignoré l'indice "opération" : cet indice renvoie aux contraintes de succession logique : on exige de certaines opérations d'une tâche qu'elles vérifient la relation :

pour tt.i: $t_{ijk} \geqslant t_{i(j-1)k°}$ où t_{ijk} représente la date à laquelle une opération j de la tâche i peut commencer sur une machine k. Elle ne pourra commencer qu'après la fin d'une autre opération (j - 1) de la même tâche i traitée sur la machine k°. La date à laquelle cette opération sera terminée est la somme de sa date initiale et de sa durée de traitement. En conclusion, l'indice j, contrairement aux indices i et k, indique l'ordre de traitement des opérations composant une tâche. Ce "routing" peut revêtir trois formes (le sigle $\ll$ signifie "précède immédiatement") :

- chaque tâche se présente comme une séquence linéaire d'opérations : un ordre strict est imposé entre toutes les opérations ;
  pour tout i :

  $$x_{i1k} \ll x_{i2k} \ll x_{i3k} \ll \dots \ll x_{ig_i k} \quad (k = 1 \dots m) ;$$

- aucun ordre n'est imposé : l'indice "opération" perd sa signification puisqu'il n'indique que l'ordre dans lequel les opérations doivent être réalisées ;

- certaines opérations d'une tâche sont liées par des contraintes de succession. Pour éviter toute ambiguïté, on est obligé de décomposer l'indice j en deux sous-indices : - un sous-indice z donne un numéro d'identification arbitraire à la chaîne formée par deux opérations au moins (z peut varier de 1 à $g_i - 1$),
  - un sous-indice j° indique l'ordre de l'opération dans la chaîne z.

  L'ensemble des opérations de la tâche i se compose alors de deux sous-ensembles : celui qui ne comporte que des opérations indépendantes et celui qui ne comprend que des chaînes. Pour tout i, on aura :

$$\begin{cases} x_{ik};\ x_{ik'};\ x_{ik''} \qquad (k = 1 \dots m) \\ x_{iz1k} \ll x_{iz2k}\ ;\ x_{iz'1k} \ll x_{iz'2k} \ll x_{iz'3k} \dots \\ \qquad (k = 1 \dots m \text{ et } z = 1 \dots (g_i - 1) \end{cases}$$

Cette notation, extrêmement lourde, est, en général, remplacée par une adaptation de la matrice P (P'). On traduit les contraintes de succession par des arcs horizontaux qui permettent l'élimination de l'indice j. (dans notre exemple, nous avons une séquence stricte pour chaque tâche).

M A C H I N E S

$[P']$ = TACHES	M1	M2	M3	M4
T1	5	8	2	1
T2	4	1	4	8
T3	2	3	2	9
T4	6	4	3	4
T5	1	3	1	2

Ainsi, la valeur 4 qui se trouve à l'intersection de la seconde ligne et de la troisième colonne est égale à p_{233}, puisqu'il s'agit de la durée de traitement de la tâche n° 2 de l'opération à traiter en troisième lieu (arcs horizontaux) sur la machine n° 3.

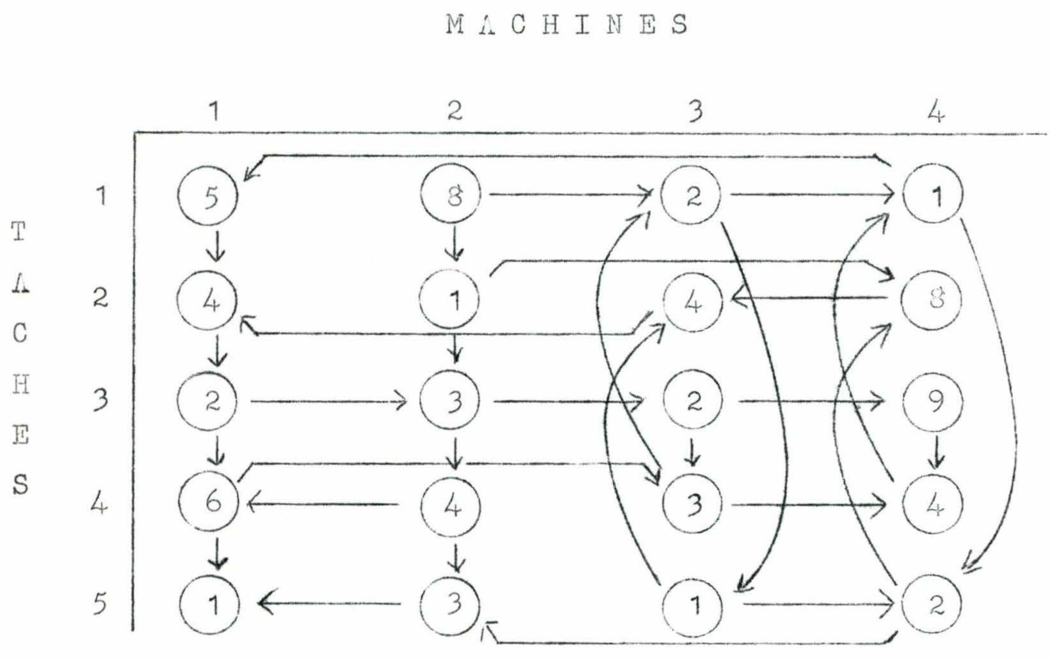
Trouver un ordonnancement consiste à déterminer une matrice de permutation Q (nxm) dont l'élément caractéristique q_{ik} indique la tâche à traiter en ième lieu sur la machine k ; auparavant, il s'agissait d'un problème de séquence des opérations d'une tâche (matrice P'), maintenant, il s'agit d'un problème de séquence des opérations à réaliser par une machine (matrice Q).

M A C H I N E S

$[Q]$ = ordre des tâches	M1	M2	M3	M4
1 :	1	1	3	3
2 :	2	2	4	4
3 :	3	3	1	1
4 :	4	4	5	5
5 :	5	5	2	2

où $q_{23}=4$ signifie que la tâche à réaliser en deuxième lieu par la machine 3 est la tâche 4 et que cette tâche suivra celle indiquée par q_{13} c'est-à-dire la deuxième. Cette matrice nous permet de tracer des arcs verticaux qui lient les opérations assignées à une même machine. On obtient de la sorte un graphe potentiel-tâche où :

- les <u>sommets</u> identifient les différentes opérations et leurs durées respectives de traitement ; après avoir éliminé l'indice j par l'introduction d'arcs horizontaux, la présentation matricielle du graphe permet la suppression des indices i et k ;
- les <u>arcs</u> reliant deux sommets se différencient selon qu'ils sont :
  . <u>horizontaux</u> : ils traduisent le transfert de la tâche réalisée sur une machine vers une autre machine (matrice [P']),
  . <u>verticaux</u> : ils traduisent la séquence de traitement des différentes tâches sur une même machine (matrice [Q])



Le fait de traiter d'un problème d'ordonnancement non unitaire se déduit du graphe :

- la présence d'arcs verticaux témoigne de l'existence de plusieurs tâches ;
- chaque sommet est l'extrémité initiale et terminale d'un seul arc vertical, manifestation des contraintes disjonctives par lesquelles une machine ne peut traiter qu'une seule opération à la fois.

La distinction entre les arcs verticaux et les arcs horizontaux mérite des explications supplémentaires :

- <u>les arcs verticaux</u> : à chaque machine est associé un graphe dont les arcs forment un "chemin élémentaire". Comme un seul arc vertical s'origine et aboutit à un sommet, nous avons affaire à des relations de précédence <u>immédiate</u> dont l'ensemble forme un ordre complet c'est-à-dire ne permettant pas d'alternatives.
  En termes d'ordonnancement, nous n'aurons toujours qu'un seul arc vertical entre deux sommets et tout sommet ne sera le point d'origine et d'aboutissement que d'un seul arc ; il ne peut y en avoir plus parce qu'une machine ne peut traiter qu'une opération à la fois et que ces opérations doivent se suivre dans le temps ; il ne peut y en avoir moins, car c'est l'objet même de l'ordonnancement de trouver l'ordre de séquence des opérations sur chaque machine.

- <u>les arcs horizontaux</u> : ils sont d'une tout autre nature que les arcs verticaux, car, contrairement à ceux-ci, ils n'expriment pas nécessairement des relations de précédence immédiate (notre exemple représente le cas où le routing des opérations est strict). Supposons le cas où la tâche envisagée réside dans le montage d'une voiture ; pour simplifier, ces opérations se réduisent à :

  - amener le bloc moteur, la carrosserie et les pare-chocs dans l'atelier (x1),
  - la finition et l'inspection des pare-chocs (x2), du bloc moteur (x3), de la carrosserie (x4),
  - la peinture (x5),
  - l'assemblage des trois pièces (x6).

Des contraintes technologiques exigent que :

<u>1</u> l'on amène les pièces avant de les inspecter,

<u>2</u> la peinture suive l'inspection des pare-chocs et de la carrosserie ; on permet cependant les alternatives suivantes :

  - les pare-chocs sont envoyés au département peinture où ils sont assemblés à la carrosserie ;

- les pare-chocs sont d'abord assemblés à la carrosserie et, ensuite, l'ensemble est amené au département peinture ;

3 l'assemblage termine l'ensemble du processus.

Ces contraintes peuvent être introduites dans une matrice booléenne (une matrice dont l'élément significatif est 1 ou 0). Si l'arc xixj existe, xij = 1, sinon xij = 0).

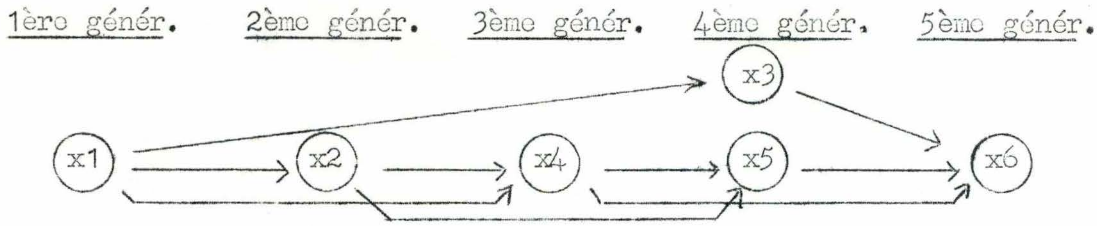
O P E R A T I O N S

O P E R A T I O N S		x1	x2	x3	x4	x5	x6
	x1	0	1	1	1	0	0
	x2	0	0	0	1	1	0
	x3	0	0	0	0	0	1
	x4	0	0	0	0	1	1
	x5	0	0	0	0	0	1
	x6	0	0	0	0	0	0

Pour ordonnancer les sommets, on les décompose en sous-ensembles disjoints de telle sorte que les sommets d'un sous-ensemble ne seront atteints que si les sommets des sous-ensembles antérieurs ont été réalisés ; un sous-ensemble répondant à ces conditions est appelé "génération". Ce concept de génération a pour but de suppléer au manque de précision résultant de la présence simultanée de relations de précédence simple et de relations de précédence immédiate entre les sommets (nous noterons que, dans le cas des arcs verticaux, chaque sommet constitue à lui-seul une génération), en établissant des relations de précédence "immédiate" entre générations. Les sommets qui font partie d'une même génération sont indépendants les uns des autres. L'algorithme de Mallebranche [1] permet cette décomposition en générations ; on aboutit, pour notre exemple, au graphe suivant :

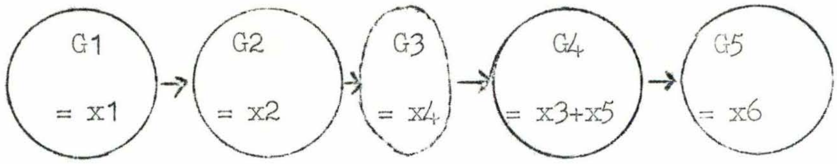
---

[1] KAUFMANN, A. et DESBAZEILLE, G. : "la méthode du chemin critique" Dunod 1966 p. 6 ss



Nous noterons que :

- le nombre d'opérations étant limité et les relations faciles à tracer, l'application de l'algorithme est superflue dans notre exemple, ce qui n'est pas le cas lorsque les opérations et les contraintes sont plus nombreuses ;
- les sommets de la 4ème génération ne sont pas reliés par des arcs, ce qui témoigne de l'indépendance des éléments faisant partie de la même génération ;
- l'établissement d'un graphe, construit logiquement et sans ambiguïté, donne une meilleure idée du genre de contraintes auxquelles on a affaire :
  - . les contraintes exprimées par les arcs x1-x4, x2-x5,x4-x6 sont redontantes parce que ces arcs joignent deux sommets reliés par une <u>suite</u> d'arcs représentant des relations de précédence <u>immédiate</u> : il s'agit respectivement des suites x1-x2-x4, x2-x4-x5 et x4-x5-x6. Si nous pouvons affirmer que ce sont les arcs x2-x4 et x4-x5 qui expriment les précédences immédiates et non pas x2-x5, c'est grâce à la définition de la génération (la génération 2 contenant x2 <u>doit</u>, par définition, précéder la génération 3 comprenant x4 qui elle-même doit précéder la génération 4 qui comporte x5) ;
  - . tous les autres arcs (c'est-à-dire ceux qui subsistent après l'élimination des arcs redontants) expriment des relations de précédence immédiate mais l'application de la définition d'une génération permet la suppression des arcs x1-x3 et x3-x6 (ou x5-x6) ;
  - . après ces manipulations, le graphe est réduit à sa plus simple expression, les sommets représentant des générations et les arcs des relations de précédence immédiate entre ces générations.



Remarque terminale :

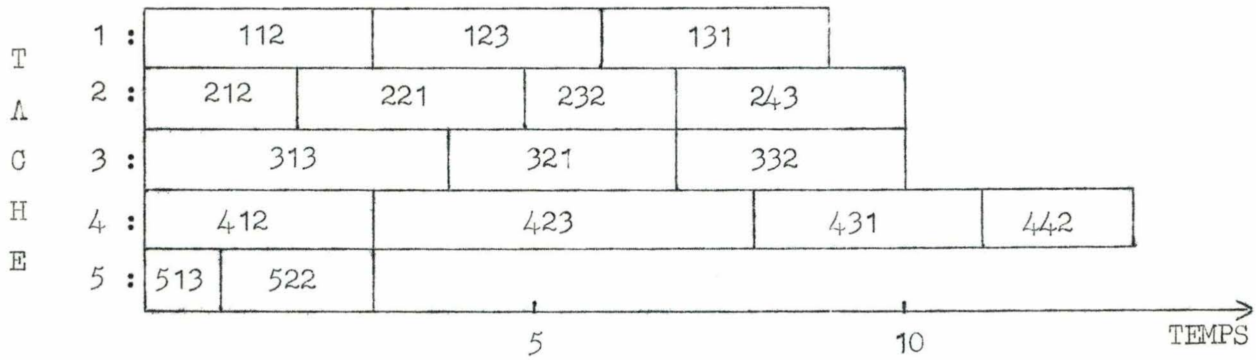
Nous avons montré la différence de nature entre les arcs verticaux (liens entre les opérations à traiter par une machine) dus aux contraintes disjonctives et les arcs horizontaux (liens entre les opérations d'une tâche) dus aux contraintes de routing. Nous l'avons fait en prenant un exemple d'assemblage, mais nous aurions pu envisager le cas où aucun routing n'était imposé ; l'idée prédominante est que l'ordonnancement détermine un ordre strict, tandis que le routing pas nécessairement : dans le cas ci-dessus, deux chemins sont possibles :

$$\begin{cases} x1-x2-x4-x3-x5-x6 \\ x1-x2-x4-x5-x3-x6 \end{cases}$$

Pourtant, la plupart des praticiens traitant de l'ordonnancement ne font pas appel à la théorie des graphes pour visualiser et sytématiser ce problème ; ils lui ont préféré la représentation graphique appelée diagramme de Gantt pour des raisons essentiellement pratiques : les caractéristiques de chaque commande, énumérées sur une "feuille de route" (durée, équipement, nécessaire, date de livraison, routing ...) sont reprises sur des morceaux de papier (un par opération). Les pièces de ce puzzle sont alors manipulées par les responsables de l'ordonnancement de manière à obtenir un ordonnancement satisfaisant (l'axe horizontal de ce puzzle représente le temps). Le puzzle réalisé permet d'établir un ordre du jour remis aux responsables de l'exécution des commandes. Si, pour une raison ou une autre, l'avancement réel des travaux ne correspond pas à leur avancement théorique, il sera toujours facile, grâce aux pièces du "puzzle", de remanier l'ordonnancement en fonction des nouvelles données ; on trouvera dans l'annexe II un aperçu de l'origine et des applications de ce diagramme.

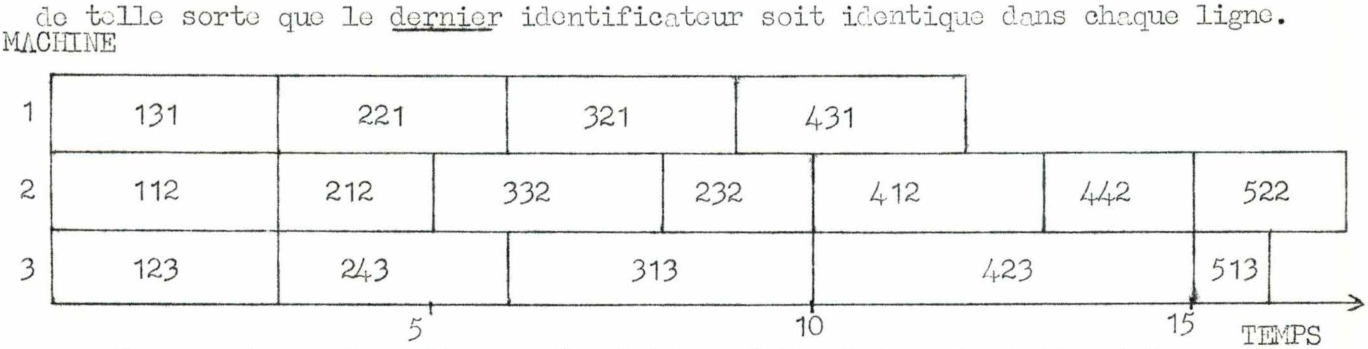
Adapté à l'ordonnancement non unitaire, le diagramme de Gantt prend deux formes qui expriment à leur manière les matrices de routing et d'ordonnancement.

Chaque tâche i se décompose en g_i blocs, un par opération ; chacun d'eux possède trois identificateurs : i,j,k, suivant le numéro de la tâche, l'ordre de séquence de l'opération et le numéro de la machine ; la longueur du bloc est proportionnelle à la durée de l'opération. L'ensemble des opérations peut être partitionné par le premier ou par le troisième identificateur en sous-ensembles disjoints, exhaustifs et mutuellement exclusifs appelés respectivement tâches et machines. Sur cette base, le diagramme-tâche consiste à arranger sur une ligne les blocs appartenant à une même tâche.



- le premier identificateur de chaque opération doit être identique à l'intérieur d'une même ligne ;
- le second identificateur forme une séquence croissante dans chaque ligne, indiquant par là les contraintes de routing.

Le diagramme-machine réarrange ces blocs en lignes-machines (et non plus en lignes-tâches), sans changer les identificateurs ni la longueur des blocs, de telle sorte que le dernier identificateur soit identique dans chaque ligne.



Les avantages de cette représentation résident dans le fait qu'elle :

- permet de se rendre immédiatement compte de la longueur de la file d'attente devant chacune des machines ;
- permet de situer les machines qui forment des goulots d'étranglement, c'est-à-dire celles dont le chargement dépasse la moyenne des autres ;
- donne la durée minimum d'occupation de chaque machine et la limite inférieure en dessous de laquelle il est impossible de terminer toutes les tâches ; en d'autres termes, elle permet de répondre à la question de savoir quand les tâches auront quitté l'atelier ;
- visualise les contraintes disjonctives puisque, à tout instant, il n'y a qu'une opération sur une même machine ;
- rend le calcul des mesures d'évaluation (voir infra) très simples.

Bien que, dans ce diagramme-machine, les contraintes disjonctives soient respectées, les contraintes de routing ne le sont pas. Les opérations 221 et 212 sont traitées en même temps. L'ordonnancement non unitaire consiste à provoquer un réarrangement tel que, si on voulait réarranger les opérations de ce diagramme-machine dans un diagramme-tâche, les opérations seraient dans leur ordre original (sans se recouper) ce qui, dans notre exemple, donne :

Diagramme machine

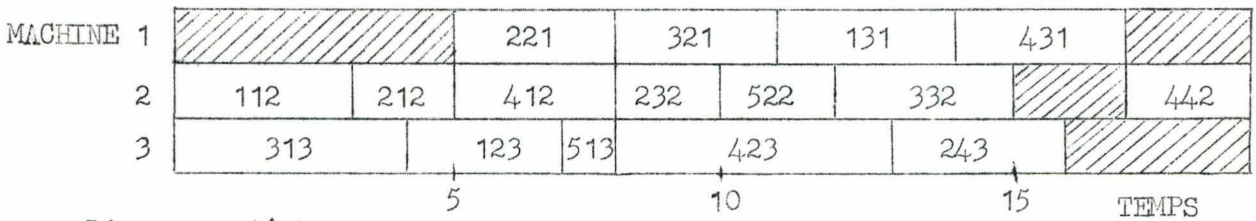
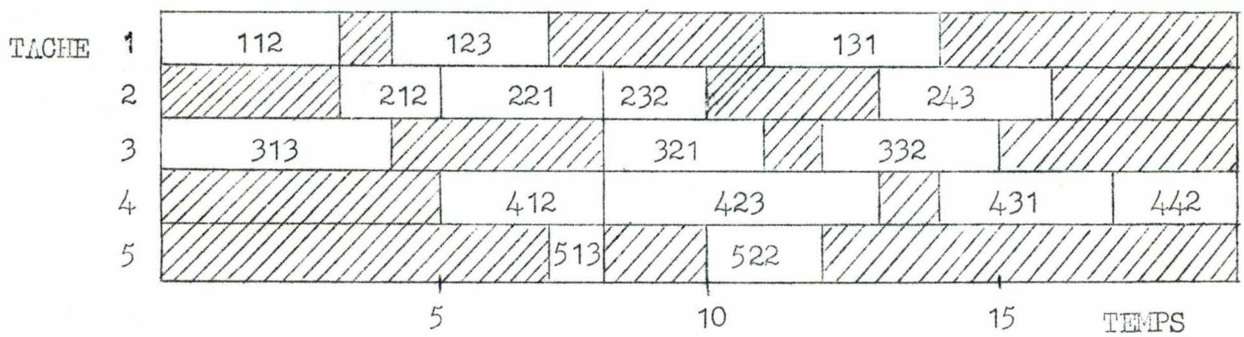


Diagramme tâche



Pour vérifier les contraintes disjonctives et de routing, il faudra 19 unités de temps avant que toutes les tâches ne soient terminées.

Dans la présentation de l'ordonnancement par la théorie des graphes, nous avons parlé d'une matrice de permutation ; ici, nous n'avons pris qu'un ordonnancement-machine ; si nous n'avons rien dit en ce qui concerne la provenance de cette matrice et de cet ordonnancement, c'est parce qu'ils sont totalement arbitraires. Sous réserve de considérations que nous établirons plus tard, le nombre de matrices de permutation approche rapidement de l'infini, leur nombre s'élevant à $N = (n!)^m$. En effet, s'il y a n opérations à placer à n endroits, la théorie combinatoire nous apprend qu'il s'agit d'une permutation de n éléments dont la valeur est n!. Comme ces permutations peuvent se réaliser sur les m machines, le nombre total de possibilités s'élève à : n!.n!.n!.n!.n! ... (m fois) ou $(n!)^m$. Un atelier comportant 5 machines et ayant 4 opérations à traiter par machine nécessite l'examen de $(5!)^4$ = 207 360 000 ordonnancements différents ! L'exactitude de ce nombre suppose la vérification des hypothèses suivantes :

- le nombre d'opérations par tâche est égal au nombre de machines. Dans le cas inverse, le nombre de possibilités s'élève à $g_1!.g_2!.g_3!\ldots g_n!$ où g_k = nombre d'opérations à traiter par la machine k (k = 1.......m) ;
- aucune opération n'est fractionnée ; dans le cas inverse, le nombre de possibilités serait beaucoup plus élevé ;
- il y a autant de machines que d'opérations par tâche, chaque opération étant assignée à une machine différente ; dans le cas contraire, il n'y aurait pas indépendance entre les opérations à réaliser par une même machine, par l'intervention des contraintes de routing (voir ci-dessus dans le diagramme-machine pour les opérations 412 et 422) ;
- les tâches sont calées le plus à gauche possible.

Notons que ces hypothèses ne sont pas souvent respectées sauf la dernière, de telle sorte que l'on peut considérer le nombre théorique obtenu comme une limite inférieure. Nous reviendrons sur l'ensemble de ces hypothèses, mais la dernière nous paraît essentielle, car elle n'est pas toujours explicitée. L'ordonnancement consiste à établir, par machine, un séquence de traitement des opérations en fonction d'un critère à optimiser ; la recherche de cet optimum empêche "d'élargir" le diagramme-machine en y insérant des durées d'innoccupation-machine qui ne soient pas nécessaires. L'expression "élargir, par machine, une séquence..." ne signifie pas uniquement "mettre les opérations dans un certain ordre", mais aussi "assigner à chacune d'elles une date qui spécifie sa date de départ ; on pourrait garder le même ordre, mais retarder systématiquement leur départ de t unités de temps : on respecterait toutes les contraintes et on aboutirait à un nombre infini d'ordonnancements par machine ; cette éventualité est écartée par la recherche d'un optimum.

Comme nous le signalions au début de ce chapitre, la théorie des graphes et la présentation graphique permettent la visualisation d'<u>un</u> ordonnancement, mais elles ne permettent pas de résoudre des problèmes d'une certaine dimension. Il ne suffirait pas de calculer le nombre d'ordonnancements possibles ; il faudrait encore énumérer chacun d'entre eux, en établir le diagramme, calculer pour chacun la valeur prise par la fonction-critère adoptée et enfin trouver <u>l</u>'ordonnancement qui donne la valeur optimale.

<u>Section 2</u> : <u>Les hypothèses de base des modèles d'ordonnancement</u>

Aux définitions concernant les tâches, les opérations, les machines et la manière de construire un ordonnancement, il faut ajouter des restrictions supplémentaires ; au fur et à mesure de leur énumération, nous indiquerons leurs conséquences quant à la possibilité d'application des résultats théoriques à une situation particulière. Nous supposons préalablement qu'il n'y a qu'<u>une</u> ressource limitée : les machines ; on pourrait concevoir qu'une opération exige simultanément une machine, une équipe de main-d'oeuvre et un outillage particulier ; dans ce cas, l'assignation d'une opération ne pourrait être faite que lorsque la machine, l'équipe et l'outillage sont disponibles <u>simultanément</u>.

Hypothèses :

1. Chaque machine est continuellement disponible : cela signifie deux choses :

   - il n'y a pas de division significative de l'échelle de temps (en jours, en heures...) : la période de temps couvrant l'ordonnancement est dépendante du nombre de tâches à réaliser et indépendante de l'existence des limites dues aux capacités des machines. Il ne faudrait pas croire que cette hypothèse soit fort restrictive, car il suffit que les responsables de la production ajustent leurs équipements aux exigences de l'ordonnancement : en modifiant le nombre de machines si certaines d'entre elles entraînent des goulots d'étranglement, en équilibrant les courbes de charge et, de façon plus générale, en découpant l'échelle de temps après que l'ordonnancement ait été établi.
   - on ne tient pas compte de l'indisponibilité passagère de certaines machines, pour des raisons de panne ou d'entretien ; il y a plusieurs manières de lever cette hypothèse, aucune n'étant sans inconvénients. Après avoir déterminé un coefficient d'indisponibilité, on pourrait répartir ce coefficient sur toutes les opérations à traiter par cette machine ; au temps de traitement normal, on ajoute un certain nombre d'unités de temps au prorata de la durée de ces opérations ; cette méthode pénalise les opérations longues qui sont déjà pénalisées (voir infra) du fait de leur durée. Inversément, diviser le temps total d'indisponibilités par le nombre d'opérations pénalise les opérations courtes.

   En résumé, chaque machine, est un simple intervalle de temps (0, T) où T est un nombre arbitrairement élevé.

2. Il n'y a qu'une machine de chaque type dans l'atelier : les numéros d'identification des machines sont uniques et représentent plus un type de machine qu'une machine particulière. Cette hypothèse est une conséquence de la première où chaque machine est un simple intervalle de temps.

3. <u>Une machine ne peut traiter qu'une opération à la fois</u> : cette hypothèse très restrictive "en soi" ne limite cependant pas l'application du problème d'ordonnancement non unitaire, car elle correspond à la nature même du problème ; c'est à cause de l'existence de contraintes disjonctives qu'un problème d'ordonnancement <u>non unitaire</u> existe.

4. <u>Une opération commencée doit être terminée</u> : une tâche en cours d'exécution en un poste donné ne peut pas être interrompue par des travaux de priorité supérieure ; on ne peut faire jouer les priorités que lorsqu'une machine est inoccupée. Restrictive dans le cas où les époques d'arrivée des commandes sont inconnues ou ne sont connues que selon des distributions de probabilité, elle l'est moins dans le cas statique (voir infra). Cette hypothèse, en interdisant le <u>fractionnement des opérations</u>, facilite énormément le problème en diminuant de façon appréciable le nombre d'ordonnancements à envisager ; elle reste spécialement contraignante lorsque la variance des durées de traitement est élevée : une opération nécessitant une longue durée de traitement risque d'entraîner, lors de son traitement, une longue période d'immobilisation sur les autres machines (à cause des contraintes de routing) et, par conséquent, le retard d'opérations courtes. On remédie à cet inconvénient en basant la politique d'ordonnancement sur la longueur des durées de traitement.

5. <u>Une série ne peut être fractionnée</u> : chaque tâche est une entité et, même si elle constitue un lot de parties individuelles, un lot ne peut être traité sur plusieurs machines à la fois. Cette hypothèse rend les contraintes de routing encore plus strictes ; ce cas se présente lorsque les tâches concernent la fabrication de petites pièces identiques et indépendantes les unes des autres ; on pourrait admettre que, lorsque certaines pièces ont subi le premier stade de fabrication, elles puissent passer directement au second stade sans attendre que toutes les pièces de la commande aient subi le premier stade de traitement. Pour pallier

cet inconvénient, il suffirait en théorie de redéfinir les opérations en considérant non plus le lot, mais chaque part individuelle comme <u>une</u> opération ; en pratique, cette méthode entraînerait un nombre infini d' opérations à définir. Pour les mêmes raisons, cette condition exclut les cas d'assemblage, puisque certaines pièces jusqu'alors traitées indépendamment, sont ensuite réunies. Cette hypothèse se vérifie facilement sur le diagramme de Gantt, puisque, à tout instant, on ne peut trouver qu'une seule fois le même indice-tâche par unité de temps.

6. <u>La durée de traitement d'une opération est connue soit avec exactitude soit en probabilité</u> : dans le cas statique, (voir infra), on utilisera des estimateurs ou la simulation lorsque les durées de traitement sont connues en probabilité.

7. <u>Les intervalles de temps sont indépendants de l'ordre dans lequel les opérations sont réalisées</u> : cette hypothèse exclut les durées de préparation rendues nécessaires lors du changement de la gamme de production. Prenons le cas d'un atelier de peinture qui doit être nettoyé chaque fois que l'on change de couleur ; cette durée de nettoyage varie selon la couleur qui <u>était utilisée</u> et la <u>nouvelle couleur à adopter</u> : elle sera, en effet, plus élevée s'il faut passer du noir au blanc que du bleu au vert ; on ne peut donc dans ce cas ajouter cette durée de préparation à la durée de traitement d'une tâche.

8. <u>Le "routing" de chaque commande ne permet pas d'alternative</u> : l'hypothèse 5 concernait des cas particuliers de cette hypothèse. Nous supposons que les tâches ont été préalablement décomposées en opérations élémentaires au sens le plus strict du terme et que la suite des opérations ainsi que l'assignation de chacune d'elles à une machine spécifique a été faite. Introduire cette hypothèse réduit le nombre de conflits qui se présentent devant une machine, puisque, au maximum, <u>chaque</u> tâche n'y sera représentée que par <u>une</u> de ses opérations. Cette hypothèse est à

rapprocher de la deuxième, car dans le cas de machines "équivalentes", la même opération d'une tâche pourrait attendre devant chacune d'entre elles, bien qu'elle ne soit traitée que par l'une d'entre elles. Cela n'exclut cependant pas la possibilité de traiter plusieurs opérations non consécutives d'une même tâche sur une même machine.

9. En respectant uniquement les contraintes de routing, une opération doit commencer le plus tôt possible : cela signifie qu'il n'est pas admis de laisser une machine inoccupée pour assurer le traitement futur d'une opération à priorité supérieure, lorsqu'une autre opération est susceptible d'y être traitée maintenant. Cette hypothèse rejoint également ce que nous avons dit à la fin de la première section à propos de l'élargissement de l'échelle de temps dans le diagramme de Gantt.

10. Les dates d'arrivée des tâches sont connues avec certitude : cette hypothèse, apparemment restrictive, est pourtant levée en distinguant deux types de problèmes d'ordonnancement non unitaire : les problèmes statiques et les problèmes dynamiques (voir section 4).

11. Les durées de transport d'une machine à l'autre sont négligeables : dans le cas d'un routing strict, c'est-à-dire lorsque l'on sait sur quelle machine l'opération suivante devra être réalisée, on incorpore ces durées de transport dans la durée de traitement des opérations. Cette hypothèse a quelque relation avec l'hypothèse 9, mais nous y reviendrons en mentionnant le modèle de Mitten.

Les premiers modèles d'ordonnancement supposaient la vérification de toutes ces hypothèses et les rendaient très peu applicables à la réalité d'une entreprise ; mais c'est en partant de modèles aussi "théoriques", que les recherches actuelles ont pu juger de leur aspect restrictif et ont cherché à y remédier. Les modèles généraux n'existent pas et on a pu tout au plus établir des modèles où une, deux ou trois de ces hypothèses étaient levées.

Section 3 : L'évaluation et les critères d'optimalité en matière d'ordonnancement

§ 1. Les mesures d'évaluation

La variété des critères employés pour mesurer la valeur d'un ordonnancement reflète la variètè des circonstances où des problèmes d'ordonnancement se posent ainsi que celle des coûts supposés significatifs. Il faut distinguer entre deux types de variables :

a. les variables qui définissent le problème (données et paramètres) : notées par une lettre minuscule, elles sont données :

r_i : date d'arrivée de la tâche i ; elle indique soit le moment d'arrivée de la tâche dans l'atelier soit la date à laquelle, au plus tôt, la première opération de cette tâche peut commencer ;

d_i : date de livraison : date à laquelle, pour des raisons externes, une tâche devrait avoir quitté l'atelier, ou la date à laquelle la dernière opération de cette tâche devrait être achevée pour ne pas entraîner de retards ;

a_i : marge de la tâche i : $= d_i - r_i$; c'est la durée totale pendant laquelle une tâche peut rester dans l'atelier sans entraîner de retards ;

p_i : durée de traitement total de la tâche i ;

$$p_i = \sum_{j=1}^{g_i} p_{ij}$$

b. les variables décrivant la solution du problème (variables de décision) : notées par une lettre majuscule, elle dérivent toutes d'un même concept : la durée d'attente d'une tâche : la question de savoir dans quel ordre les opérations doivent être traitées sur une machine, revient à déterminer pendant combien de temps chaque opération doit attendre son traitement ; dans le diagramme-machine cidessus (section 1), c'est la partie hachurée précédant le traitement d'une opération :

W_{ij} = durée d'attente précédant le traitement de l'opération de la tâche i ou la durée séparant le moment de l'achèvement de l'opération (j - 1) et le début de l'opération j ;

$W_i = \sum_{j=1}^{g_i} W_{ij}$ = <u>durée totale d'attente de la tâche i</u>

Le résultat d'un processus d'ordonnancement est complètement spécifié par l'ensemble des W_{ij} et les autres variables ne sont introduites que par facilité et parce qu'elles ont plus de signification :

1. <u>le moment où la tâche i quitte l'atelier</u> : C_i

$$C_i = r_i + W_{i1} + p_{i1} + W_{i2} + p_{i2} + \dots + W_{ig_i} + p_{ig_i}$$
$$= r_i + \sum_{j=1}^{g_i} (p_{ij} + W_{ij}) = r_i + p_i + W_i$$

2 . <u>la durée totale passée dans l'atelier par la tâche i</u> : F_i

$$F_i = W_{i1} + p_{i1} + W_{i2} + p_{i2} + \dots + W_{ig_i} + p_{ig_i}$$
$$= \sum_{j=1}^{g_i} (p_{ij} + W_{ij}) = p_i + W_i = C_i - r_i$$

3. <u>l'écart entre le moment auquel une tâche en traitement quitte réellement l'atelier et le moment où elle devrait le quitter</u> (d_i) ; il y a trois manières de faire cette comparaison :

- calculer cet écart pour chaque tâche, indépendamment du signe de la différence :

$$L_i = C_i - d_i = F_i + r_i - d_i = F_i - a_i$$

- calculer cet écart lorsqu'il est positif : on ne s'intéresse qu'aux tâches terminées <u>après</u> leur date de livraison :

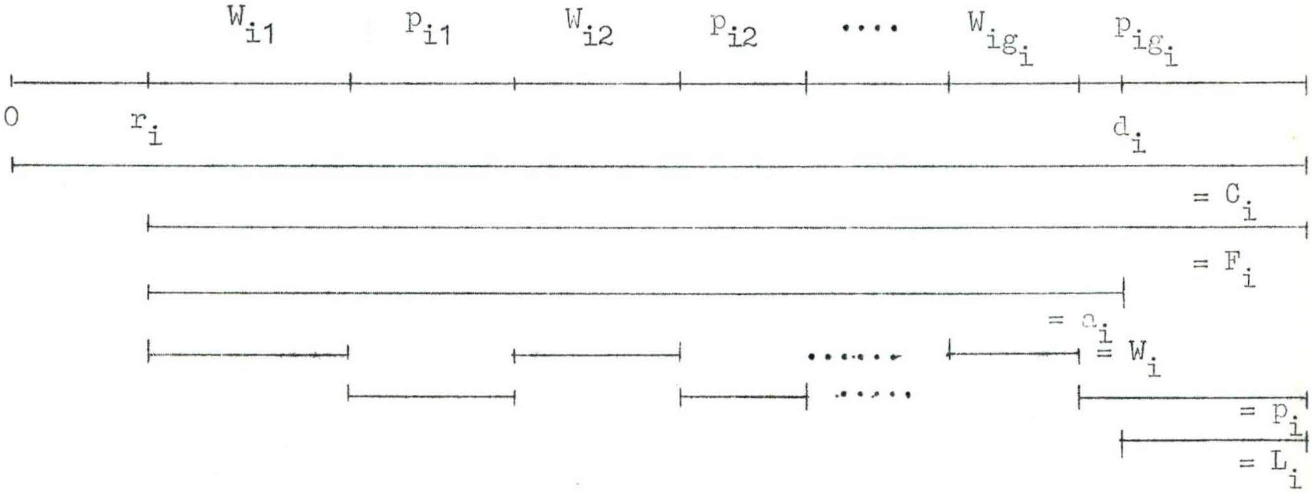
$T_i = \max(0, + L_i)$; T_i est donc le retard de la tâche i

- calculer cet écart lorsqu'il est négatif : on ne s'intéresse qu'aux tâches terminées <u>avant</u> leur date de livraison :

$E_i = \max(0, - L_i)$; E_i est donc l'avance de la tâche i

En général, la deuxième mesure est la plus souvent utilisée car elle entraîne des coûts de retard. Cela n'empêche que l'emploi de la première permet de se rendre compte si, <u>en moyenne</u>, les dates de livraison ont été bien prédéterminées (en arrivant à $\sum_{i=1}^{n} L_i = 0$). L'emploi de la troisième mesure permet éventuellement de raccourcir le délai de livraison, d'accepter un plus grand nombre de tâches en d'avoir un coefficient d'utilisation des machines plus élevé. La seconde mesure est plutôt passive.

Les relations entre ces variables se présentent de la manière suivante (pour tout i) : [1]



NB : $L_i = T_i$ puisque la tâche est en retard

c. <u>Remarques</u> :

1. Jusqu'à présent, nous avons évalué un ordonnancement sur base des tâches, mais on l'évalue parfois sur base de certaines caractéristiques du centre de production ; les deux principales sont :

   - <u>le coefficient d'utilisation des machines</u> : c'est la fraction de la capacité de la machine utilisée pour le traitement des opérations, c'est-à-dire le rapport entre les durées de traitement et la durée totale disponible. Dans un processus continu (ou dynamique : voir infra) où les machines sont "éter-

[1] Tous les symboles mathématiques sont repris dans l'appendice

nellement" disponibles, l'utilisation moyenne est un paramètre du problème et n'est pas affecté par l'ordonnancement (au contraire, celui-ci est affecté par ce coefficient). Dans les problèmes finis ou statiques, on suppose que toutes les machines sont disponibles pour le traitement des n tâches (c'est-à-dire depuis le minimum des r_i jusqu'au maximum des C_i). Dans ce cas, le coefficient moyen d'utilisation se définit comme étant égal au rapport entre la durée totale de traitement nécessaire et le nombre de machines multiplié par la durée maximum d'utilisation d'une machine (F_{max})

$$\overline{U} = \frac{\sum_{i=1}^{n} p_i}{nF_{max}}$$

- le travail en cours : suivant le problème, on peut être intéressé soit par le nombre moyen de tâches présentes dans l'atelier, soit par "le volume des travaux" (en heures) d'un atelier.

  i) Si N(t) est le nombre de tâches dans l'atelier au moment t et si $\overline{N}(t_1, t_2)$ est le nombre moyen de tâches dans l' atelier pendant l'intervalle de temps $t_1 - t_2$ on a :

$$\overline{N}(t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} N(t)dt$$

  ii) Si P(t) est la somme des durées de traitement de toutes les opérations présentes dans l'atelier au moment t et si $\overline{P}(t_1, t_2)$ est le contenu moyen de travail dans l'atelier pendant l'intervalle de temps $t_1 - t_2$, on a :

$$\overline{P}(t_1, t_2) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} P(t)dt$$

On pourrait encore fractionner $\overline{N}(t_1, t_2)$ et $\overline{P}(t_1, t_2)$,en tenant compte des tâches en traitement (p) et des tâches en attente (q).

$$\overline{N}(t_1, t_2) = \overline{N}_p(t_1, t_2) + \overline{N}_q(t_1, t_2)$$

2. Quels sont les coûts associés aux décisions d'ordonnancement ?

l'hypothèse suivant laquelle l'ensemble des tâches est prédéterminé et inaffecté par les décisions d'ordonnancement implique que le revenu total de l'entreprise est fixé ou du moins est inaffecté par l'ordonnancement. En conséquence de quoi, le revenu n'est pas considéré comme une variable significative du problème.

Les hypothèses suivant lesquelles l'équipement est donné et la manière dont il est utilisé n'est pas affectée par les décisions d'ordonnancement, impliquent que tous les coûts directs sont sans signification dans notre étude.

Certains coûts, par contre, sont les conséquences immédiates des décisions concernant la séquence des opérations sur chaque machine. Parme ceux-ci, il faut citer :

- les coûts de stockage intermédiaire : minimer ces coûts suppose que $\overline{U}$ (coefficient moyen d'utilisation) soit le plus élevé possible et que la valeur de $\overline{N}_p$ soit une fraction plus ou moins fixe de $\overline{N}$. En d'autres termes, il faut minimer une certaine fonction des durées moyennes d'occupation d'un atelier par les différentes tâches ;
- les coûts d'inoccupation des machines : ces coûts, que l'on pourrait qualifier "d'opportunité", peuvent être minimés de la même manière que les coûts de stockage intermédiaire.
- les coûts de retard : la difficulté d'évaluation de ces coûts provient de l'immatérialité de certaines de ses composantes. Interviennent dans ces coûts, les salaires dus aux heures supplémentaires, les coûts de sous-traitance, la perte de "good-

will" (probablement le plus important, mais le moins quantifiable) les pénalités contractuelles ...
La minimisation de ces coûts suppose une meilleure utilisation de l'équipement et la minimisation d'une certaine fonction des retards.

§ 2. Les critères d'optimalité

De ces considérations sur les coûts, il ressort que les critères fondamentaux sont d'une part : minimiser la durée d'occupation de l'atelier par les différentes tâches et d'autre part minimiser les retards. Pourtant, les valeurs moyennes offrent des inconvénients, car ce qui nous intéresse particulièrement ce sont les extrêmes (F_{max} et T_{max}) :

- la valeur de $\overline{F}$ (durée moyenne passée dans l'atelier par une tâche) ne donne pas une idée exacte des goulots d'étranglement qui peuvent se présenter ; c'est pourquoi le critère d'optimalité min $\overline{F}$ est avantageusement remplacé par le critère min F_{max} ; la mesure d'évaluation F_{max} représente la durée maximale d'occupation de l'atelier par une tâche ; sur le diagramme de la section 1, F_{max} est égal à 19 et est à imputer à la tâche 4 ; l'examen de cette tâche montre que deux de ses opérations sont à traiter par la machine 2 et l'on peut dès lors se demander si ce n'est pas au niveau de cette machine que se situe le goulot d'étranglement le plus important ; ces goulots risquent d'entraîner des retards considérables et par le fait même des coûts. Parmi tous les ordonnancements possibles, il faut prendre celui qui minime F_{max} ; min F_{max} est donc le critère qui minime le temps nécessaire pour traiter toutes les tâches présentes dans l'atelier.

- la valeur $\overline{T}$ (retard moyen d'une tâche) ne donne pas une idée exacte des coûts associés à ces retards, surtout lorsqu'ils sont une fonction exponentielle de la durée des retards ; pour cette raison, on préfère prendre comme critère d'optimalité min T_{max}, en attirant ainsi l'attention sur les tâches qui accusent les retards les plus importants.

Ces deux critères (minimisation du temps total de traitement des n tâches et respect des dates de livraison) sont les critères auxquels les théoriciens se sont le plus intéressés et sur lesquels nous nous baserons presque exclusivement. Cela ne doit pas nous faire oublier l'existence d'un grand nombre d'autres critères ; ils n'ont pas toujours une raison d'être <u>économique</u> bien précise ; ces critères sont entre autres min $\overline{F}$, min $\overline{L}$ min $\overline{W}$, min $\overline{T}$, min F_{min}, min C_{min}, min W_{max}, max L_{min}, max F_{min} ....

Puisque toutes les mesures d'évaluation basées sur les tâches dérivent des durées d'attente, on serait tenté de croire qu'un ordonnancement qui minimise une de ces mesures minimise automatiquement toutes les autres ; c'est exact dans la mesure où deux conditions sont simultanément remplies :

- les mesures diffèrent l'une de l'autre par des constantes, ce qui est le cas pour F, C, W ;
- les mesures sont <u>régulières</u> : une mesure est régulière lorsqu'elle peut s'exprimer comme une fonction des dates auxquelles les tâches quittent l'atelier (C_i) ET lorsque la valeur de cette mesure s'accroît si une de ces dates au moins augmente.

C et F sont des mesures régulières, mais des mesures plus complexes faisant appel à des moyennes pondérées et à des combinaisons de moyennes et de maxima ne le sont pas :

- un ordonnancement optimal quant à $\overline{F}$ l'est aussi quant à $\overline{L}$, mais on ne peut rien en conclure quant à L_{max} **dont l'ordonnancement optimal l'est aussi en fonction de F_{max}.**
- même si $T_i - E_i = L_i$ et si $\overline{T} - \overline{E} = \overline{L}$, cela ne signifie pas qu'un ordonnancement qui minime $\overline{L}$, minimise aussi $\overline{T}$ (sauf si toutes les tâches sont en retard)

Spécifier une mesure d'évaluation dans un problème d'ordonnancement revient à spécifier un ensemble de classes d'ordonnancements équivalents et à établir un ordre de préférence entre elles ; la mesure d'évaluation C signifie donc que :

- tous les ordonnancements ayant le même C sont équivalents,
- un ordonnancement dont la valeur est de C' sera préféré à un ordonnancement dont la valeur est C" si et seulement si

  $C' < C''$ dans le cas d'un critère à minimiser

  $C' > C''$ dans le cas d'un critère à maximiser.

Un ordonnancement est dit "optimal" eu égard à une mesure de performance s'il appartient à une classe d'équivalence $\{S'\}$ tel qu'il n'existe pas de classes qui sont préférées à $\{S'\}$.

Section 4 : Classification des modèles d'ordonnancement

Les manières de classifier ces modèles sont aussi nombreux que les critères de classification eux-mêmes (les hypothèses retenues dans le modèle, le nombre de machines, de tâches, le genre de routing, la mesure de performance utilisée).

Nous utiliserons la classification de Conway [1] pour qui un problème d'ordonnancement spécifique se décrit par quatre types d'informations :

A. La nature des arrivées des tâches dans l'atelier : les problèmes sont alors ou statiques ou dynamiques :

   - dans un problème statique, les tâches arrivent dans l'atelier soit en même temps, soit à des dates connues; l'atelier est vide

---

[1] CONWAY, R.W., MAXWELL, W.L. et MILLER, L.W. : op. cit. p. 6 ss

et disponible ; aucun travail supplémentaire ne se présentera pendant la période nécessaire à l'ordonnancement de ces tâches ;

- dans un problème dynamique :
  - l'atelier est un processus continu, de sorte qu'il n'est pas nécessairement vide au départ ;
  - les travaux arrivent de façon intermittente et les dates de leur arrivée sont des variables stochastiques ;
  - les travaux supplémentaires sont incorporés dès leur arrivée.

- Le problème statique ou déterministe se caractérise donc par une connaissance parfaite des dates d'arrivée des travaux, ce qui permet la construction d'un ordonnancement avant d'entamer la production réelle (même s'il faut recourir à des estimateurs pour la durée des opérations). Ce genre de problème est réaliste dans la mesure où :
  - suivant l'unité de temps utilisée, la période n'est pas trop longue : on peut concevoir qu'un carnet de commandes recense les travaux à réaliser et qu'à la fin de la période (journée ou semaine) l'ordonnancement de ces travaux est construit : on ne prendra QUE les travaux reçus pendant cette période (plus éventuellement, les travaux des périodes précédentes qui n'ont pu être terminés). En résumé, les travaux reçus pendant la période seront traités à la période t + 1 ;
  - la fabrication se fait plus sur base de commandes que sur base de centres de distribution à réapprovisionner régulièrement.

- Dans le processus dynamique, les travaux arrivent selon des lois de probabilité et sont incorporés parmi les travaux à ordonnancer dès leur arrivée ; dans ce cas, l'ordonnancement est continu ;

des critères tels que min F_{max} ou min T_{max} n'ayant plus de signification, on utilisera des critères tels que l'espérance mathématique du nombre de travaux en attente, l'espérance mathématique de la durée d'attente, des retards ...

Nous n'étudierons pas les problèmes dynamiques parce que :

- leur examen dépasse le cadre de ce travail ;
- l'outil mathématique y est fort complexe (théorie des files d'attente) ;
- ils ne sont que très partiellement résolus : ils ne le sont que dans le cas d'une seule machine, ce qui réduit leur possibilité d'application.

Pour pallier ces inconvénients, on fait appel à la simulation dont nous discuterons plus tard.

B. Le nombre de machines dans l'atelier (m)

C. Les disciplines qui restreignent la manière dont les opérations se succèdent. L'ordre dans lequel les numéros d'identification des machines apparaît dans la suite des opérations d'une tâche détermine le type de routing (voir section 1) :

- chaque tâche a un nombre d'opérations inférieur ou égal au nombre de machine ET toutes les tâches suivent le même ordre, de sorte que la première opération de chaque tâche sera traitée sur la machine désignée 1... (ce type de routing est appelè en anglo-saxon "flow-shop"). Notons qu'il n'est pas nécessaire que chaque tâche passe sur chacune des machines ;
- aucun routing n'est imposé : chaque opération est assignée à une machine particulière, mais rien ne relie les opérations les unes aux autres (cas du "randomly routed job-shop")
- entre ces deux extrêmes, toutes sortes de routings sont possibles :

certaines opérations sont indépendantes, d'autres sont strictement reliées les unes aux autres (en anglo-saxon : "general routed job-shop").

D. Le critère d'évaluation : voir section 3

Conway établit alors une notation quadri-paramétrique pour identifier chaque problème particulier : A/B/C/D où

- A décrit le processus d'arrivée des tâches :
  - dans le cas statique, A identifie le nombre de tâches (sauf indication contraire, elles sont supposées arriver toutes en même temps ;
  - dans le cas dynamique, A indique la distribution de probabilité des intervalles entre deux arrivées successives ;
- B indique le nombre de machines utilisées ;
- C décrit le type de routing : F = flow shop

  R = randomly routed job shop

  G = general routed job shop ;
- D décrit le critère d'évaluation : principalement F_{max} et T_{max}.

Le modèle de Johnson [1] (voir infra) se note $n/2/F/F_{max}$ puisqu'il s'agit d'un modèle statique avec un nombre de tâches quelconque, à 2 machines, que le routing est de type strict (flow-shop) et que le critère à optimer est min F_{max}.

Section 5 : Méthodologie des chapitres suivants

Pour comprendre les lacunes des approches couramment utilisées pour résoudre les problèmes d'ordonnancement, lacunes qui entraînent la nécessité de

---

[1] JOHNSON, S.M. : "Optimal Two and Three-Stage Production Schedules with Setup-Times included" dans MUTH, Y.F. et THOMPSON, G.L. op. cit. chap. 2

reprendre le problème d'un tout autre angle (l'objet de la dernière partie), il nous faut voir en quoi consistent ces approches.

Il y a essentiellement deux manières de les considérer :

- chaque problème particulier nécessite une méthode d'approche particulière ; les avantages de cette manière de voir résident en ce qu'elle :
  . suit de façon chronologique les modèles qui ont été élaborés ;
  . part des modèles les plus restrictifs quant à leurs hypothèses de départ jusqu'aux modèles généraux où on tente de lever la plupart des hypothèses.

  Le nombre quasiment infini de problèmes particuliers en est malheureusement l'inconvénient essentiel ; pour s'en rendre compte, il suffit de reprendre la notation quadri-paramétrique de Conway et de donner à chaque variable une valeur particulière ; notons que chacun de ces problèmes se subdivise encore selon les hypothèses qu'il implique.

- on examine toutes les méthodes d'approche et on y associe les problèmes que ces méthodes sont susceptibles de résoudre. L'avantage de cette seconde manière de voir réside dans la réduction considérable des méthodes à prendre en considération. L'inconvénient de cette approche provient de ce que plusieurs méthodes ont été développées parce qu'un cas particulier se posait. D'autre part, il est dangereux de définir un cadre de méthodes mathématiques en essayant d'y intégrer le plus de cas possibles. En ce faisant, on risque de dénaturer les problèmes particuliers.

Nous utiliserons la première manière pour insister sur l'aspect "problème", mais pour mettre un certain ordre parmi les problèmes particuliers qui peuvent se présenter, on peut diviser les approches de résolution de ceux-ci en quatre "corps de méthodes" :

1. L'analyse combinatoire : cette expression, empruntée à Sisson [1] groupe toutes les approches qui font appel, de manière indépendante ou conjointe, à la théorie combinatoire, aux théories graphiques, mathématiques et statistiques, et à certains algorithmes de recherche opérationnelle. La diversité des méthodes est un indice de la diversité des problèmes auxquels elles s'appliquent. En raison du nombre d'hypothèses nécessaires et de sa limite essentielle -qui consiste à réduire le nombre de machines à un maximum de trois unités- l'application des résultats à des situations réelles est très rarement possible; malgré cela, leur étude a un intérêt scientifique évident, non seulement parce qu'elle forme la base de toutes les recherches ultérieures, comme nous venons de le signaler, mais aussi parce qu'elle permet de mettre le doigt sur les difficultés inhérentes au problème d'ordonnancement. La question qui se pose finalement est : "Ce problème est-il, par nature, résoluble ?" et si la réponse n'est pas purement négative : "La manière de l'aborder était-elle bonne ?" C'est à ces questions que nous aurons à répondre dans la dernière partie.

2. La programmation mathématique : lorsque les méthodes précédentes deviennent inefficaces par la disparition de la limite que constitue le nombre de machines utilisées, la programmation linéaire en nombres entiers et la programmation dynamique ouvrent de nouveaux horizons intéressants ; si, malgré le nombre important d'hypothèses qui subsistent, les résultats peuvent paraître satisfaisants (?), nous nous heurtons à une nouvelle difficulté d'ordre essentiellement pratique ; à quoi sert-il en effet de développer des théories techniquement au point, si leur application est impossible dans la majorité des cas ; l'application des méthodes combinatoires était rendue impossible parce que des situations aussi simplistes ressortent d'un domaine touchant

---

[1] SISSON, R.L. : "Sequencing Theory" dans ACKOFF,R.L. : "Progress in Operations Research" ; John Wiley and Sons, Inc. , N.Y. 1961 ; volume 1, chap 7.

à l'utopie ; ici, par contre, des situations permettant l'application de la programmation existent réellement, mais les méthodes pour exécuter cette application sont faibles. Les ordinateurs courants actuels n'ont pas une capacité de mémoire suffisante pour traiter des cas de dimension moyenne ; subsidiairement, il faut se demander si le coût de traitement d'un problème par ordinateur est suffisamment compensé par le coût d'un ordonnancement sous-optimal trouvé par simulation par exemple.

3. La simulation : résoudre des problèmes d'une certaine dimension sans subir les inconvénients précédents peut se faire par simulation ; par cette expression, nous entendons toutes les méthodes qui, d'une manière ou d'une autre, impliquent l'emploi de règles de priorité. Suivant le critère utilisé, la simulation permet de tester la valeur de différentes règles et aboutit à sélectionner celles qui paraissent significativement les meilleures. L'envers de la médaille vient de ce que la simulation n'aboutit que rarement à des résultats optimaux.

Chapitre 2 :

Les modèles d'ordonnancement de n commandes sur une machine (n/1/..)

Dans ce chapitre, nous étudierons les modèles d'ordonnancement relatifs au traitement de n commandes sur une seule machine.

- Dans la première section, nous examinerons les hypothèses particulières de ces modèles ; il nous faudra également discuter de l'intérêt de l'examen du cas où une seule machine est prise en considération : cet intérêt n'est pas à priori évident. Nous terminerons cette section par les tactiques utilisables lorsqu'un ordonnancement doit être dressé.

- Dans la seconde section, nous discuterons d'abord des résultats théoriques obtenus lorsque toutes les hypothèses énumérées dans le chapitre précédent sont d'application ; nous énumérons ensuite les avantages respectifs de l'utilisation de tel ou tel critère : les critères retenus sont essentiellement fonction de la durée d'occupation de l'atelier et des retards encourus par les différentes tâches. Sur base d'un exemple, nous jugerons enfin de la valeur respective de plusieurs ordonnancements possibles ; par cet exemple, nous voudrions souligner la complexité résidant dans le choix à priori d'un seul critère à optimiser : un ordonnancement optimal en fonction d'un seul critère peut s'avérer incompatible si on tient compte d'autres éléments (non retenus dans la fonction-objectif).

- Troisième section : si le choix d'un ordonnancement s'avère déjà difficile, il le devient encore plus lorsque l'on désire lever l'une ou l'autre des hypothèses restreignant l'application pratique des résultats théoriques obtenus. Dans cette section, nous lèverons quelques-unes de ces hypothèses et nous aurons dès ce moment l'occasion de constater la pauvreté des résultats, même lorsque l'ordonnancement ne concerne qu'une seul machine. Les hypothèses que nous tenterons de lever concernent les ordonnancements où :

    - les durées de traitement des opérations ne sont pas connues avec certitude ;
    - les tâches ne sont pas d'égale importance ;
    - les durées de préparation sont incorporées ;
    - les opérations arrivent de façon intermittente ;
    - il existe plusieurs machines du même type.

Section 1 : Le modèle général

§ 1. Introduction

A. Hypothèses particulières de ce modèle

1. Chaque tâche comporte une seule opération : les conséquences qui en découlent sont les suivantes :

   - chaque opération sera assignée à une machine particulière et, comme il y a indépendance entre les opérations, chaque machine pourra être ordonnancée séparément ; nous ne porterons donc notre attention que sur une machine ;
   - il n'y a pas de contraintes de routing : le troisième indice de la notation quadri-paramétrique de Conway n'a donc plus de raison d'être.

2. Le problème est statique et se caractérise par un nombre fini d'opérations $\{n\}$; dont les durées de traitement sont connues ; toutes les opérations doivent être traitées au cours de la période d'ordonnancement envisagée.

3. La machine est continuellement disponible et ne représente qu'un simple intervalle de temps de 0 à T, où T est le moment où toutes les opérations sont terminées.

4. Toutes les tâches sont disponibles au moment où la période d'ordonnancement commence à courir (nous noterons ce moment : 0). En d'autres termes, la séquence des opérations peut commencer avec n'importe laquelle des n opérations présentes, puisqu'il n'existe qu'un conflit de "passage" sur la machine (entre toutes les opérations à ordonnancer).

5. Il n'y a pas de durée de préparation-machine. Si elle existe, elle sera inclue dans la durée de traitement, ce qui suppose l'indépendance entre la durée de préparation et l'ordonnancement de deux tâches ordonnancées consécutivement.

B. Notations

$g_i = 1$: pour i variant de 1 à n

$r_i = 0$: pour i variant de 1 à n puisque toutes les tâches sont disponibles simultanément, ce qui permet de supposer qu'elles arrivent au moment où la période d'ordonnancement débute.

$p_{ijk} = p_i$: la notation des durées de traitement est simplifiée du fait qu'il n'y a qu'une opération par tâche et qu'une seule machine à envisager ; pour les mêmes raisons : $W_{ijk} = W_i$

$C_i = F_i$ et $a_i = d_i$ puisque $r_i = 0$ pour tout i.

$[i]$ = opération qui, d'après le choix de l'ordonnancement doit être traitée en ième position en séquence : un ordonnancement consiste à déterminer une séquence : $[1] - [2] - [3] - \ldots [n]$

$p_{[i]}$ est donc la durée de traitement de l'opération placée en ième position dans la séquence.

C. Intérêt de l'examen du cas d'une seule machine

1. d'un point de vue théorique :

- illustrer la différence de diverses procédures de résolution et les conséquences du choix d'une mesure d'évaluation particulière,
- montrer que ces modèles forment la base de procédure d'approche pratique ;

2. d'un point de vue pratique : ces modèles sont d'application dans le cas :

- où un atelier de production ne comporte qu'une seule machine : cas trivial,

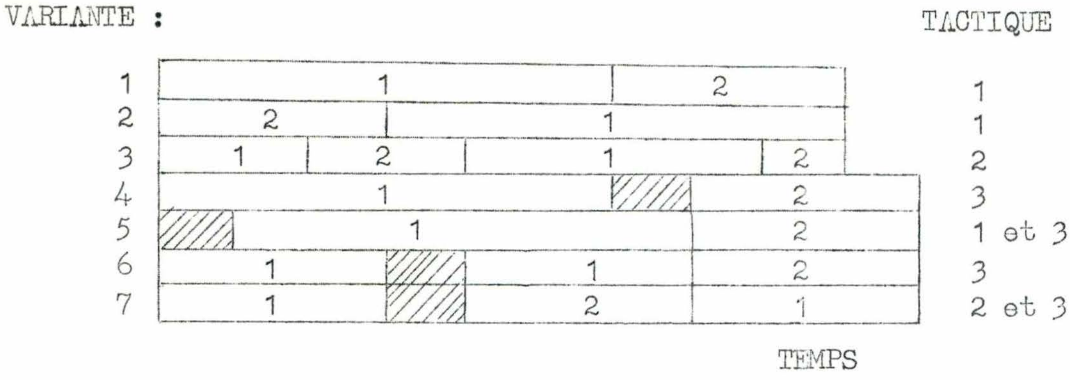
- d'un équipement complexe qui se comporte comme s'il n'y avait qu'une seule machine (chaîne continue),
- de plusieurs machines, mais où une paraît si importante qu'elle est envisagée comme si les autres n'existaient pas,
- où une machine forme un goulot d'étranglement.

D. Les ordonnancements de permutation

Dans une situation plus complexe que celle envisagée ici, trois tactiques d'ordonnancement sont possibles, indépendamment de l'ordre dans lequel les opérations sont traitées :

- faire en sorte qu'il n'y ait pas d'interruptions entre le traitement des opérations : au moment où une opération se termine, une autre commence ; les opérations ne peuvent être interrompues pendant leur traitement,
- interrompre le traitement d'une opération avant qu'elle ne soit achevée, lui en substituer une autre pour reprendre le traitement de la première,
- introduire des temps morts soit à la fin d'une opération, soit au cours de son traitement.

Dans le cas de deux opérations, ces situations donnent lieu à un nombre infini d'ordonnancements. La première tactique donne lieu à deux ordonnancements : (2,1) et (1,2) ; par la seconde, le nombre d'ordonnancements dépendra du nombre d'interruptions, nombre qui peut être très important mais fini ; par la troisième, le nombre d'ordonnancements est infini, vu le nombre de périodes de temps morts qu'il est possible d'insérer. Certaines des variations possibles sont représentées dans le diagramme suivant :



Dans le cas de <u>plusieurs</u> machines, toutes les variantes peuvent se présenter, mais, (comme nous l'avons signalé dans le premier chapitre) l'existence d'un critère à optimiser réduit ce nombre à une valeur finie : l'introduction de temps morts, qui ne sont pas dus aux contraintes de routing, ne constitue pas une politique optimale. Dans le graphique ci-dessus, les variantes 4,5,6,7 ne sont pas optimales, car elles allongent <u>sans raison</u> la durée d'occupation de l'atelier. Nous éliminerons donc la tactique d'insertion de temps morts. Le même graphique nous indique que les tactiques 1 et 2 (c'est-à-dire les variantes 1,2,3) sont équivalentes en ce qui concerne le critère F_{max}. Une démonstration plus rigoureuse prouve que :

<u>THEOREME 1</u> : lorsqu'on ordonnance un problème n/1 et lorsque la mesure d'évaluation est <u>régulière</u>, il n'est pas nécessaire de considérer les ordonnancements qui impliquent des interruptions ou l'insertion de temps morts.

Cela revient à dire qu'il suffit de tenir compte des <u>ordonnancements de permutation</u> (variantes 1 et 2), c'est-à-dire de ceux qui sont complètement spécifiés en donnant l'ordre dans lequel les opérations seront traitées dans leur totalité. Il y en a n!.

Section 2 : L'ordonnancement optimal

§ 1. Le critère d'optimisation est une fonction de la durée d'occupation de l'atelier par l'ensemble des n opérations (F)

1. F_{max} : durée d'occupation maximale de l'atelier : à quel moment les tâches auront-elles toutes quitté l'atelier ?

La détermination de cette valeur permet :

- de déceler les goulots d'étranglement se présentant sur une ou plusieurs machines et par là même leur inégalité de charge (sur -ou sous- investissement)

- de déterminer à quel moment une nouvelle série de commandes peut être mise en fabrication.

La durée de traitement des opérations étant connue, on déduit du théorème précédant que F_{max} est égal à la somme de toutes les durées de traitement, quelle que soit la position en séquence de ces opérations. Il y a donc indépendance entre la position en séquence d'une opération et sa durée de traitement.

2. $\overline{F}$: durée moyenne d'occupation de l'atelier par une tâche

Dans le cas dynamique, des critères tels que F_{max} n'ont aucune signification, puisque l'ordonnancement y est un processus continu. On le remplace alors par $\overline{F}$ ou plus exactement par l'espérance mathématique de la durée moyenne d'occupation de l'atelier par une tâche. Cette valeur aura un impact sur la longueur des files d'attente, sur le nombre de tâches en attente, sur l'acceptation de nouvelles commandes etc... $\overline{F}$ est pourtant un critère intéressant dans le cas statique :

- les critères F_{max} et $\overline{F}$ sont complémentaires, le premier reflétant l'état de l'atelier, le second celui des différentes tâches (F_{max} est déterminé par une seule d'entre elles, $\overline{F}$ par l'ensemble de celles-ci). Un ordonnancement optimal par rapport à l'un ne l'est pas nécessairement par rapport à l'autre.

- La valeur de $\overline{F}$ reflète indirectement le retard des tâches à ordonnancer, puisque $T_i = F_i - d_i$ et que d_i est une donnée.
- la minimisation de $\overline{F}$ limite les coûts de stockage intermédiaire, puisque les tâches quittent plus rapidement l'atelier.
- les durées d'inoccupation des machines ainsi que le nombre d'arrêts de celles-ci seront réduits en minimisant $\overline{F}$; cela permet d'accepter plus rapidement d'autres commandes qui ne doivent pas être traitées par la machine déterminant la valeur de F_{max}.

Comme $F_i = p_i + W_i$ et que p_i est consant, F_i ne peut être minimisé qu'en minimisant W_i. Si l'opération i est placée en première position, sa durée d'attente est nulle ($W_i = 0$), tandis que si elle est placée en seconde position, sa durée d'attente est égale à la durée de traitement de l'opération qui la précède ; et ainsi de suite pour les autres positions. En résumé, le temps que passe une opération dans l'atelier dépend de sa position dans la séquence ; si elle est placée en kième position $F_{[k]} = \sum_{i=1}^{k} p_{[i]}$; la durée d'occupation d'une opération placée en kième position dans la séquence est égale à la durée de traitement de cette opération plus la durée de traitement de toutes les opérations qui la précèdent, c'est-à-dire $[1], [2], \ldots, [k-1]$

Si on prend toutes les opérations et que l'on veut minimiser $\overline{F}$, il faut minimiser la durée totale d'attente (puisque les durées de traitement sont constantes) : on démontre que la règle de la plus courte opération (SPT) minimise $\overline{F}$. Cette règle consiste à placer les opérations dans l'ordre non décroissant des durées de traitement, c'est-à-dire selon une séquence $p \leqslant p_2 \cdots \leqslant p_n$

(cfr. Annexe III section 1) [1]

§ 2. Le critère d'optimisation est fonction des délais de livraison

Contrairement aux fonction de F, la règle SPT ne minimise pas les mesures d'évaluation, fonctions des dates de livraison. Ce ne serait le cas que si ces dates avaient été établies de telle sorte que les opérations étaient en retard ; rien d'étonnant à cela, puisque cette règle ne tient compte que des durées de traitement et non pas des dates de livraison.

Si l'objectif consiste à minimiser le retard maximum d'une opération (T_{max}) où l'écart maximum (L_{max}) entre le temps passé par une opération dans l'atelier (F_i) et sa date de livraison (d_i), Jackson [2] a démontré que:

THEOREME 2 : Lorsqu'on ordonnance un problème $n/1/L_{max}$ ou $n/1/T_{max}$, l'écart et le retard maximum d'une opération sont minimisés en ordonnançant les opérations dans l'ordre non décroissant des dates de livraison, c'est-à-dire de telle sorte que : $d_1 \leqslant d_2 \dots \leqslant d_n$

Si l'objectif consiste à maximiser le retard minimum (T_{min}) ou l'écart minimum (L_{min}) d'une opération, Jackson démontre que :

THEOREME 3 : Lorsqu'on ordonnance un problème $n/1/L_{min}$ ou $n/1/T_{min}$, le retard ou l'écart minimum sont maximisés en ordonnançant les opérations dans l'ordre du temps restant le plus court.

---

[1] Pour alléger le texte, nous avons préféré insérer les développements mathématiques en annexe (III)

[2] JACKSON J.R. : "Scheduling a Production time to Minimize Maximum Tardiness" Research Report 43 , MSRP, UCLA, janvier 1955

Le temps restant d'une opération est la différence algébrique entre sa date de livraison et sa durée de traitement à un moment donné :

$$s_i = d_i - p_i - t.$$

Exemple :

	durée de traitement	date de livraison
tâche 1	5	20
tâche 2	4	15

Les temps restants (slack time) sont respectivement de 15 et 11. Dans le cas statique, et dans le cas d'une seule opération par tâche, toutes les tâches sont disponibles au même moment ; la date à laquelle elles peuvent toutes commencer est donc égale à 0 (t = 0).

On ordonnance alors les opérations de telle sorte que : $(d_{[1]} - p_{[1]}) \leqslant (d_{[2]} - p_{[2]}) \dots \leqslant (d_{[n]} - p_{[n]})$. L'objectif max T_{min} n'est pas aussi étrange qu'il paraisse. Il peut en effet être intéressant de connaître le retard maximum qu'une opération pourrait subir LORSQU'on sait que cette opération a un retard <u>moindre</u> que les autres ; cela signifie que toutes les <u>autres opérations</u> subiront un retard plus <u>élevé</u> que celle-ci. Dans le premier cas au contraire (min T_{max}), on s'intéresse à l'opération dont le retard est <u>le plus élevé</u> par rapport à toutes les autres et, en minimisant ce retard, on sait que toutes les <u>autres opérations</u> auront un retard <u>moins important</u> (cette remarque s'applique aussi à min F_{max} et à max F_{min}).

Reprenons tout ceci sur base d'un exemple :

- numérotation des opérations :	1	2	3	4
- durée de traitement : p_i =	2	4	3	1
- date de livraison : d_i =	1	2	4	6

Règle	Séquence	Donnés	1	2	3	4	F	F_{max}	F_{min}	L	L_{max}	L_{min}	T	T_{max}	T_{min}
quelconque	1,3,4,2	F_i	2	10	5	6	5,75	10	2						
		L_i	1	8	1	0				$\boxed{2,50}$	8	0			
		T_i	1	8	1	0							$\boxed{2,50}$	8	0
SPT	4,1,3,2	F_i	3	10	6	1	$\boxed{5}$	10	$\boxed{1}$						
$p_{[1]} \leqslant p_{[2]} \cdots$		L_i	2	8	2	-5				1,75	8	-5			
		T_i	2	8	2	0							3	8	0
LPT	2,3,1,4	F_i	9	4	7	10	7,50	10	4						
$p_{[1]} \leqslant p_{[2]} \cdots$		L_i	8	2	3	4				4,25	8	2			
		T_i	8	2	3	4							4,25	8	2
date de livraison	1,2,3,4	F_i	2	6	9	10	6,75	10	2						
		L_i	1	4	5	4				3,50	$\boxed{5}$	1			
$d_{[1]} \leqslant d_{[2]}$		T_i	1	4	5	4							3,50	$\boxed{5}$	1
temps restant	2,1,3,4	F_i	6	4	9	10	7,25	10	4						
		L_i	5	2	5	4				4	5	2			
$d_{[1]} - p_{[1]} \leqslant d_{[2]} - p_{[2]}$		T_i	5	2	5	4							4	5	2

min : □
max : ○

<u>Commentaires</u> :

1. <u>Quant à la construction du tableau</u>

Prenons le cas de la règle SPT. Par définition, on mettra en première position l'opération dont la durée de traitement est la plus courte, c'est-à-dire celle portant le n° 4, en deuxième position celle dont la durée de traitement est la plus courte lorsque la précédente a été éliminée, c'est-à-dire celle portant le n° 1, etc...

En fonction de cette séquence, on calcule F_i : l'opération 4, traitée en premier lieu sera terminée après une unité de temps (par le théorème 1, on exclut les interruptions et les temps morts) : $F_4 = 1$; l'opération 1, seconde en séquence, sera terminée lorsque l'opération 4 le sera et que sa durée de traitement sera écoulée : $F_1 = 1 + 2$; etc...

L'écart $L_i = C_i - d_i$; comme les dates d'arrivée de toutes les opérations sont identiques (on les suppose arrivées au moment 0), $C_i = F_i$ ce qui donne $L_i = F_i - d_i$.

Le retard de chaque opération s'obtient en calculant $T_i = \max(0, L_i)$; donc si une opération n'a pas de retard, c'est-à-dire si $d_i \geqslant F_i$, on obtient $T_i = 0$.

2. Quant aux résultats

a. en ce qui concerne F, on vérifie que :

- F_{max} est constant : il est en effet égal (dans le cas où il n'y a qu'une seule machine) à la somme des durées de traitement ;
- SPT minimise F_{min} et $\overline{F}$ (Annexe III Section 1) ;
- la règle antithétique à SPT (LPT) maximise ce que SPT minimise ; LPT minimise F_{min} et $\overline{F}$.

b. en ce qui concerne L, on vérifie que:

- SPT minimise $\overline{L}$ (dérivation de la minimisation de $\overline{F}$) ;
- la règle antithétique à SPT (LPT) maximise $\overline{L}$;
- la règle des dates de livraison minimise L_{max} (théorème 2) ;
- la règle du temps restant maximise L_{min} (théorème 3) ;

c. en ce qui concerne T :

- nous avons dit que SPT ne minimisait $\overline{T}$ que si toutes les opérations étaient en retard ; nous avons ici un exemple à contrario, puisque l'opération 4 ne subit aucun retard : on s'aperçoit en effet que SPT ne minimise pas $\overline{T}$ puisque le retard moyen (obtenu par la règle quelconque) est plus petit que celui obtenu par la règle SPT ;
- le théorème 2 se vérifie puisque min T_{max} est obtenu par la règle des dates de livraison ;
- le théorème 3 se vérifie puisque max T_{min} est obtenu par l'application de la règle des temps restants.

3. Quant aux conclusions

Du cas où une seule machine est utilisée, nous retiendrons qu'il soulève déjà un certain nombre de difficultés :

- les critères utilisés étaient simples ; ils sont plus complexes dans la pratique ;
- le critère utilisé était à chaque fois unique ; les méthodes de résolution des problèmes d'ordonnancement permettent difficilement de tenir compte de plusieurs critères liés entre eux ;
- parallèlement, s'il est possible de trouver un ordonnancement optimal en fonction d'un seul critère, cet ordonnancement peut s'avérer incompatible par son impact sur d'autres éléments : un ordonnancement optimal en fonction de la durée d'occupation de l'atelier peut entraîner des retards importants estimés inacceptables ;
- si les hypothèses de base restreignent déjà l'application pratique des résultats théoriques à des cas bien particuliers et si elles ne suffisent pas toujours pour trouver une solution optimale, nous allons voir, dans la section

suivante, que la levée de certaines hypothèses rend le problème encore plus complexe et encore plus partiellement résolu.

Remarque : on trouvera en annexe le cas particulier d'un ordonnancement optimal tant en fonction des dates de livraison que de la durée d'occupation de l'atelier (Annexe III Section 2).

Section 3 : Modèles particuliers d'ordonnancement de n opérations sur une seule machine

Jusqu'à présent, nous avons fait appel à toutes les hypothèses énoncées au chapitre 1 ; le problème étant réduit au cas d'une seule machine, certaines d'entre elles n'étaient pas nécessaires (hypothèses 4,5,8,9). Nous allons voir rapidement, sans entrer dans tous les détails, l'impact sur le modèle général de la suppression de l'une ou de l'autre hypothèse. Notons auparavant que certaines ne seront jamais levées : il s'agit principalement des hypothèses 1 et 3, car elles ont trait soit à la nature même du problème d'ordonnancement, soit à la manière de le présenter.

§ 1. Les durées de traitement ne sont pas connues avec certitude

Dans le cas où nous ne possédons aucune information quant aux durées de traitement, (elles sont donc considérées comme des variables aléatoires), il n'est plus possible d'appliquer les règles utilisant ces durées. On est alors obligé de recourir à l'ordonnancement aléatoire (Random), principalement lorsque le critère d'optimisation est une fonction de la durée d'occupation de la durée (F). Il y a deux manières d'engendrer des ordonnancements aléatoires :

- la durée de traitement est une variable aléatoire ; en prenant les différentes valeurs de cette variable, on applique pour chaque problême la <u>même</u> règle d'ordonnancement : l'estimateur $\hat{p}$ est l'espérance mathématique de la durée de traitement de l'opération placée en ième position ; comme on suppose que les durées de traitement sont des variables aléatoires <u>indépendantes et identiquement distribuées</u> :

$$\hat{p} = E\,(p_{[1]}) = E\,(p_{[2]}) = \ldots\; E\,(p_{[n]}).$$

- on applique des règles <u>différentes</u> (n!) au <u>même</u> ensemble d'opérations : la règle employée est alors la variable et les durées de traitement des opérations sont estimé par une durée moyenne de traitement, notée $\bar{p}$.

<u>THEOREME 4</u> : si le <u>nombre</u> des opérations est connu et que l'on connaît soit la moyenne de leurs durées de traitement ($\bar{p}$), soit leur espérance mathématique ($\hat{p}$), l'espérance mathématique de la durée moyenne d'occupation est égale à :

$$E\,(\bar{F}) = \frac{n+1}{2}\,\bar{p} \quad \text{ou } E\,(\bar{F}) = \frac{n+1}{2}\,\hat{p}$$

<u>Remarque</u> : si nous n'avons pas une connaissance exacte des durées de traitement, mais si nous possédons quelques informations à leur sujet, il est possible, à partir d'estimations concernant chacune des opérations, de trouver des ordonnancements optimaux. Bornons nous simplement à signaler qu'il suffit, pour avoir un ordonnancement optimal, d'avoir des estimateurs qui ne soient pas parfaits mais suffisants que pour obtenir un ordre relatif entre les opérations. Si tous les estimateurs étaient biaisés de la même façon, la qualité de ces estimateurs en pâtirait, mais non l'ordonnancement (l'important est la mise des opérations en séquence, ce qui ne suppose qu'un ordre relatif).

§ 2. Inégalité dans l'importance des opérations

Dans les modèles précédents, (excepté le cas où les dates de livraison étaient différentes), aucune opération n'était prioritaire à priori. C'est le processus d'ordonnancement lui-même qui leur attribuait des priorités, en se basant sur des caractéristiques des opérations (essentiellement les durées de traitement). Des éléments tels que l'importance du client ou de la commande peuvent entraîner l'affectation d'un coefficient d'importance à chaque opération. Si on note l'importance relative de l'opération i, u_i, la durée moyenne et pondérée d'occupation de l'atelier devient : $\overline{F}_u = \frac{1}{n} \sum_{i=1}^{n} u_i F_i$ et le retard moyen pondéré : $\overline{T}_u = \frac{1}{n} \sum_{i=1}^{n} u_i T_i$. ; (pour plus de détails, voir l'annexe III Section 3).

§ 3. L'incorporation des durées de préparation

Nous avons supposé que le temps nécessaire pour préparer une machine à traiter une opération était indépendant de l'opération que cette machine vient de traiter : la durée de préparation pouvait être incorporée dans la durée de traitement. Lorsqu'il y a dépendance entre l'ordonnancement et la durée de préparation, il faut tenir compte d'une matrice S dont l'élément caractéristique s_{ij} exprime le temps nécessaire pour préparer une machine qui vient de traiter l'opération i à traiter l'opération j ;

$s_{[1][i+1]}$ exprimera le temps nécessaire pour préparer une machine, qui vient de traiter l'opération, placée en ième position à traiter celle qui est placée en (i + 1) ème position.

Supposons un atelier de peinture dont la âche consiste à peindre différentes pièces (opérations) dans des couleurs différentes. Il est logique de supposer que le temps nécessaire pour nettoyer l'équipement et pour préparer la nouvelle couleur dépend de la couleur qui vient d'être utilisée (il faut plus de temps pour passer du noir au blanc que du bleu au vert). La matrice S pourrait se présenter de la façon suivante :

	gris	jaune	rouge	bleu
gris	0	1	2	3
jaune	6	0	1	3
rouge	8	6	0	1
bleu	10	8	6	0

On introduit en général une ligne et une colonne supplémentaire pour indiquer le temps nécessaire pour amener une machine de l'état de départ (où elle est inoccupée) à un état tel qu'elle soit prête à exécuter la première opération de la séquence : $s_{[0][1]}$.

Les modifications à apporter au modèle général sont les suivantes :

$$F_{[1]} = s_{[0][1]} + p_{[1]}$$
$$F_{[2]} = F_{[1]} + s_{[1][2]} + p_{[2]}$$
$$\cdots\cdots$$
$$F_{[i]} = F_{[i-1]} + s_{[i-1]} + p_{[i]}$$

Auparavant, F_{max} était constant parce qu'il était égal à la somme des durées de traitement ; ici, F_{max}, qui, par définition est le temps nécessaire pour que la dernière opération de la séquence soit achevée, s'exprime :

$$F_{max} = F_{[n]} = F_{[n-1]} + s_{[n-1][n]} + p_{[n]} = \sum_{i=1}^{n} s_{[i-1][i]} + \sum_{i=1}^{n} p_{[i]}$$

Le dernier terme de l'équation restant constant, F_{max} sera minimisé en minimisant le premier terme.

Ce problème peut être résolu de deux manières :

- <u>la programmation dynamique</u> : son application est possible parce que nous nous trouvons devant un processus Markovien : l'ordonnancement d'une tâche dépend uniquement de la tâche qui vient d'être ordonnancée. Puisqu'il existe un lien entre la tâche ordonnancée et une des tâches à ordonnancer - lien constitué par la durée de préparation nécessaire pour passer d'une tâche à l'autre -, le principe d'optimalité, base de la programmation dynamique, peut s'appliquer ;

- l'algorithme du Branch and Bound : faisant appel à la théorie des graphes, cet algorithme repose sur le principe suivant : le problème original est décomposé en problèmes simples mais fortement liés les uns aux autres ; chacun de ces problèmes aboutit à une solution plus ou moins complète du problème original. Par des routines de calcul, certains de ces problèmes sont résolus, d'autres sont éliminés ou partitionnés, parce qu'ils apparaissent encore trop complexes.

Ces deux méthodes sont très importantes en matière d'ordonnancement (nous aurons encore l'occasion de voir à quels autres types de problèmes elles sont applicables) ; c'est la raison pour laquelle nous leur attribuons une place importante. (Annexe III Section 4).

L'avantage de la programmation dynamique réside dans la connaissance à priori du nombre d'itérations nécessaires pour aboutir à la solution. Elle permet, en outre, de trouver TOUS les ordonnancements optimaux. Les inconvénients résultant de l'utilisation de la programmation dynamique sont les conséquences de ses avantages et sont essentiellement d'ordre pratique :

- le nombre d'éléments à prendre en considération croît exponentiellement avec la dimension du problème : de 33 éléments dans le cas de 4 opérations à ordonnancer, nous en aurons 81 lorsqu'il y en a 5 (dans l'exemple pris en annexe, il y en a 335).

Conway [1] souligne qu'un ordinateur dont la capacité de mémoire s'élève à 32 768 mots ne peut pas résoudre un problème d'ordonnancement comprenant plus de 13 opérations ;

- bien qu'elle soit connue, la durée de traitement d'un tel problème peut s'avérer longue, puisque la programmation dynamique explore systématiquement toutes les solutions possibles.

Du point de vue pratique, le Branch and Bound présente des avantages indéniables par rapport à la programmation dynamique :

- un ordinateur de la capacité ci-dessus peut traiter un problème comportant jusqu'à 40 opérations ;

- bien qu'inconnue, puisque le nombre d'itérations à effectuer n'est pas déterminable à priori, il est possible d'aboutir rapidement à une solution.

Il est cependant à noter que le Branch and Bound ne donne en général qu'un [2] ordonnancement optimal et qu'il est plus difficile à programmer que la programmation dynamique.

Il n'existe pas (à notre connaissance) de méthode qui donne des résultats optimaux lorsque le critère est fonction des retards. Nous avons déjà eu l'occasion, à propos de l'exposition de la méthode de Schild et Fredman (Annexe III Section 3), d'insister sur la difficulté de ce critère, difficulté qui provient de la définition du retard : il n'est pas une différence algébrique entre certaines valeurs, mais le résultat d'une comparaison entre deux valeurs. Comme dans le cas de Schild et Fredman,

---

[1] CONWAY R.W., MAXWELL W.L. et MILLER L.W. : op. cit. p. 65

[2] Il ne s'agit pas à proprement parler d'une limite à la méthode, car il est possible de trouver tous les ordonnancements optimaux, mais le bénéfice de la méthode (rapidité) serait en grande partie perdue.

ces procédures (programmation dynamique et Branch and Bound) ne seraient optimales que si toutes les dates de livraison étaient établies de telle façon qu'aucune opération ne subit de retard, auquel cas min T_{max} serait nul.

§ 4. Arrivées intermittentes des opérations

Dans le modèle général, toutes les opérations étaient disponibles au moment 0 ; en d'autres termes, le premier conflit qui se présentait devant la machine apparaissait au moment 0 et était l'oeuvre de toutes les opérations. On pourrait supposer que les opérations n'arrivent pas au même moment ; nous ne quittons pas le problème statique, car différemment au cas dynamique, nous connaissons avec certitude les dates d'arrivée des opérations ainsi que leurs durées de traitement ; nous pouvons donc établir un ordonnancement à priori, même en l'absence de certaines opérations à exécuter.

Lorsque l'hypothèse selon laquelle $r_i = r_j$ est levée, il faut considérer (différemment au modèle général) la possibilité d'interruption et d'insertion de temps morts, parce que certaines opérations, indisponibles au début de la période d'ordonnancement, peuvent apparaître prioritaires (que cette priorité soit imposée de l'extérieur ou par le processus d'ordonnancement lui-même). Que se passe-t-il lorsqu'une opération est interrompue ? Deux cas extrêmes sont possibles (ils concernent la manière dont les opérations interrompues sont reprises) :

- ou bien, on peut reprendre le traitement de l'opération là où on l'a laissée ; la durée réelle de traitement de cette opération est égale à la durée de traitement donnée.
- ou bien, on ne peut pas reprendre le traitement de l'opération là où on l'a laissée; tout le traitement est à recommencer. La durée réelle de traitement sera alors plus grande ou égale à la durée de

traitement donnée.

Les cas intermédiaires sont très nombreux, mais, en général, on trouve raisonnable de supposer que les durées de traitement d'une opération relèvent du premier type tandis que les durées de préparation relèvent du second. (lorsque l'on reprend une opération, tout le temps nécessaire pour préparer la machine à l'exécuter est à recommencer)

a. <u>lorsqu'on néglige les durées de préparation et lorsque les durées de traitement</u> sont soumises au premier type de reprise, la situation avec arrivées intermittentes est essentiellement la même que celle avec arrivées simultanées.

<u>Exemple</u> :

opération	:	1	2
durée de traitement	:	4	1
moment d'arrivée	:	0	1

L'opération 1 est la seule disponible au départ de la période d'ordonnancement et doit donc être ordonnancée en premier lieu ; si l'opération 1 n'est pas interrompue :

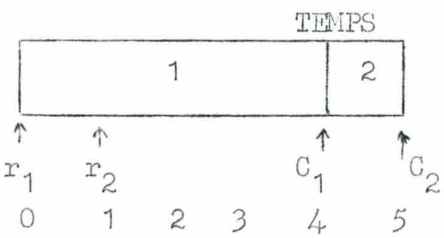
$F_1 = C_1 - r_1 = 4 - 0 = 4$

$F_2 = C_2 - r_2 = 5 - 1 = 4$

$C_{max} = 5$

$F_{max} = 4$

$\overline{F} = 4$



Si l'opération 1 est interrompue lors de l'arrivée de la seconde opération, parce que cette dernière est prioritaire (sa durée de traitement étant plus courte) :

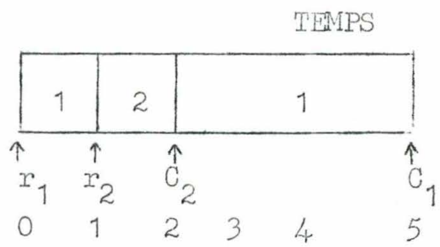
$F_1 = C_1 - r_1 = 5 - 0 = 5$

$F_2 = C_2 - r_2 = 2 - 1 = 1$

$C_{max} = 5$

$F_{max} = 5$

$\overline{F} = 3$



On constate :

- l'intérêt d'interrompre une opération, lorsqu'une autre arrive et qu'elle a une durée de traitement plus courte que celle que était en traitement : cette manière de faire minimise $\overline{F}$.

- que la situation est essentiellement la même que celle où les arrivées sont simultanées :

  . C_{max} est constant : dans le modèle général, nous avons dit que F_{max} était constant ; cela provenait du fait que $F_i = C_i$, ce qui n'est plus le cas ici. En fait, c'est la durée maximale d'occupation de la machine qui est constante (exprimée par le moment où la machine termine la dernière opération de la séquence (C_{max}) ; F_{max} par contre, exprime la durée maximale pendant laquelle une des opérations reste dans l'atelier ; ces deux valeurs sont identiques dans le cas d'arrivées simultanées. En toute logique, nous aurions dû parler de C_{max} dans le modèle général, mais nous avons préféré parler de F_{max} qui a l'avantage d'insister sur les opérations et non sur l'atelier.

  . dans les deux cas, on prend des décisions concernant toutes les opérations :
    - si les arrivées sont simultanées, l'ordonnancement se faisait sur base de la règle SPT (durée de traitement la plus courte)
    - si les arrivées sont intermittentes, l'ordonnancement se fait sur base de la règle de la durée restante (au moment de l'interruption) de traitement la plus courte (SPRT : Shortest remaining processing time).

      Cette discipline minimise (dans le cas d'arrivées intermittentes) toutes les variables qui étaient minimisées par SPT (dans le cas d'arrivées simultanées).

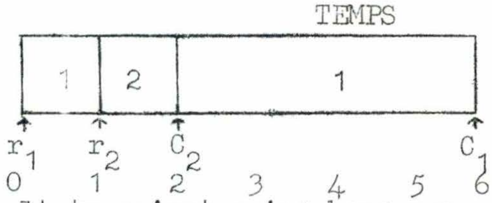
b. lorsqu'on néglige les durées de préparation et que les durées de traitement sont soumises au deuxième type de reprise, il n'existe pas de résultats généraux aussi intéressants. La seule chose que nous puissions dire est que l'interruption d'opérations et l'insertion de temps morts est souvent une meilleure politique que celle qui consisterait à ne pas interrompre et à ne pas insérer de temps morts.

Reprenons notre exemple : - sans interruption, $\overline{F} = 4$;

- avec interruption, $\overline{F} = 3,5$;

en effet, : $F_1 = C_1 - r_1 = 6 - 0 = 6$

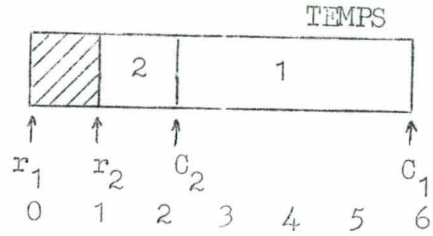
$F_2 = C_2 - r_2 = 2 - 1 = 1$



On pouvait s'attendre à ce que les résultats soient moins bons que dans le premier cas, puisqu'il faut répéter le traitement de la première opération. Notons que les décisions d'ordonnancement se font de la même façon qu'avant (selon SPRT), mais si, dans le premier cas, cette discipline minimise $\overline{F}$ et C_{max}, elle ne minimise ici que $\overline{F}$. Pour minimiser C_{max}, on aurait mieux fait de ne pas interrompre.

Remarque importante : dans le premier cas de reprise, il n'était pas nécessaire d'avoir des informations complètes relatives aux dates d'arrivée et aux durées de traitement de toutes les opérations, alors que, dans le second cas, ces informations sont indispensables ; si nous savons qu'une opération A va arriver avant qu'une opération B ne soit achevée et que A va interrompre celle qui est en cours de traitement (B), il ne faut pas commencer l'opération B , non seulement pour des raisons relatives à l'ordonnancement, mais surtout pour des raisons économiques relatives aux coûts : coûts de main-d'oeuvre attachée à une machine qui traite des opérations inutilement, coûts de matières perdues, coûts d'entretien et d'énergie dépensée inutilement, etc... Les valeurs données aux variables resteraient identiques :

$\overline{F} = 3{,}5$; $F_{max} = C_{max} = 6$



§ 5. Machines parallèles

L'hypothèse 2 (chapitre 1) spécifiait qu'une seule machine PAR TYPE était disponible. Lever cette hypothèse revient à chercher un ordonnancement lorsque plusieurs machines sont susceptibles de traiter la même opération (nous restons dans le cas où il n'y a qu'une opération par tâche). Les données nécessaires pour résoudre ce problème peuvent être inscrites dans un matrice P de dimension n x m, dont l'élément caractéristique p_{ij} est la durée de traitement que nécessiterait l'opération i si elle était entièrement traitée par la machine j.

Machines

$$
\begin{array}{ll}
 & \begin{array}{ccccc} \quad 1 & \quad 2 & \quad 3 & & \quad m \end{array} \\
\text{opérations} \begin{array}{c} 1 \\ 2 \\ 3 \\ \cdots \\ n \end{array} &
\left|\begin{array}{ccccc}
p_{11} & p_{12} & p_{13} & \cdots\cdots & p_{1m} \\
p_{21} & p_{22} & p_{23} & \cdots\cdots & p_{2m} \\
p_{31} & p_{32} & p_{33} & \cdots\cdots & p_{3m} \\
\cdots & \cdots & \cdots & \cdots\cdots & \cdots \\
p_{n1} & p_{n2} & p_{n3} & \cdots\cdots & p_{nm}
\end{array}\right|
\end{array}
$$

Remarques :

- les machines sont identiques si le temps qu'elles mettent pour traiter la même opération est identique ; dans le cas contraire, on parlera de machines <u>parallèles</u> ;

- lorsque certaines machines sont incapables de traiter une opération, l'élément p_{ij} sera noté : infini ;

- selon que l'opération représente un lot de pièces identiques ou peut être divisée en sous-opérations (fractionnement de séries : hyp. 5 et fractionnement d'opérations : hyp. 4), le modèle diffère de celui qui oblige une opération à être exécutée entièrement par une seule et même machine. Le fractionnement se distingue de l'interruption en ce que la reprise de l'opération interrompue aura lieu après la fin de la première partie ; on permet, par contre, que les différentes parties d'une opération fractionnée puissent commencer en même temps (sur différentes machines, selon l'hypothèse 3). Il existe donc dans le premier cas des contraintes de routing qui sont inexistantes dans le second (on évite ces ambiguïtés en parlant d'opérations élémentaires qui, par définition, forment un tout) : on parlera alors d'opérations élémentaires ET d'opérations de série (si une machine doit exécuter le même travail sur 1000 pièces, il serait fastidieux de faire de cette opération, 1000 opérations élémentaires). Dans ce qui suit, nous parlerons uniquement de fractionnement de séries (dans le cas d'opérations élémentaires, la matrice ci-dessus n'est pas suffisante, car par définition, p_{ij} est la durée de traitement de l'opération i lorsqu'elle est traitée entièrement par une machine).

Dans ce qui suit, nous ignorons les durées de préparation-machine et nous envisageons quelques cas particuliers :

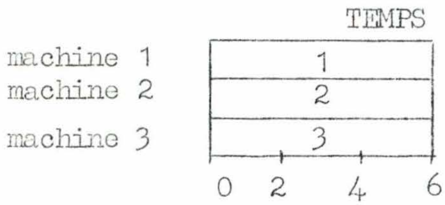
a. il y a autant de machines que d'opérations à traiter ; ces opérations ont toutes la même durée de traitement (les machines sont donc à fortiori identiques) ; pour tout i et pour tout j : $p_{ij} = p$

1. le fractionnement des opérations est interdit :

Le volume de travail s'élève à np et sera également divisé entre les m (=n) machines ; ce cas revient au modèle général ou chaque opération pouvait être envisagée comme une entité indépendante. Toutes les opérations seront terminées en même temps, c'est-à-dire après p unités de temps (on suppose que toutes les opérations sont disponibles en même temps) indépendamment de la manière dont les opérations sont assignées aux diverses machines

$\overline{F} = F_{max} = C_{max} = p$ (p étant constant, il n'y a rien à minimiser)

L'ordonnancement est le suivant (si $p = 6$) :



2. Le fractionnement des opérations est permis

On suppose que l'opération est décomposable en sous-opérations indépendantes (voir ci-dessus). Le meilleur ordonnancement relatif à la minimisation de $\overline{F}$ est établi en décomposant chaque opération en $m = n$ parties et en affectant une partie à chacune des $m = n$ machines. Toutes les opérations, sauf la dernière de la séquence, seront terminées plus tôt que dans le cas où le fractionnement est interdit. La première opération sera terminée au moment p/m, la seconde au moment $2\ p/m$ et la dernière: $mp/\ m = p$. Si on cherche la date moyenne à laquelle une opération sera terminée, on a :

$$\overline{F} = \frac{p}{m}\left(\frac{1}{m} + \frac{2}{m} \dots \frac{m}{m}\right) = \frac{m+1}{2m}\ p$$

En reprenant l'exemple ci-dessus, le diagramme de Gantt est le suivant :

	TEMPS		
machine 1	1	2	3
machine 2	1	2	3
machine 3	1	2	3
	0 — 2	2 — 4	4 — 6

$F_1 = 2$; $F_2 = 4$; $F_3 = 6$; $\bar{F} = 4$

$F_{max} = C_{max} = p$

Ce cas revient à considérer les trois machines traitant en même temps une même opération comme <u>une seule</u> machine ayant une capacité triple à celle d'une des machines de base. Les résultats sont intéressants, mais n'oublions pas que nous avons négligé les durées de préparation ; si nous les avions inclues, nous aurions remarqué que des économies d'échelle substantielles auraient été réalisées en prenant une machine trois fois plus puissante.

b. <u>Les durées de traitement sont différentes d'une opération à l'autre ; le nombre des machines est quelconque ; le fractionnement est possible</u>

1. <u>les machines sont identiques</u> : pour tout i :

   $p_{ij} = p_{i2} = \dots p_{im}$

   Il suffit d'introduire une nouvelle donnée : une "pseudo" durée de traitement : $p'_i = p_i/m$

2. <u>les machines sont parallèles</u> : pour tout i :

   $p_{ij} \neq p_{ik}$ avec j différent de k, la pseudo-durée de traitement :

$$p'_i = \frac{1}{\sum_{j=1}^{m} p_{ij}}$$

Dans les deux cas, chaque machine peut alors être considérée indépendamment l'une de l'autre ; il suffit de reprendre les manières de construire des ordonnancements optimaux (modèle général) en remplaçant, dans toutes les décisions de priorité, basées sur la durée de traitement, p_i par p'_i. Dans le cas particulier où l'on veut minimiser $\overline{F}$, $\overline{C}$, $\overline{L}$, la séquence sera :

$$p'_{[1]} \leqslant p'_{[2]} \leqslant \cdots \leqslant p'_{[n]}$$

c. Les durées de traitement sont différentes d'une opération à l'autre; le nombre de machines est quelconque ; le fractionnement est interdit

Il faut donc que chaque opération soit assignée à une machine particulière qui traitera cette opération dans sa totalité :

1. les machines sont identiques :

pour tout i, $p_{i1} = p_{i2} = \cdots p_{im}$

Cette situation se présente très souvent : guichets de banques, pistes d'atterrissage, caisse de grands magasins, dans une entreprise (lorsque les coûts de préparation-machine sont trop élevés pour envisager qu'une opération puisse être traitée par plusieurs machines). Le problème d'ordonnancement devient beaucoup plus réaliste car on met l'accent sur sa double mission : assigner une machine à chaque opération et déterminer pour chaque machine la séquence des opérations qui lui sont assignées. Soit $j_{[k]}$ l'opération à traiter en kième position sur la machine j et n_j le nombre d'opérations que cette machine doit traiter.
Il en découle :

a) $n = \sum_{j=1}^{m} n_j$

b) $F_{j[1]} = p_{j[1]}$; $F_{j[2]} = p_{j[1]} + p_{j[2]}$; ...

$F_{j[k]} = \sum_{i=1}^{k} p_{j[k]}$

c) $\overline{F} = \frac{1}{n} \sum_{j=1}^{m} \sum_{k=1}^{n_j} (n_j - k + 1)\, p_{j[k]}$

Cette formule, presque identique à celle du théorème de Smith (Annexe III, section 1) a au numérateur une somme de n termes dont chacun est le produit d'un coefficient entier et d'une durée de traitement ; chaque durée de traitement n'apparaît qu'une seule fois (il y a n termes $p_{j\,k}$) et les coefficients apparaissent selon la suite :

$(n_1, n_1 - 1, n_1 - 2 \;\ldots\; 2,1\,; n_2, n_2 - 1 \ldots 2,1\,; n_m, n_m - 1, \ldots 2,1)$

Comme dans le theorème de Smith cité en annexe, on sait que cette somme sera minimisée si une des suites est en ordre décroissant et l'autre en ordre croissant. Or il y a m coefficients 1 ; il suffit de les associer aux m valeurs les plus élevées des durées de traitement ; aux m coefficients 2, on associe les m durées de traitement les plus longues lorque les m premières ont été éliminées etc... Cela signifie que les m opérations dont les durées de traitement sont les plus longues apparaîtront en dernier lieu dans les m séquences-machine et ainsi de suite pour les m opérations qui seront placées en avant-dernière position. On peut donc renuméroter les opérations selon un ordre de durée de traitement non décroissant : $p_1 \leqslant p_2 \leqslant \ldots \leqslant p_n$ et il suffit de numéroter les machines lors de chaque assignation :

opération : 1 2 3 ... m	m + 1 m + 2 m + 3 ... 2m		2m+1..n
machine : 1 2 3 ... m	1 2 3 m		1 ..

Cette discipline, variante de SPT, minimise $\overline{F}$, $\overline{W}$, $\overline{L}$.

<u>Exemple</u> : 7 opérations dont les durées de traitement sont respectivement de 3, 4, 5, 6, 7, 8, 9, et 3 machines. En appliquant les règles qui précèdent, les opérations 5, 6, 7 seront placées en dernière position, les opérations 2, 3, 4 en avant-

dernière position et l'opération 1 en première position.
Dans l'exemple sous revue, quelques-uns des ordonnancements pourraient être :

ORDONNANCEMENTS

machine 1	:	1-4-7	4-7	3-6	1-4-6
machine 2	:	2-5	2-5	1-4-5	2-5
machine 3	:	3-6	1-3-6	2-7	3-7
F_1	=	3	3	3	3
F_2	=	4	4	4	4
F_3	=	5	8	5	5
F_4	=	8	5	8	8
F_5	=	11	11	15	11
F_6	=	13	16	13	16
F_7	=	17	14	13	14
$\overline{F}$	=	61/7	61/7	61/7	61/7
F_{max}	=	17	16	15	16

Dans l'exemple qui précède, nous nous apercevons que $\overline{F}$ est identique pour tous les ordonnancements, mais que F_{max} varie ; il est même possible que min F_{max} ne soit pas parmi tous les ordonnancements établis d'après cette discipline :

Exemple : 4 opérations dont les durées de traitement sont 1, 2, 3, 10.

Des ordonnancements optimaux en fonction de $\overline{F}$ seraient :

machine 1 : 1-3 ou 2-3

machine 2 : 2-4 ou 1-4

Dans ces ordonnancements, F_{max} est 12 et 11. Or l'ordonnancement optimal en fonction de F_{max} serait :

machine 1 : 1-2-3

machine 2 : 4

où F_{max} = 10

2. les machines sont parallèles ($p_{ij} \neq p_{ik}$ pour tout i et $j \neq k$)

Il n'existe aucun résultat général ; cependant, si les durées de traitement varient de telle sorte qu'une machine est toujours plus lente ou plus rapide qu'une autre, on peut minimiser $\overline{F}$ ar la méthode précédente en affectant à chaque machine un coefficient k_j tel que, pour tout i, on vérifie $k_1 p_i = k_2 p_i = k_m p_i$ (dans le cas précédent k = 1)

Remarque : Eastman, Even et Isaacs [1] ont généralisé ce genre de problème en affectant des coefficients d'importance aux opérations ; ils ont seulement abouti à montrer que, dans le cas où il y a m machines identiques et où chaque opération doit être entièrement traitée par une même machine, la limite inférieure de $\overline{F}u(m) \geqslant \frac{m + n}{m(n+1)} \overline{F}u(1)$ où $\overline{F}u(1)$ est le résultat obtenu par $\overline{F}u$ si toutes les opérations sont traitées par une seule machine.

Conclusion du chapitre 2

Nous terminons ici l'étude des problèmes d'ordonnancements relatifs à une seule machine. Nous sommes conscients de ne pas avoir, ne fut-ce que cité tous les développements théoriques qui ont été élaborés ; nous avons pris ceux qui aboutissaient à des résultats optimaux concrets, en mettant l'accent sur certaines méthodes utilisées pour attaquer ces problèmes. Les modèles à une seule machine, s'ils manquent d'application pratique, ont au moins l'avantage de situer le problème et de se rendre compte de sa difficulté de résolution même s'il est fort simplifié ; le critère par lequel on définit un ordonnancement optimal doit rester extrêmement simple ; nous nous sommes rendus compte des difficultés provenant d'un critère basé sur les retards. Les chapitres qui suivent ont pour objet l'étude de modèles

---

1 EASTMAN W.L., EVEN S. et ISAACS I.M. : "Bounds for the Optimal Scheduling of n Jobs on m Processors" ; Management Science vol. 11 n° 2 ; novembre '64

qui envisagent plus d'une machine (il s'agit alors de "vrais" problèmes d'ordonnancement) et les difficultés rencontrées dans le cas d'une seule machine présagent de la pauvreté des méthodes de résolution que nous allons rencontrer.

Chapitre 3 :

L'ordonnancement de n tâches dans un atelier uniforme possédant m types de machines (n/m/F//)

Section 1 : Introduction

§ 1. Routing des tâches et ateliers uniformes

Si les théoriciens ont l'habitude d'appeler les problèmes que nous avons étudiés des problèmes de séquence ("sequencing" et non "scheduling"), c'est parce que ces problèmes n'envisagent qu'un des deux aspects des véritables problèmes d'ordonnancement ; ces deux aspects sont : l'existence de plusieurs machines et l'établissement d'une séquence pour chacune d'elles. Dans les modèles qui précèdent :

- nous avions n tâches comprenant chacune une opération
- les n tâches étaient indépendantes les unes des autres,
- la notion de routing n'avait aucune signification, puisqu'elle indique une dépendance entre plusieurs opérations d'une même tâche.

Cette notion de routing distingue les modèles qui précèdent des modèles qui suivent : l'existence du routing nécessite l'emploi de machines de types différents ; notons que cette nécessité est compatible avec l'hypothèse 2 (cfr. chap. 1) selon laquelle une machine de chaque type est utilisée, mais elle est incompatible avec les modèles relatifs aux machines parallèles où nous avions plusieurs machines du même type.

Deux formes extrêmes de routing peuvent se présenter. Dans le routing STRICT, chaque tâche se compose d'une séquence d'opérations aux relations de précédence immédiate : pour toute tâche i, l'opération notée 1 précède l'opération notée 2 qui elle-même précède l'opération notée 3 etc.. Par l'existence de contraintes de routing, on exclut le chevauchement de plusieurs opérations d'une même tâche ; il faut donc qu'une opération soit terminée avant que la suivante ne puisse commencer. Le routing LÄCHE ne consiste pas dans une indépendance entre toutes les opérations d'une tâche (auquel cas on considérera chaque opération comme une tâche) ; il suffit que certaines opérations ne soient pas liées par des contraintes de précédence ; le meilleur exemple est celui de l'assemblage où plusieurs pièces sont fabriquées indépendamment les unes des autres jusqu'à ce que l'opération d'assemblage ait lieu ; celle-ci ne pourra commencer que lorsque les pièces à assembler sont terminées.

Ce n'est pas cette notion de routing qui constitue l'hypothèse restrictive introduite dans ce chapitre ; cette hypothèse consiste à poser que toutes les tâches suivent la même direction dans leur déplacement à l'intérieur de l'atelier. En d'autres mots, l'atelier uniforme est celui où les machines sont numérotées de telle façon qu'elles expriment la suite des opérations d'une tâche. Si une tâche doit subir un traitement par toutes les machines, l'opération notée i (la notation respecte les contraintes de précédence) sera traitée par la machine notée i.

Supposons 6 machines et 4 tâches :

- la tâche 1 passe successivement sur les machines 2, 3, 4, 5, 6
- la tâche 2 passe successivement sur les machines 1, 2, 3, 4, 5, 6
- la tâche 3 passe successivement sur les machines 1, 2, 6
- la tâche 4 passe successivement sur les machines 2, 3, 4.

Il n'est donc pas nécessaire que toutes les tâches passent par toutes les machines.

On simplifie donc la notation en abandonnant l'indice opération et en numérotant les machines de telle sorte que, pour toute tâche i, l'opération j est réalisée par une machine dont le numéro d'identification est plus élevé que celle qui traite l'opération j', lorsque l'opération j' précède (immédiatement ou non) l'opération j ; un atelier où toutes les tâches suivent une direction unique sera appelé : atelier uniforme ; cette particularité est indiquée dans la notation quadriparamétrique de Conway par la lettre F (en anglo-saxon : flow shop).

§ 2. Les modèles d'ordonnancement dans un atelier uniforme

Dans la section suivante, nous envisagerons les cas particuliers où des méthodes d'analyse combinatoire permettent d'aboutir à des ordonnancements optimaux ; ces cas sont très peu nombreux : il faut notamment

- que la mesure d'évaluation soit régulière (ce qui exclut les retards),
- que le nombre de machines ne dépasse pas trois,

ou que le nombre de tâches ne soit pas supérieur à 2.

L'étude de ces problèmes se résume donc à l'examen des cas $n/2/F/F_{max}$, $n/3/F/F_{max}$, $2/m/F/F_{max}$. Dans la section 3, nous dirons quelques mots du modèle général n/m/F//.

§ 3. Théorème de base

La simplicité des modèles du chapitre précédent résidait dans le fait qu'il suffisait de considérer les ordonnancements de permutation, réduisant le nombre d'ordonnancements à explorer à (n!). Dans le cas le plus général où le mouvement des tâches n'est pas uniforme (voir infra, chapitre 4), ce nombre s'accroit presque indéfiniment, même si on interdit les possibilités d'interruption et d'insertion de temps morts : si g_m est le nombre d'opérations à traiter par la machine m, ce nombre est de $g_1! . g_2! . g_3! . \dots g_m!$. Lorsque le mouvement des tâches est uniforme, la situation est intermédiaire ; il suffit non seulement d'explorer les ordonnancements de permutation, mais ce nombre peut être réduit grâce aux théorèmes 5 et 6 :

<u>THEOREME 5</u> : Lorsqu'on ordonnance un problème n/m/F// et que toutes les opérations sont disponibles au même moment, il suffit, pour <u>minimiser une mesure de performance régulière</u>, de considérer les ordonnancements où la même séquence des tâches se présente sur les deux premières machines.

Le nombre d'ordonnancements à envisager est donc réduit à $(n!)^{m-1}$ [1]

<u>THEOREME 6</u> : Lorsqu'on ordonnance un problème /m/F/F_{max} et que toutes les tâches sont disponibles au même moment, il suffit de considérer les ordonnancements où la même séquence se présente sur les deux premières et sur les deux dernières machines.

Lorsque le critère est min F_{max}, le nombre d'ordonnancements se réduit à $(n!)^{m-2}$ [2]

<u>Exemple</u> : 2/3/F/.. ; les durées de traitement sont :

		machines		
		1	2	3
opérations :	A	4	1	1
	B	1	4	4

Sans l'existence des théorèmes 5 et 6, il y a $(21)^3$ ordonnancements possibles ; nous les grouperons de la manière suivante : parmi les 8 ordonnancements à retenir, les quatre premiers sont ceux dont la séquence sur la première machine est A-B ; les deux premiers parmi ceux-ci ne vérifient pas les théorèmes 5 et 6 ; les deux derniers vérifient le théorème 5, mais seul le dernier vérifie les deux théorèmes. Les quatre derniers sont ceux dont la séquence sur la première machine est B-A ; la subdivision de

---

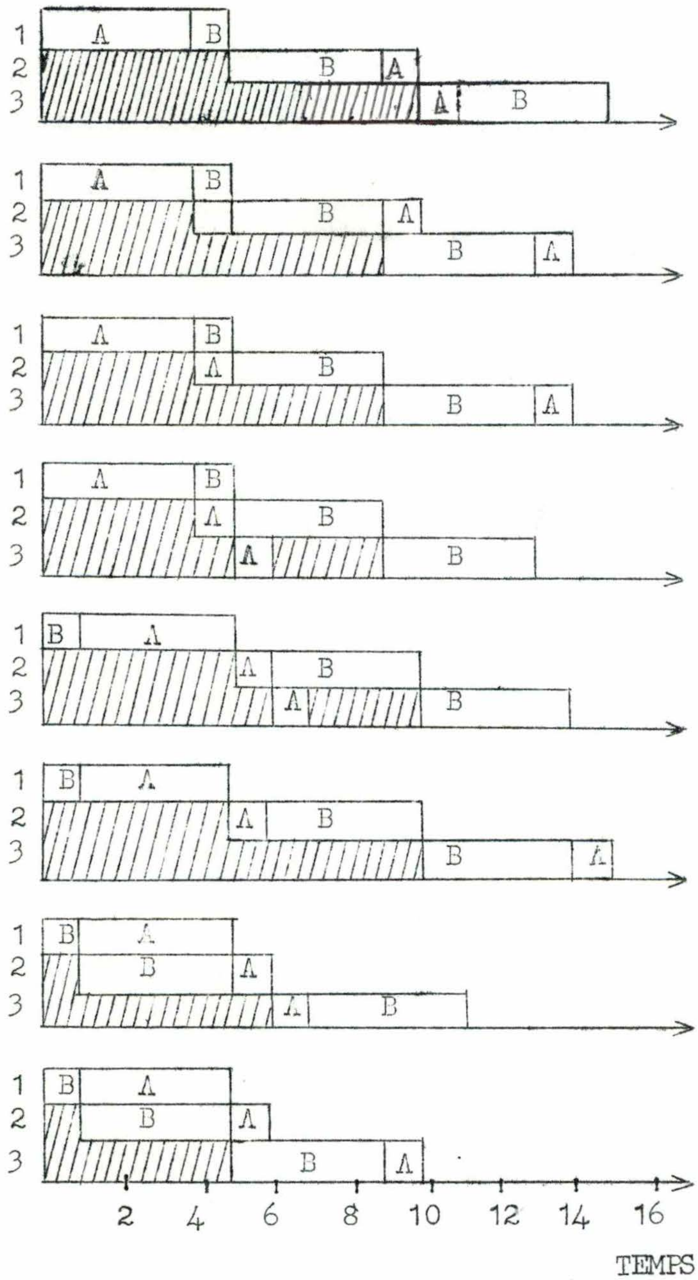
[1] CONWAY R.W., MAXWELL W.L. et MILLER L.W. : op. cit. p. 80

[2] Repris par Conway, ce théorème est dû à ROY B. : "Cheminement et connexité dans les graphes" ; Applications de l'ordonnacement ; METRA, série spéciale n° 1, 1962

ceux-ci est identique à celle faite pour A-B.

On a donc les ordonnancements :

	1	2	3	4	5	6	7	8
machine : 1	AB	AB	AB	AB	BA	BA	BA	BA
2	BA	BA	AB	AB	AB	AB	BA	BA
3	AB	BA	BA	AB	AB	BA	AB	BA



Ordonnancement	F_{max}	F_i	$\bar{F}$
1: machine	15	A:11 B:15	13
2:	14	A:14 B:13	13,5
3:	14	A:14 B:13	13,5
4:	13	A: 6 B:13	9,5
5:	14	A: 7 B:14	10,5
6:	15	A:15 B:14	14,5
7:	11	A: 7 B:11	9
8:	10	A:10 B: 9	9,5

Commentaires :

- le théorème 5 ne dit pas qu'un ordonnancement vérifiant les conditions soit toujours meilleur (en ce qui concerne la minimisation d'une mesure régulière) qu'un ordonnancement qui ne vérifie pas ces conditions ($\overline{F}$ de l'ordonnancement 1 est moins élevé que celui de l'ordonnancement 3). Il en est de même pour min F_{max} : les ordonnancements 3 et 5 sont égaux quant à ce critère, mais le second ne remplit pas les conditions du théorème 5.

- le théorème 5 exprime seulement que, parmi les ordonnancements qui ont la même séquence sur les deux premières machines (3,4 et 7,8), il y en a au moins un qui, vérifiant les conditions du théorème, possède une mesure régulière ($\overline{F}$ ou F_{max}) inférieure ou égale à celle d'un ordonnancement qui ne vérifie pas ces conditions. Pour la séquence AB sur les deux premières machines les valeurs $\overline{F}$ et F_{max} des ordonnancements 3 et 4 sont inférieures ou égales à celles des ordonnancements 1 et 2 ; il en va de même lorsqu'on compare les ordonnancements 7 et 8 avec les ordonnancements 5 et 6.

- le théorème 6 dit seulement que parmi les ordonnancemnts qui ont la même séquence sur les deux premières et sur les deux dernières machines, il y en a au moins un qui, vérifiant les conditions du théorème, a une valeur de F_{max} inférieure ou égale à celle d'un ordonnancement qui ne vérifie pas ces conditions : lorsque la séquence de la première machine est AB, F_{max} de l'ordonnancement 4 est inférieur ou égal à celui des ordonnandements 1, 2, 3. Pour BA, F_{max} de l'ordonnancement 8 est inférieur à celui des ordonnancements 5, 6, 7.

- en ce qui concerne F_{max}, les conditions imposées par les théorèmes 5 et 6 sont logiques. Il faut essayer de commencer les opérations le plus rapidement possible et donc commencer la deuxième opération d'une tâche dès que la première est terminée, sans attendre la fin de la première opération de la deuxième tâche de la séquence. Lorsque la séquence des deux premières machines est identique, la deuxième tâche commencera au même moment que la première opération de la deuxième tâche, alors que si les deux séquences n'étaient pas identiques, il faudrait attendre que les deux premières tâches de la séquence aient achevé leur première opération avant que la deuxième opération de ces tâches ne puissent commencer. L'argumentation est la même en ce qui concerne l'ordonnancement identique sur les deux dernières machines. Sans entrer dans la démonstration du théorème 5, l'argumentation est moins claire lorsqu'on considère d'autres mesures d'évaluation régulières, telle que $\overline{F}$. Notons seulement que commencer plus tôt les opérations à traiter par la deuxième machine réduira la durée moyenne d'occupation de l'atelier (cela dépend en fait des durées de traitement : l'ordonnancement 3 a une valeur de $\overline{F}$ supérieure à celle de l'ordonnancement 1, et pourtant F_{max} de l'ordonnancement 3 est inférieur. Cela provient de ce que la tâche A a, quant à sa durée de traitement, une troisième opération beaucoup plus courte que celle de B).

- Pour trouver la valeur <u>minimum</u> de la mesure régulière, il ne suffit pas de regarder à l'intérieur de chaque groupe (se différenciant par la séquence des tâches sur la première machine), mais tous les groupes ; ceux-ci sont au nombre de n! (dans notre exemple, 2) ; à l'<u>intérieur de chaque groupe</u>, $(n!^{m-2})$ ordonnancements vérifient les conditions du théorème 5 et $(n!^{m-3})$ celles du théorème 6.

En effet, <u>pour le théorème 5</u> :

. la séquence de la première machine est donnée (puisque l'on ne s'intéresse qu'à un groupe)

. la séquence de la seconde est identique à la première.

Il reste donc à examiner les séquences des (m-2) machines restantes.

Pour le <u>théorème 6</u> :

. la séquence de la première machine est donnée

. la seconde est identique à la première

. la séquence de la dernière est identique à celle de l'avant-dernière.

Il reste donc à envisager les séquences des (m-3) autres machines. Puisque le nombre de séquence sur la première machine s'élève à n!, il y a $(n!)(n!)^{m-2} = (n!)^{m-1}$ ordonnancements à examiner lorsqu'il faut trouver celui qui minimise une mesure régulière et $(n!)^{m-3} = (n!)^{m-2}$ ordonnancements lorsqu'il faut trouver celui qui minimise F_{max}.

Section 2 : Ordonnancements optimaux dans le cas d'un atelier uniforme

§ 1. $n/2/F/F_{max}$

Nous gardons les hypothèses posées dans le chapitre 1. L'indice n de la notation quadriparamétrique indique que n tâches sont disponibles au même moment, tandis que l'indice m=2 indique qu'il y a deux types de machines, ce qui suppose que chaque tâche se compose d'un maximum de deux opérations ; nous sommes en effet dans un atelier uniforme : s'il y avait plus de deux opérations par tâche, cela supposerait que plusieurs opérations d'une même tâche doivent être traitées plus d'une fois par une même machine ; dans ce cas, le mouvement des tâches d'une machine à l'autre ne serait certainement plus uniforme. Des considérations qui précèdent, on retiendra que :

- la séquence des deux machines doit être identique (théorème 6) ;
- puisque toutes les tâches sont disponibles au même moment et que l'on ne peut allonger les ordonnancements si ce n'est pour des raisons de routing, il est clair que ni les interruptions, ni les temps morts ne sont à envisager.

Ce problème a reçu une solution grâce à l'algorithme de Johnson [1] dont on trouvera le principe, l'interprétation géométrique et un exemple

---

[1] JOHNSON S.M. : op. cit.

dans l'annexe III section 5 § 1. On y trouvera aussi une généralisation de cet algorithme due à Mitten [1]. Il a levé l'hypothèse selon laquelle les tâches doivent commencer le plus rapidement possible, par l'introduction de délais (hyp. 9 du premier chapitre).

§ 2. $n/2/F/\bar{F}$

Hormis certains cas bien particuliers, il est impossible de trouver un ordonnancement optimal lorsqu'on applique le critère $\bar{F}$ à deux machines. Si nous avions déjà rencontré des difficultés en ce qui concerne les retards, nous rencontrons ici les premières, lorsque le critère a trait à la durée d'occupation de l'atelier par une tâche. On trouvera dans l'annexe III section 5 § 2 de plus amples informations concernant ce problème.

§ 3. $n/3/F/F_{max}$

De nombreux auteurs se sont préoccupés de ce problème, car sa résolution ouvrait la porte à celle du modèle général ; énumérons quelques-unes des idées importantes à retenir :

- le théorème 6 (section précédente) est d'application : au lieu de devoir envisager $(n!)^3$ ordonnancements, l'examen de $(n!)$ d'entre eux suffit puisque les deux premières et les deux dernières machines doivent avoir la même séquence ; la séquence sera donc identique sur les trois machines.

- Johnson a généralisé l'algorithme qu'il avait élaboré pour le problème $n/2/F/F_{max}$ en démontrant qu'il procurait un ordonnancement optimal dans le cas de trois machines si et seulement si on vérifie : $\min A_i \geqslant \max B_i$ ou si $\min C_i \geqslant \max B_i$ [2].

  L'ordonnancement optimal est obtenu en faisant précéder la tâche i

---

[1] MITTEN L.G. : "Sequencing n Jobs on 2 Machines with Arbitrary Time Lags" ; Management Science , vol. 5 n° 3 ; avril 1959.

[2] A_i, B_i, C_i representent la durée de traitement de la tâche i sur la machine A, B et C.

par la tâche j si $\min(A_i + B_i , C_j + B_j) < \min (A_j + B_j , C_i + B_i)$. Il suffit donc de remplacer dans l'algorithme (cfr Annexe) la colonne A_i par $A_i + B_i$ et la colonne B_i par $B_i + C_i$.

- Ignall et Schrage [1] ont appliqué l'algorithme du Branch and Bound. Le principe de cette technique est semblable à celui que nous avons envisagé antérieurement [2] mais les modalités opératoires sont différentes car nous n'avons plus affaire à des durées de préparation-machine reliant deux tâches entre elles ; les données ne peuvent plus être inclues dans une matrice qui serait à minimiser. On trouvera dans l'annexe IV une application de ce problème ainsi que la résolution d'un problème $n/3/F/F_{max}$ par l'algorithme de Johnson.
- Jackson [3] a considéré un cas fort particulier en supposant que toutes les tâches subissent leur première opération sur une même machine ainsi que la dernière, tandis que la seconde est réalisée par autant de machines qu'il y a de tâches. Au total, l'atelier comporte donc n+2 machines. L'algorithme de Johnson est alors applicable selon les mêmes modalités que ci-dessus. Cette particularité est intéressante dans les circonstances suivantes :

  - le routing des tâches est tel que chacune des deuxièmes opérations doit être traitée par une machine différente ;
  - B_i, durée de traitement de la deuxième opération d'une tâche i, traitée sur la machine B est un délai arbitraire qui doit être observé entre deux opérations d'une même tâche : inspection, refroidissement, séchage ... (l'atelier se compose de deux machines) ;

---

[1] IGNALL E. et SCHRAGE L. : "Applications of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems" ; Journal of the O.R.S.A. vol. 13, n° 3 ; mai 1965 ; voir aussi LOMNICKI Z.A. : "A Branch and Bound Algorithm for the Exact Solution of the Three-Machine Scheduling Problem" ; Operational Research Quarterly, vol. 16, 1965.

[2] Chapitre 2, section 3, § 3 et l'Annexe III Section 4

[3] JACKSON J.R. : "An Extension of Johnson's Result on Job-Lot Scheduling" ; Nav. Res. Log. Quat. vol. 3 N° 3 ; septembre 1956

. B_i pourrait prendre des valeurs négatives, exprimant la situation où l'hypothèse 5 (chap. 1) est levée dans un atelier à deux machines ; en permettant le chevauchement, certaines parties d'une tâche pourraient commencer leur seconde opération alors que la première n'est pas terminée dans sa totalité.

§ 4. $n/m/F/F_{max}$

Lorsque l'atelier se compose d'un nombre important de machines (plus de trois), le problème le plus simple est celui où il n'y a que deux tâches à réaliser. Akers [1] donne une représentation géométrique de ce problème, légèrement différente de celle vue pour le problème $n/2/F/F_{max}$ (cfr. § 1 et Annexe III section 5). Sur les deux axes, nous représentons, dans l'ordre les durées de traitement des tâches (dans un espace vectoriel à n dimensions, on pourrait représenter n tâches). Le graphique est du type tâche, alors que celui de Johnson est du type machine.
La représentation des tâches est une adaptation du diagramme-tâche (où les contraintes d'ordonnancement ne sont pas respectées). Les blocs ne représentent plus le traitement d'une tâche par une seule machine, mais le traitement de deux tâches par une même machine. Il s'agit donc de blocs-machines. F_{max} est une ligne V qui :

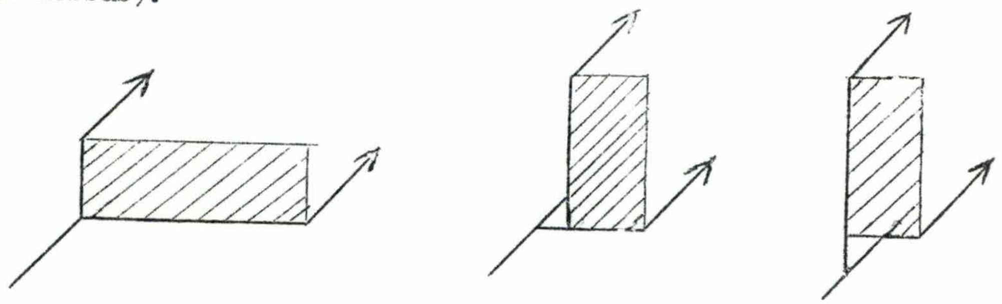
- part de (0,0) et aboutit en ($\sum_{j=1}^{m} p_{1j}, \sum_{j=1}^{m} p_{2j}$) ;
- est composée de . segments horizontaux lorsque la tâche 1 est en traitement,
  . de segments verticaux lorsque la tâche 2 est en traitement,
  . de segments à 45° lorsque les deux tâches sont en traitement ;

---

[1] AKERS, Jr. S.B. : "A graphical Approach to Production Scheduling Problems" Journal of the O.R.S.A., vol. 4 n° 2 ; avril 1956

- ne peut entrer dans un bloc-opération ; le cas inverse supposerait qu'une machine travaille sur deux tâches en même temps.

2^m lignes V peuvent être construites sur ce graphique ; elles forment un graphe appelé "arbre" : en partant de (0,0), on explore deux branches chaque fois qu'une ligne d'ordonnancement rencontre un bloc-machine (en-dessous et au-dessus).



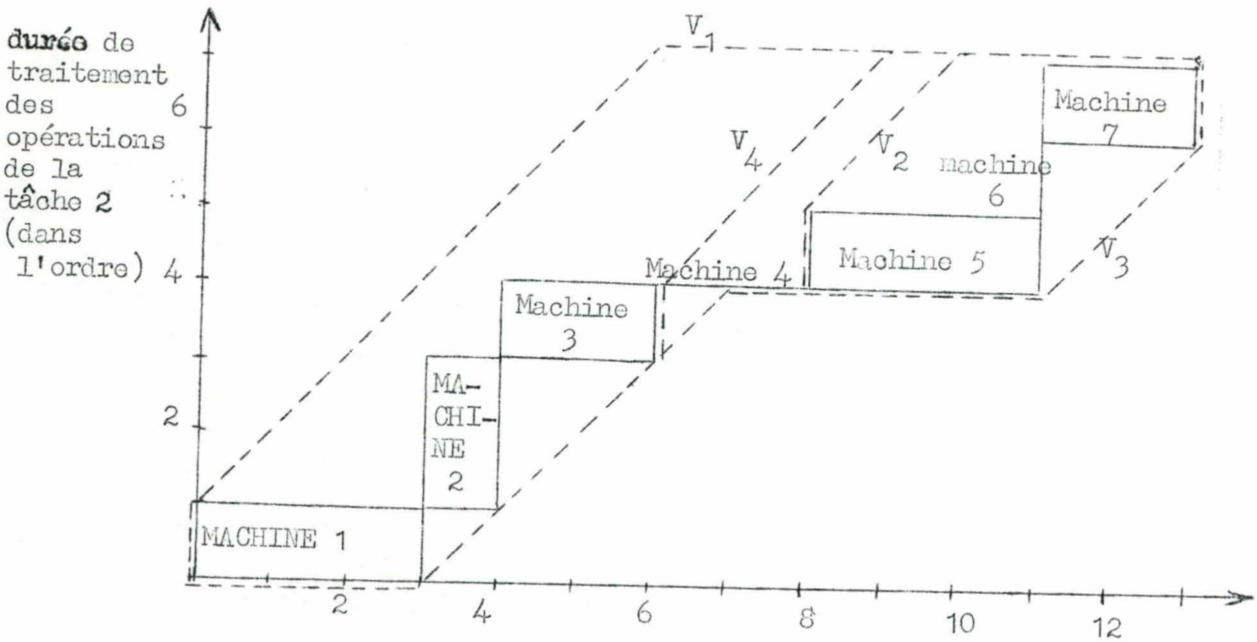
La valeur de F_{max} est obtenue en ajoutant aux durées de traitement de la tâche 1 la somme des segments verticaux de la ligne V (elle est également obtenue en ajoutant aux durées de traitement de la deuxième tâche la somme des segments horizontaux de la ligne V correspondante).

Comme les durées de traitement sont des constantes, l'ordonnancement optimal est donné par la ligne V dont la longueur des segments verticaux ou horizontaux est la plus faible. Si la ligne V optimale passe au-dessus d'un bloc-machine j, la tâche 1 précèdera la tâche 2 sur la machine j, tandis qu'elle suivra la tâche 2 si la ligne V passe en-dessous du bloc-machine j.

Remarquons que la construction de ce graphique est plus simple que celui du problème à deux machines, car, contrairement à celui-ci, il n'est que très rarement nécessaire de construire les 2^n lignes, beaucoup d'entre elles ne restant que potentielles ; en plus, certaines se rejoignent pour concourrir ensemble vers le même sommet. On notera également que l'on peut tenir compte du théorème 6 en ne construisant qu'une branche pour le second bloc et une pour le dernier (lorsqu'un branchement est nécessaire). Dans l'exemple qui suit, il n'y a pas plus de 4 lignes à construire :

Exemple : $2/7/F/F_{max}$

	Machines						
	1	2	3	4	5	6	7
Tâche : 1	3	1	2	2	3	0	2
2	1	2	1	0	1	1	1



La valeur de F_{max} est de 14 pour le quatre ordonnancements. (somme des segments horizontaux plus les durées de traitement de la deuxième tâche)

Les 4 ordonnancements sont :

- avec V_1 : sur toutes les machines : tâche 2 et puis la tâche 1
- avec V_2 : pour les machines 1 à 4, la séquence est 1 - 2 ; pour les machines 5 à 7, elle est 2 - 1 ;
- avec V_3 : sur toutes les machines : tâche 1 et puis la tâche 2

- avec V_4 : pour les machines 1 à 3, la séquence 1 -2 ; pour les machines 4 à 7, elle est 2 - 1.

Section 3 : Le modèle général d'ordonnancement dans un atelier uniforme (n/m/F/...)

Nous ne nous attarderons pas sur ce modèle car il s agit d'un cas particulier d'un modèle plus général (chapitre 4). Il nous faut pourtant citer deux études qui ont trait aux ateliers uniformes :

- Dudek et Teuton [1] ont développé un algorithme ; élaboré au départ pour résoudre un problème à trois machines, ils ont prétendu pouvoir l'appliquer à des ateliers plus importants. Karush [2] a réfuté la valeur de cet algorithme.
- Heller [3] engendra de façon aléatoire 3 000 ordonnancements dans des problèmes $100/10/F/F_{max}$ où les durées de traitement étaient distribuées uniformément entre 1 et 9. Il conclut que la distribution des F_{max} est asymptotiquement normale. Si cette conclusion peut servir à opérer un choix parmi les règles de décisions théoriques, deux obstacles s'opposent à une procédure opératoire :
  - la normalité de cette distribution n'est intéressante que lorsque le critère est la moyenne des F_{max} et non lorsqu'il se base sur les extrêmes de cette distribution ;
  - il est inconcevable qu'il n'y ait pas de classes d'ordonnancement qui soient meilleures que d'autres (cfr. chap. 4) ; il ne s'agit donc par d'engendrer des ordonnancements au hasard : un

---

[1] DUDEK R.A. et TEUTON O.F. : "Development of M - Stage decision Rule for Scheduling n Jobs Through m Machines" ; Journal of the O.R.S.A., vol. 12 n° 3 ; mai 1964.

[2] KARUSH W.: "A Counterexample to a Proposed Algorithm for Optimal Sequencing of Jobs" : Journal of the O.R.S.A., vol. 13 n° 2 ; mars 1965

[3] HELLER J. "Some Numerical Experiments for an M xJ Flow Shop and its Decision Theoretical Aspects" ; Journal of the O.R.S.A. vol. 8 n°2, mars 1960

ordonnancement qui placerait une tâche i en dixième position sur une machine et en première position sur la machine suivante ne peut être optimal ; il semble donc plus intéressant (mais pas nécessairement optimal) de prendre des ordonnancements où la séquence de toutes les machines est identique. Nugent [1] a développé cette théorie et s'est aperçu que, s'il s'agissait là d'un bon point de départ pour échantillonner des ordonnancements, des valeurs plus faibles de F_{max} étaient obtenues par des ordonnancements qui ne respectent pas strictement cet ordre, mais présentent des permutations entre tâches voisines (pour des machines consécutives).

Chapitre 4 :

Le problème d'ordonnancement général (n/m/g//)

Section 1 : Introduction

Contrairement aux modèles du chapitre 3, l'atelier est quelconque ; cela signifie qu'il n'est plus possible de numéroter les machines de telle façon qu'elles expriment la séquence des opérations d'une tâche. La direction du mouvement des tâches n'est donc plus identique pour chacune d'entre elles : certaines tâches peuvent nécessiter le passage de plusieurs opérations sur la même machine (les opérations ne peuvent pas être consécutives).

Exemple :

La tâche 1 passe successivement par les machines 6,5,4,3,2,1
" " 2 " " " " " 1,2,3,4,5,6

---

[1] NUGENT C.E. : "On Sampling Approaches to the Solution of the n by m Static Sequencing Problem" Ph. D. Thesis, Cornell University, september 1964.

la tâche 3 passe successivement par les machines 1,2,3,2,3,4,6

Dans ce modèle, il est indispensable que chaque opération possède trois identificateurs i, j, k où i est le numéro d'identification de la tâche, j le numéro de séquence de l'opération et k le numéro d'identification de la machine qui doit traiter cette opération. Nous renvoyons au premier chapitre de cette partie pour la présentation du problème par le diagramme de Gantt.

Deux problèmes sont résolus jusqu'à présent et ils sont évoquées dans la section 2 ; les sections suivantes examinent d'autres manières d'envisager le modèle général mais, pour une raison ou une autre, elles ne fournissent pas de résultats optimaux.

Section 2 : Les problèmes généraux résolus : $n/2/G/F_{max}$ et $2/m/G/F_{max}$

§ 1. $n/2/G/F_{max}$

Jackson [1] a généralisé l'algorithme de Johnson [2] qui supposait que toutes les tâches subissaient leur première opération sur la machine notée A et la seconde sur la machine notée B. Jackson divise les n tâches en quatre sous-ensembles :

$\{A\}$: ensemble des tâches n'ayant qu'une opération à réaliser et celle-ci est à traiter par la machine A

$\{B\}$: ensemble des tâches n'ayant qu'une opération à réaliser et celle-ci est à traiter par la machine B

$\{AB\}$: ensemble des tâches se composant de deux opérations, dont la première est à traiter par la machine A et la seconde par la machine B

[1] JACKSON J.R. : "An Extension of Johnson's Result on Job-Lot Scheduling", Nav. Res. Log. Quart. vol. 3 n° 3, september 1956

[2] JOHNSON S.M. : op. cit.

$\{BA\}$: ensemble des tâches se composant de deux opérations dont la première est à traiter par la machine B et la seconde par la machine A.

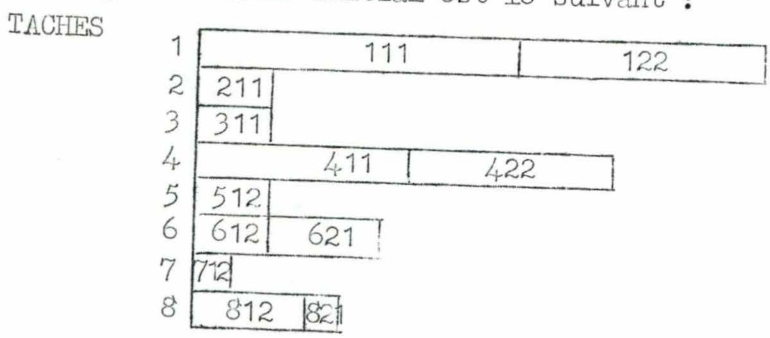
L'algorithme est le suivant :

i) par la procédure de Johnson [1], déterminer la séquence des tâches appartenant aux ensembles $\{AB\}$ et $\{BA\}$;

ii) la séquence des tâches appartenant aux ensembles $\{A\}$ et $\{B\}$ est quelconque ;

iii) l'ordonnancement optimal est celui où, sur la machine
A, l'ensemble $\{AB\}$ est placé avant l'ensemble $\{A\}$ lui-même placé avant $\{BA\}$
B, l'ensemble $\{BA\}$ est placé avant l'ensemble $\{B\}$ lui-même placé avant $\{AB\}$

Au lieu de démontrer l'optimalité de cette procédure (elle se base sur le temps morts se présentant entre les opérations), nous préférons la montrer par un exemple $8/2/G/F_{max}$.
Le diagramme-tâche initial est le suivant :



$\{A\}$ se compose des tâches 2 et 3, $\{B\}$ des tâches 5 et 7, $\{AB\}$ des tâches 1 et 4 et l'ensemble $\{BA\}$ des tâches 6 et 8. L'algorithme de Johnson met les tâches $\{AB\}$ dans l'ordre 4-1 et celles de l'ensemble $\{BA\}$ dans l'ordre

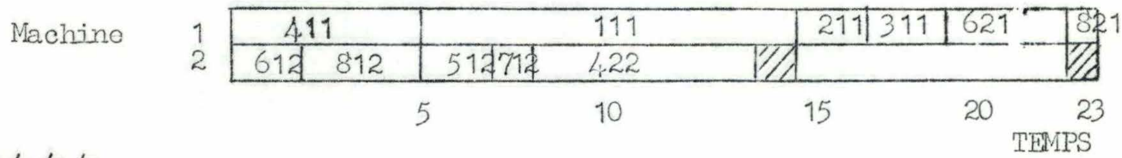
---

[1] Annexe III, section 5, § 1

6-8. Il y a quatre ordonnancements optimaux selon l'ordre des tâches appartenant à {A} et à {B}.

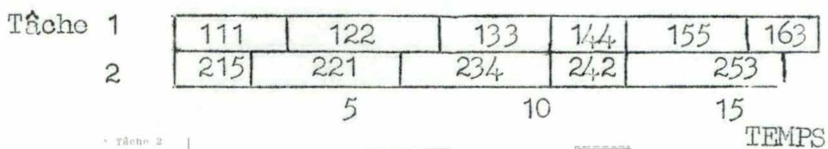
La séquence sur la machine A est : 4-1-2-3-6-8 (ou 4-1-3-2-6-8)

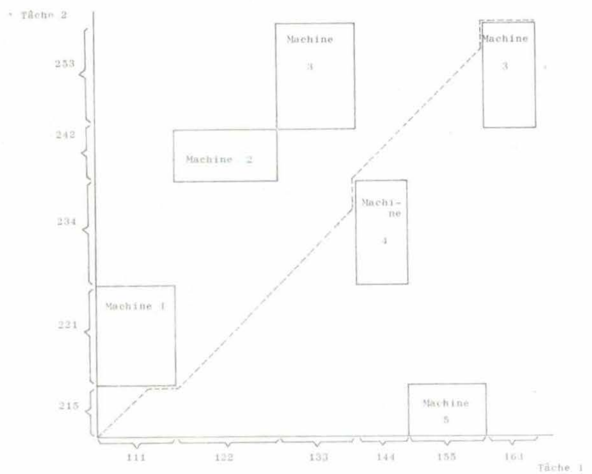
B est : 6-8-5-7-4-1 (ou 6-8-7-5-4-1)



§ 2. $2/m/G/F_{max}$

La résolution du problème $2/m/G/F_{max}$ est identique à celle du problème $2/m/F/F_{max}$ [1]. La seule différence réside dans la place occupée sur le graphique par les bolcs-machines : ces blocs ne sont plus nécessairement attenant les uns aux autres. Cette particularité provenait de ce que l'opération i d'une tâche était traitée par la machine notée i. Comme dans notre problème de nombreux blocs se situeront en dehors du parcours des lignes V, celles-ci seront moins nombreuses.





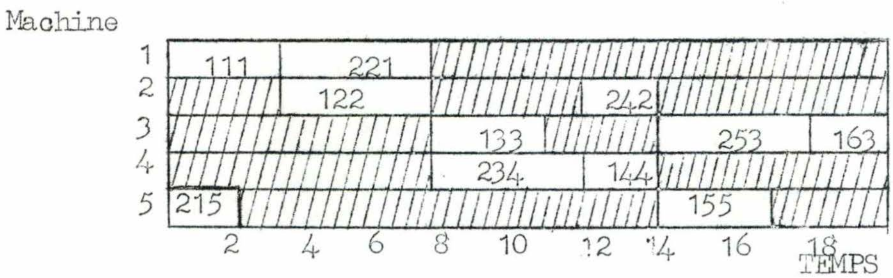
---

[1] Chapitre 3, Section 2 § 4

(Pour être complet, nous aurions dû construire d'autres lignes V notamment celles qui contournent les blocs-machines 1 et 4). La valeur de F_{max} est donnée par la somme des durées de traitement de la tâche 1 et de la longueur des segments verticaux : 17 + 2 = 19. La situation des blocs par rapport à la ligne V nous donne l'ordonnancement suivant :

sur la machine 1 :	111	précède	221
" " " 2 :	122	"	242
" " " 3 :	133	"	253
	253	"	163
" " " 4 :	234	"	144
" " " 5 :	215	"	155

Grâce à ces relations de précédence, le diagramme-machine se construit aisément. Il suffit de respecter les contraintes de routing et de caler les opérations le plus à gauche possible :



Section 3 : Programmation linéaire en nombres entiers

Toutes les méthodes que nous avons rapidement survolées étaient du type "combinatoire" ; elles fournissent toutes des algorithmes simples permettant d'aboutir facilement et rapidement à des ordonnancements optimaux. Il est important de rappeler ici que l'ordonnancement est un problème essentiellement pratique ; pour cette raison, il est inutile de développer des théories complexes si elle n'aboutissent pas à une formulation d'algorithmes simples. L'analyse combinatoire possède cette simplicité, mais manque de généralité. La simulation possède cette généralité, mais ne donne qu'accidentellement des ré-

sultats optimaux. La programmation occupe une place intermédiaire : elle est capable de traiter des problèmes complexes, aboutit rarement à des résultats optimaux, mais est loin d'être simple.
Contrairement à la programmation dynamique, la programmation linéaire n'explore pas tout l'espace des solutions (cfr. méthode du Simplexe). Il en est de même pour les méthodes "combinatoires". On pourrait donc inclure la programmation linéaire parmi celles-ci ; nous ne l'avons pas fait par souci de classification. Les difficultés résident essentiellement dans la formulation du problème et le coût de résolution, en supposant que les mémoires des ordinateurs soient suffisantes, ce qui est loin d'être le cas. Est-il préférable d'obtenir un ordonnancement optimal par une méthode lente et coûteuse ou un ordonnancement suffisant par une méthode simple et peu coûteuse ? La plupart des chercheurs et des industriels opteront pour la seconde solution.

Bowman [1], Wagner [2], Story [3], et Manne [4] se sont penchés sur l'utilisation de la programmation linéaire dans les problèmes d'ordonnancements. Ils supposent que chaque tâche requiert le service de chacune des m machines et une seule fois : le problème est donc moins général qu'il n'y paraît. Sans entrer dans les détails, la formulation de Manne est la suivante :

p_{ik} = durée de traitement de la tâche i sur la machine k (l'indice opération n'est plus nécessaire car, à cause de l'hypothèse qui précède
r_{ijk} = 1 si la jième opération de la tâche i exige un traitement par la machine k ; sinon, $r_{ijk} = 0$; on exprime ainsi le routing des tâches ;
T_{ik} = date de départ de la tâche i sur la machine k.

---

[1] BOWMAN E.H. : "The Schedule - Sequencing Problem " ; Journal of the O.R. S.A. vol. 7 n° 5, septembre 1959
[2] WAGNER H.M. : "An Integer Linear Programming Model for Machine Scheduling"; Nav. Res. Log. Quart. vol. 6 n° 2, juin 1959
[3] STORY A.E. et WAGNER H.M. : Dans MUTH, Y.F. et THOMPSON G.L. op. cit. chap. 14
[4] MANNE A.S. : "On the Job-Shop Scheduling Problem" Journal of the O.R.S.A. vol. 8 n° 2, mars 1960

- Les contraintes d'ordonnancement sont :

$T_{ik} - T_{jk} \quad p_{jk} \geqslant$ ou $T_{jk} - T_{ik} \geqslant p_{ik}$

Comme la programmation ne permet pas ce genre d'alternative, nous sommes obligés d'introduire des variables entières supplémentaires :

Y_{ijk} = 1 si i précède j directement on non
= 0 autrement

Les deux alternatives ci-dessus sont remplacées par les deux contraintes suivantes :

$$(M + p_{jk})Y_{ijk} + (T_{ik} - T_{jk}) \geqslant p_{jk}$$
$$(M + p_{ik})\ (1 - Y_{ijk}) + (T_{jk} - T_{ik}) \geqslant p_{ik}$$

M est une constante dont la valeur est suffisamment grande pour qu'une de ces contraintes soit restrictive lorsque Y_{ijk} = 0 ou 1. Dans le cas où i précède j, Y_{ijk} = 1, la première contrainte reprend une des alternatives ci-dessus, tandis que la seconde sera réalisée automatiquement.

- Les contraintes de routing

Les contraintes de précédence entre les opérations sont prises en considération en notant que $\sum_{k=1}^{m} r_{ijk}\ T_{ik}$ est le moment de départ de la jième opération de la tâche i (dans cette somme, il n'y aura qu'un terme positif). Pour toutes les opérations d'une tâche, sauf la dernière, il faut que :

$$\sum_{k=1}^{m} r_{ijk}\ (T_{ik} - p_{ik}) \leqslant \sum_{k=1}^{m} r_{i(j+1)k}\ T_{ik}$$

- Les contraintes de date de livraison (éventuellement)

Pour tout i : $\sum_{k=1}^{m} r_{imk} (T_{ik} + p_{ik}) \leqslant d_i$

On exprime ainsi que la date de départ de la dernière opération augmentée de sa durée de traitement doit être inférieure ou égale à la date de livraison.

- La fonction objectif

Si le critère d'optimalité est :

- min $\overline{F}$, il faut minimiser la date de départ de la dernière opération de chaque tâche : min $\sum_{i=1}^{n} \sum_{k=1}^{m} r_{imk} T_{ik}$
- min F_{max}, il faut minimiser une variable X où

$$X = \max \left(\sum_{k=1}^{m} r_{imk} (T_{ik} + p_{ik}) \right)$$

- min $\overline{L}$, il faut minimiser

$$\left(\sum_{i=1}^{n} \sum_{k=1}^{m} r_{imk} (T_{ik} + p_{ik} - d_i) \right)$$

- min $\overline{T}$, il faut minimiser la même somme que pour $\overline{L}$, mais il faut en plus introduire des contraintes qui éliment les retards négatifs :

$$r_{imk} T_{ik} + p_{ik} - d_i \geqslant 0$$

Dans le cas où l'on cherche à minimiser $\overline{T}$ ou $\overline{L}$, il est clair qu' il ne faut pas introduire les contraintes de date de livraison, car elle sont en opposition avec la fonction objectif. A ce propos, il est à signaler que, si aucune solution n'est obtenue, cela provient en premier lieu de ces contraintes de date de livraison : il faut de toute façon que pour tout i, $d_i \geqslant \sum_{k=1}^{m} p_{ik}$

Si on utilise la formulation de Manne, le système reste extrêmement complexe même lorsque les problèmes sont de petite dimension : avec 4 machines et 10 tâches, nous avons 220 variables et 390 contraintes (en supposant que l'on ignore les variables d'écart et les contraintes de non-négativité).

Variables :	Nombre :
$T_{ik} \geqslant 0$	mn
$Y_{ijk} = 0$ ou 1	$\frac{m(n!)}{2!(n-2)!} = \frac{mn(n-1)}{2}$

Equations :

1) $\sum_{k=1}^{m} r_{ijk}(T_{ik} + p_{ik}) \leqslant \sum_{k=1}^{m} r_{i(j+1)k} T_{ik}$ $\qquad (n-1)n$

2) $(M + p_{jk})Y_{ijk} + (T_{ik} - T_{jk}) \geqslant p_{jk}$ $\qquad \frac{mn(n-1)}{2}$

3) $(M + p_{ik})(1 - Y_{ijk}) + (T_{jk} - T_{ik}) \geqslant p_{ik}$ $\qquad \frac{mn(n-1)}{2}$

Manne conclut en disant [p. 191] : "In weighing, the benefits from exact optimization of large systems via integer programming, the investigator ought not to permit himself to ignore the likelihood that Monte Carlo methods for approximate optimization will entail lower computing costs..." Si le problème reste complexe, c'est parce que le système de contraintes n'est pas suffisant pour éliminer certaines solutions ; les contraintes ne font qu'exprimer la nature du problème : minimiser un critère en respectant les contraintes de routing et d'ordonnancement. "... we have not yet found an integer programming method that can be relied upon to solve most machine sequencing problems rapidly. We believe that future study must concentrate on deriving methods which more fully take into account the special structure of machine sequencing problems, which... are capable of finding quickly an integer solution when the optimal objective value has indeed been obtained ;... such method could reduce the number of iterations necessary to reach the optimal objective function value." [1] Il semble donc (et Wagner n'est pas le seul de cet avis) que la résolution du problème général est à trouver dans une amélioration de la technique de programmation, grâce notamment à un système de contraintes secondaires. En attendant, il reste une dernière manière d'approcher la solution optimale d'un problème d'ordonnancement :

Section 4 : La simulation

Sous ce terme, est groupé un ensemble de travaux (Akers et Friedman [2] Heller [3], Giffler et Thompson [4], Rowe [5]) ; sans nous lancer dans les détails de procédure, nous voudrions insister sur deux idées essentielles :

- parmi les ordonnancements possibles (un ordonnancement est "possible" lorsqu'il vérifie les contraintes de routing et d'ordonnancement), il existe des sous-ensembles d'ordonnancement qui sont meilleurs que d'autres ; on peut donc réduire le nombre d'ordonnancements à explorer ;
- l'emploi de règles de décision donne des résultats satisfaisants.

§ 1. Caractéristiques de l'ordonnancement optimal

a. Les ordonnancements semi-actifs, actifs et non-retardés

Nous avons déjà eu l'occasion de signaler que, s'il était virtuel-

---

[1] STORY A.E. et WAGNER H.M. : op. cit. p. 219

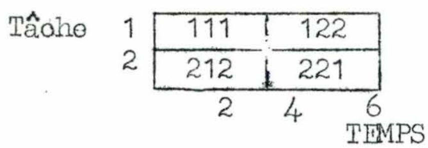
[2] AKERS S.B. et FRIEDMAN j. : "A Non-Numerical Approach to Production Scheduling Problems" ; Journal of the O.R.S.A. vol 3 n° 4 , novembre 1955

[3] HELLER J. : "Combinatorial, Probabilistic and Statistical Aspects of an M x J Scheduling Problem" ; N Y 0; 2540, AEC Research and Development Report , février 1959.

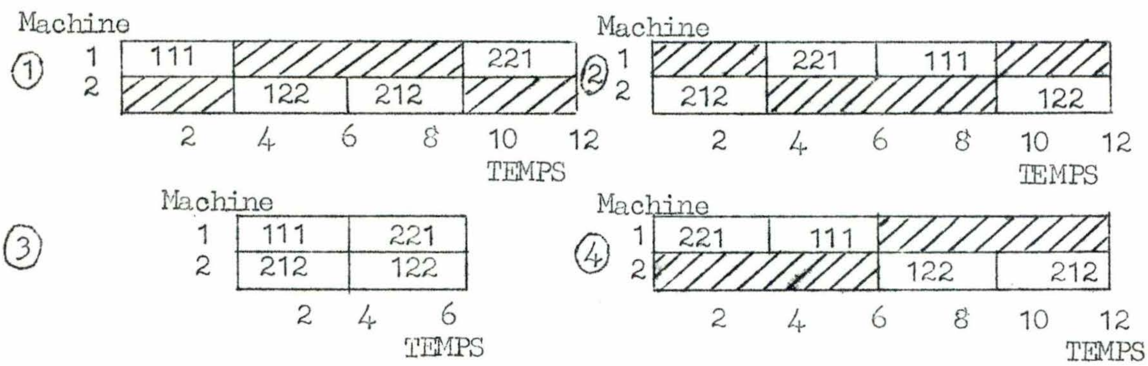
[4] GIFFLER B., THOMPSON G.L. et van NESS V. : "Numerical Experience with the Linear and Monte Carlo Algorithms for Solving Production Scheduling Problems" dans MUTH J.F. et THOMPSON G.L. : op. cit. chap. 3.

[5] ROWE A.J. : "Toward a Theory of Scheduling", Journal of Industrial Engineering, vol. 11 n° 2, mars 1960.

lement possible d'engendrer un nombre infini d'ordonnancements par l'insertion de temps morts, il n'y en a qu'un parmi ceux qui respectent le même ordre des tâches qui domine les autres quant à la mesure d'évaluation. Cet ordonnancement, nous l'appelons 'semi-actif" : il est obtenu par une opération de "faible calage à gauche"; il suffit donc de placer chaque opération le plus à gauche possible jusqu'à ce que qu'elle soit bloquée par l'opération précédente de la même tâche. Remarquons que l'ordre des opérations n'a pas changé : le nombre d'ordonnancements devient fini et s'élève à $(g_1!)(g_2!) \ldots (g_m!)$ où g_k est le nombre d'opérations à traiter par la machine k. Cette limite est une limite supérieure : s'il y a deux opérations par machine, le nombre d'ordonnancements semi-actifs s'élève à 3 et non à 4 car le quatrième viole les contraintes de précédence.



Les trois ordonnancements semi-actifs sont : (l'ordonnancement 4 n'est pas "possible")



Le quatrième aurait été obtenu que s'il n'y avait pas de contraintes de précédence entre les opérations.

Si l'ordonnancement optimal est, par définition, "semi-actif", il est également "actif". Un ordonnancement actif est obtenu par l'opération de "calage à gauche" qui consiste à avancer le moment auquel débute une opération sans retarder la date de début d'une autre opération. Le "faible calage à gauche" est un "calage à gauche", mais l'inverse n'est pas vrai car le calage à gauche ne respecte pas nécessairement l'ordre des opérations sur une machine ; il permet à une opération de s'insérer dans un intervalle de temps mort, lorsque celui-ci est suffisamment long pour l'accueillir. Dans l'exemple ci-dessus, l'ordonnancement 1 n'est pas actif car un calage à gauche est possible en plaçant l'opération 212 avant 122 sans retarder le début de cette opération. A partir de ce nouvel ordonnancement, un "faible calage à gauche" est possible : le résultat est l'ordonnancement 3 qui est le seul ordonnancement actif et qui est donc optimal. Il n'empêche que, dans un problème de plus grande dimension, le nombre d'ordonnancements actifs reste énorme sans qu'il soit possible d'en donner une limite.

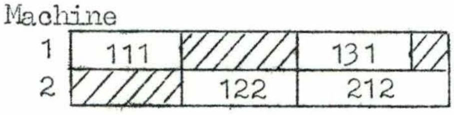
Il reste une dernière classification des ordonnancements. Une opération est "ordonnançable" sur la machine k au moment t si toutes les opérations précédentes de cette tâche ont été ordonnançées et si le traitement de la dernière d'entre elles est achevé. Si, à l'instant t, où la machine k est disponible, il n'existe pas d'opérations ordonnançables sur k, l'ordonnancement est appelé "non-retardé". Cet ordonnancement fait, par définition, partie des ordonnancements actifs, mais ce nouveau sous-ensemble n'est pas dominant en ce sens que l'ordonnancement optimal ne se trouve pas nécessairement parmi les ordonnancements non-retardés.

Exemple :

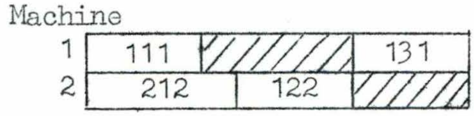
TACHE			
1	111	122	131
2	212		

Les deux ordonnancements actifs de ce problème sont :

1



2



L'ordonnancement 1 n'est pas "non-retardé" car, au moment 0, la machine 2 est inoccupée, alors que l'opération 212 est ordonnançable ; cet ordonnancement est pourtant meilleur que l'ordonnancemement 2 qui est non-retardé ; ils sont équivalents quant à F_{max}, mais l'ordonnancement 1 est meilleur quant à $\overline{F}$ et aux autres mesures régulières en général. Ce sous-ensemble est pourtant important, car il est plus facile d'engendrer des ordonnancements non-retardés que des ordonnancements actifs ; de plus, même s'ils ne contiennent pas l'ordonnancement optimal, ils sont significativement meilleurs que les ordonnancements actifs.

b. Génération d'ordonnancements actifs et non-retardés

Les propriétés de l'ordonnancement optimal nous amènent à nous poser les questions suivantes : comment engendrer un ordonnancement actif ou non-retardé ? Comment établir un choix entre plusieurs opérations ordonnançables au moment t ? Nous répondrons plus précisément à la seconde question dans le paragraphe suivant, mais disons que l'ordre dans lequel les opérations sont choisies et la manière dont leurs dates de départ sont déterminée caractérise UNE procédure de génération d'ordonnancements. La réponse à la première question se trouve dans l'annexe III, section 6 .

§ 2. Règles de décision

a. Système de priorité

La priorité d'une opération est un attribut numérique qui lui est accordé et sur lequel la sélection sera basée. L'objectif est donc la découverte d'un mécanisme qui détermine l'ordre dans lequel les opérations devront être ordonnancées. Un tel système doit être suffisamment précis pour aboutir à une seule sélection : s'il est tel que deux opérations reçoivent la même valeur de priorité, il faudra prendre un second critére de choix.

Exemple : au moment t, 3 tâches sont en conflit (elles sont notées 1,2,3). Le système de priorité pourrait être:

- choisir la tâche dont le nombre d'opérations est le plus grand : si les tâches ont respectivement 3,4,3 opérations, la tâche 2 devient prioritaire ; les deux autres reçoivent la valeur 2 ; il est donc nécessaire de (les deux possédant 3 opérations) prendre un second critère de choix.
- choisir la tâche dont le numéro d'identification est le plus petit ; ce critère tranche le conflit entre les opérations 1 et 3 : l'ordonnancement sera donc 2-1-3.

Tout ordonnancement, qu'il soit optimal ou non, répond par nature à un système de priorité. Le fait de placer une tâche i avant une tâche i' suppose l'existence d'un critère qui rend la tâche i prioritaire à la tâche i'. Toute la difficulté provient de ce que ce système peut être extrêmement complexe et parfois même incompréhensible. Si nous reprenons l'ordonnancement $n/3/F/F_{max}$ de l'annexe IV trouvé par la technique du Branch and Bound, il est à priori impossible de savoir pourquoi la tâche 3 est prioritaire par rapport à la tâche 5 et pourquoi cette dernière l'est par rapport à toutes les autres tâches. Dans un problème d'ordonnancement, il y a autant de systèmes de priorité qu'il y a d'ordonnancements possibles, mais les règles qui sont à la base de ce système restent la plupart du temps implicites sans qu'il soit même possible de les expliciter analytiquement.

b. Les règles de décision

Dans les modèles où des résultats optimaux ont été trouvés, les règles de décision étaient déterminées à postériori : sur base de manipulations algébriques ou autres, nous arrivions à la conclusion suivante : pour avoir un ordonnancement optimal, les tâches doivent être ordonnancées selon la règle de décision X (les modèles à une machine sont les plus clairs à cet égard). Dans un problème plus complexe, le traitement analytique préalable devient impossible et le raisonnement est inversé : puisque l'application de certaines règles (SPT notamment) se sont avérées efficaces dans la recherche d'ordonnancements optimaux pour des problèmes simples, ces règles ne le seraient-elles pas pour des problèmes complexes ? On va donc tester l'efficacité de certaines règles de décision déterminées à priori. L'application automatique de ces règles a permis l'obtention d'ordonnancements satisfaisants, voire optimaux (on trouvera dans l'annexe V un résumé des principales études faites à ce sujet).

Lorsque le critère d'optimalité est une fonction de F, il est apparu opportun de baser les règles de décision sur les durées de traitement des tâches ou des opérations ; lorsqu'il est une fonction des dates de livraison, il semble évident d'insérer, d'une manière ou d'autre les dates de livraison. Les règles lespplus simples et les plus couramment utilisées sont :

1. RANDOM : parmi les opérations ordonnançables sur une même machine, on choisit une opération au hasard ; il s'agit essentiellement d'une règle de référence permettant la comparaison ;
2. FCFS (First come, first served) et FIFO (First in, first out): on choisit l'opération qui est entrée la première dans l'ensemble $\{\Gamma_{so}\}$ [1] (ou dans l'atelier lorsque toutes les tâches n'arrivent pas au même moment) ;
3. SPT ou LPT (Shortest, longest processing time) : parmi les opérations ordonnançables sur une même machine, celle

---

[1] Voir Annexe III section 6

dont la durée est la plus courte (longue) est prioritaire ;

4. MOPNR (Most operation remaining) : on choisit la tâche dont il reste le plus grand nombre d'opérations à traiter indépendamment de la durée de traitement de ces opérations ;
5. LWKR et MWKR (Least and most work remaining) : on choisit la tâche dont le travail restant à faire est le moins (le plus) important ;
6. MWKR-P (Most work remaining-processing time) : on choisit la tâche dont le travail restant à faire après l'achèvement de l'opération ordonnançable est le plus important ;
7. MWKR/P (Most work remaining/processing time) : on choisit la tâche dont le rapport entre le travail restant à faire et la durée de traitement de l'opération ordonnançable est le plus élevé ;
8. DDATE (Due Date) : la tâche dont la date de livraison est la plus rapprochée est prioritaire ;
9. SLACK : on choisit la tâche dont la différence, à un moment donné, entre la date de livraison et la somme des durées de traitement des opérations restant à traiter est la moins élevée ;
10. S/OPN (Slack/opération) : on choisit la tâche dont le rapport entre le temps restant et le nombre d'opérations restant à traiter est le moins élevé.

Quelques conclusions sont tirées dans l'annexe V sur l'efficacité de ces règles. Résumons les en une proposition : la simulation, telle que nous l'avons décrite, est <u>fermée</u> : elle agit de façon automatique quel que soit le problème particulier auquel on la soumet. Les méthodes heuristiques abordées dans la troisième partie ont pour effet de rendre cette simulation ouverte : une simulation est ouverte lorsqu'elle est dynamique et adaptive.

Section 5 : Notes sur l'ordonnancement dynamique

§ 1. Principe de base

L'hypothèse la plus irréaliste des modèles statiques se résume ainsi : "tout est connu".

Dans l'ordonnancement dynamique par contre,

- les dates d'arrivée des tâches et leurs durées de traitement sont distribuées selon certaines lois de probabilités
- le processus est continu : à la limite, il n'y a plus qu'un problème d'ordonnancement.

§ 2. Principe de résolution

Puisque les tâches arrivant devant les machines ne peuvent obtenir immédiatement d'elles le service qu'elles dispensent parce que ces tâches sont trop nombreuses par rapport à la capacité des machines, nous avons affaire à un problème de réseaux de files d'attente. Ce problème n'est soluble que si l'on peut caractériser les deux paramètres essentiels de tout problème de files d'attente :

- le rythme des arrivées : on suppose toujours qu'il est Poissonien (λ)
- le rythme de service : on suppose souvent qu'il est exponentiel, hyperexponentiel (Erlang) ou uniforme (μ).

Si le taux des arrivées dépasse le taux de service, le système est dit "saturé", car les files d'attente vont s'allonger. D'où l'importance du coefficient d'utilisation de l'atelier qui s'exprime comme le rapport entre le taux d'arrivées et le taux de service. ($\rho = \frac{\lambda}{\mu}$)

§ 3. Résultats

Les critères tels que min F_{max} n'ont aucune signification puisque le processus est continu. On utilisera principalement des critères tels que :

- l'espérance mathématique de la durée moyenne d'occupation de l'atelier par une tâche $E(\overline{F})$,
- l'espérance mathématique de la durée moyenne d'attente d'une tâche $E(\overline{W})$,
- l'espérance mathématique de la longueur des files d'attente etc...

En fonction du critère, on jugera de la valeur de telle ou telle règle de priorité (la règle SPT est très importante à cet égard).

§ 4. Limites

Excepté dans des cas très particuliers, la théorie des files d'attente n'a pu résoudre de facon optimale des problèmes d'ordonnancement que lorsque l'atelier ne comprend qu'une seule machine. Dans les autres cas, il faut recourir à la simulation.

CONCLUSION DE LA SECONDE PARTIE

1. Même lorsque le problème est de dimension modeste, l'énumération complète des ordonnancements est une tâche qui devient rapidement impossible. Il faut donc recourir à des méthodes qui réduisent le nombre d'ordonnancements à explorer. On a donc eu recours à l'analyse combinatoire qui n'est utile que lorsque le problème est simple à formuler. Encore faut-il que le critère soit simple (nous avons suffisamment insisté sur la difficulté de prendre en charge des critères qui ne sont pas des mesures d'évaluation

régulières). Au fur et à mesure que le nombre de tâches et de machines s'élève, l'obtention d'ordonnancements optimaux devient aléatoire ; si nous mettons à part l'algorithme de Johnson et la technique du Branch and Bound, aucun résultat n'est disponible. Même la programmation linéaire n'est pas satisfaisante, car si elle permet la formulation des contraintes d'ordonnancement et de routing, le système de contraintes n'est pas suffisant pour aboutir à une solution. Comme le signalent Muth et Thompson : "It should be emphasized that in the solution of an integer linear programming problem, additional constraints are added during the course of its solution so that the ultimate number of constraints will be considerably larger. (pour un problème 3/4, la formulation de Wagner et de Bowman exige entre 300 et 600 variables et un nombre bien plus élevé de contraintes). None of these authors claims that his formulation is computationally parctical..." [1]

2. Cette remarque de Muth et Thompson nous invite à répéter ce qui a été dit antérieurement : la résolution d'un problème d'ordonnancement doit être rapide, simple dans son application et peu coûteuse, sous peine de perdre le bénéfice d'un bon ordonnancement. C'est pour cette raison, que les recherches se sont orientées vers les techniques de simulation où la priorité est accordée à la simplicité et à la rapidité au détriment de l'obtention du meilleur ordonnancement possible. Il existe des règles de décision complexes et légèrement plus efficaces, mais l'amélioration est trop faible pour mériter l'effort à founir pour y parvenir. La critique essentielle qu'on pourrait adresser à ces règles de décision, réside dans leur caractère aveugle et automatique ; elles sont appliquées quel que soit le problème, indépendamment de la "personalité" propre de chaque problème. Même si les résultats obtenus peuvent encore paraître satisfaisants, il ne faut pas perdre de vue qu'ils sont sujets à des hypothèses extrêmement restrictives (principalement l'interdiction du fractionnement, l'existence d'une seul machine par type et l'ignorance des durées de préparation-machine); même la simulation

---

[1] MUTH J.F. et THOMPSON : op. cit. introduction p. XI

n'est pas toujours capable de les lever.

3. Sans avoir recours à des considérations d'environnement économique (voir première partie), un ordonnancement optimal n'est pas nécessairement une bonne chose en soi ; il peut même ne pas être souhaitable. Supposons que, dans un problème quelconque, nous trouvions trois ordonnancements dont les valeurs du retard moyen soient respectivement de 4,4,5 ; nous supposerons en outre que les deux premiers sont optimaux à cet égard. Les retards se distribuent de la façon suivante :

	ord. 1	ord. 2	ord. 3	ord. 4
tâche 1 :	0	0	7	8
2 :	0	0	3	0
3 :	16	8	6	4
4 :	0	8	4	4
$\overline{T}$:	4	4	5	4
T_{max} :	16	8	7	8

Si les coûts de retard ont une forme exponentielle, ce qui est souvent le cas, et sont de la forme $C = \sum_{i=1}^{n} T_i^2$, nous aurons pour l'ordonnancement 1 : $C_1 = (16)^2 = 256$

2 : $C_2 = 2(8)^2 = 128$

3 : $C_3 = 49 + 9 + 36 + 16 = 110$

En résolvant le problème, nous aurions choisi indistinctement les ordonnancements 1 et 2, alors que, du point de vue du coût, le second est meilleur ; nous aurions en outre rejeté l'ordonnancement 3, alors qu'il est optimal en ce qui concerne les coûts. Il est partiellement erroné de croire que prendre le critère T_{max} était préférable. C'est exact dans le cas ci-dessus lorsqu'on compare les trois premiers ordonnancements. Si on ajoute l'ordonnancement 4, il est optimal quant à $\overline{T}$, tandis que l'ordonnancement 3 reste optimal quant à T_{max} ; or, la valeur de $C_4 = 96$. A lui seul, ce critère est insuffisant ; même s'il existait une méthode permettant de trouver des ordonnancements optimaux tant vis-à-vis de $\overline{T}$ que de T_{max}, on n'est pas certain d'aboutir à un seul ordonnancement. S'il y en a plusieurs, rien ne permet de choisir entre eux, si ce n'est le calcul des coûts.

Notons cependant que Reinitz [1] a obtenu, par l'utilisation de la programmation dynamique, un ordonnancement optimal quant à la minimisation de la somme des coûts de retard, de stockage et de préparation-machine. Cette technique est semblable à celle que nous avons étudiée dans le chapitre 2 : il suffit de remplacer (dans la matrice) les durées de préparation-machine par des coûts. Mais cet ordonnancement aura des répercussions sur l'ensemble de l'entreprise :

- il peut exiger un investissement en matériel et en main-d'oeuvre ;
- une révision de la tenue des carnets de commande ;
- la politique d'entretien ...

Il est donc important de remettre le problème d'ordonnancement dans son contexte en se posant les questions suivantes :

- Quelle est la hiérarchie des critères ? Le goodwill, la pleine capacité ?
- Quelles sont les contraintes ? Matériel, main-d'oeuvre, marché, ressources ?
- Quel est le coût d'obtention de l'ordonnancement optimal ? Quelles sont les limites à ne pas dépasser ?
- Le meilleur ordonnancement obtenu, quelles en sont les implications quant à l'organisation de la production et la politique à suivre dans l'avenir (investissement, goodwill) ?

---

[1] REINITZ R.C. : "On the Job-Shop Scheduling Problem" dans MUTH J.F. et THOMPSON G.L. : op. cit. chap 5.

4. Toutes les méthodes envisagées dans cette partie sont rigides :

   - le critère utilisé est unique,
   - les hypothèses sont très restrictives,
   - les solutions sont rigides :
     . elles ne s'attachent pas à la "personnalité" du problème envisagé,
     . elles ne sont pas adaptives, lorsque les hypothèses ou/et les fonctions-critères de l'entreprise changent.

Nous allons tenter de pallier cette rigidité en parlant des méthodes heuristimues.

TITRE III :

L'APPROCHE HEURISTIQUE DES PROBLEMES D'ORDONNANCEMENT

Chapitre 1 :

Vers une autre approche du problème d'ordonnancement

Nous n'avons pas l'intention de reprendre toutes les critiques que nous pourrions élever à propos de l'approche algorithmique. La seconde partie nous a permis de situer ces critiques ; cependant, certaines questions n'y ont pas reçu de réponse satisfaisante. La seconde partie se termine, en effet par quelques points d'interrogation :

- vu les résultats obtenus, le travail fait jusqu'à présent était-il utile ?
- si oui, était-il suffisant ?
- si la réponse à cette dernière question est négative, que pouvons nous faire ?

Cette troisième partie consiste dans un essai de justification d'une réponse positive à la première question et négative à la seconde.

Section 1 : Critique de l'approche algorithmique

Malgré nos efforts pour garder l'ordonnancement dans son contexte d'environnement économique, nous lui avons donné des solutions particulières ; nous en avons étudié quelques facettes, qui loin de rejaillir sur l'environnement, restaient au niveau de l'ordonnancement lui-même.

Nous pensons que c'est dans le caractère inapproprié de la réponse à la question posée qu'il faut trouver la raison essentielle de l'insuffisance de l' approche utilisée jusqu'ici. Cette expérience était pourtant utile, ne fût-ce que parce qu'elle nous a permis d'aboutir à cette conclusion. Qu'il s'agisse d'un problème d'ordonnancement ou de tout autre problème se posant à l'entreprise, il existe un danger certain de vouloir lui appliquer des solutions qui ne respectent pas toujours le caractère particulier du problème. En effet, qu'avons nous fait jusqu'à présent ? Après avoir situé l'ordonnancement dans son contexte économique, nous avons dressé un inventaire des méthodes de résolution. Devant des difficultés inhérentes au problème lui-même, nous l'avons, consciemment ou non isolé du reste, alors que la première partie avait pour objectif de résister contre cette tentation.

La recherche opérationnelle pourrait à cet égard présenter deux dangers :

- vouloir justifier une décision déjà prise alors qu'elle devrait servir à l'étayer ;
- vouloir utiliser un certain nombre d'outils préexistants, qui ne s'ajustent pas nécessairement au cadre dans lequel on voudrait les voir entrer. Elle doit au contraire trouver des outils qui soient étudiés en fonction du problème posé.

Il est significatif à cet égard que la majorité des ouvrages traitant de l'ordonnancement (celui de Conway, Maxwell et Miller par exemple) n'insistent pas sur les limites pratiques des modèles exposés. Cette critique se vérifie particulièrement lorsque les auteurs abordent le problème dynamique : la théorie des files d'attente y est fort développée, alors que son application pratique est fort limitée. (C'est la raison pour laquelle nous n'avons pas insisté sur l'aspect théorique des modèles dynamiques)

Quel est notre objectif ? Trouver un ordonnancement "satisfaisant" [1]

---

[1] Nous reviendrons sur cette notion d'ordonnancement "satisfaisant". Disons qu'il s'agit d'un ordonnancement vérifiant certains paramètres ou limites de contrôle fixés à priori et à l'intérieur desquelles on voudrait que l'ordonnancement évolue.

par une méthode simple, rapide, pratique, générale et peu coûteuse. C'est parce que la méthode énumérative est lente et coûteuse que nous la rejetons ; c'est parce que les algorithmes (Johnson et Branch and Bound en particulier) manquent de généralité que nous les laissons de côté ; c'est parce que la programmation linéaire en nombres entiers est complexe que nous la mettons entre parenthèses ; nous reprochons à l'utilisation de règles de priorité leur aspect automatique et aveugle. En un mot, toutes ces méthodes sont particulières et fermées.

Est-ce une raison pour qualifier l'approche algorithmique d'inutile ? Nous ne le pensons pas car elle nous a permis :

- de poser le problème et de le circonscrire : objet, critère, contraintes..
- de prendre conscience des limites d'adaptation de modèles théoriques aux problèmes réels, à cause des hypothèses restrictives que nous avons dû y insérer,
- de situer les points faibles et les difficultés inhérentes au problème,
- de faciliter les recherches ultérieures par des conclusions du type suivant (par exemple) : "les ordonnancements non-retardés sont statistiquement meilleurs que les ordonnancements actifs, bien que ceux-ci comprennent l'ordonnancement optimal (lorsque le critère est F_{max}).
- d'abandonner l'idée de pouvoir établir une théorie de l'ordonnancement ; nous ne pouvons construire que des modèles (pas nécessairement mathématiques).

L'approche algorithmique est donc constructive mais elle nous force à revoir le problème d'ordonnancement sous un angle complètement différent. Dans la suite, nous travaillerons simultanément sur deux niveaux : un niveau général et un niveau spécifique, celui de l'ordonnancement. Nous estimons en effet que notre raisonnement pourrait s'appliquer à n'importe quel problème d'entreprise, mais traitant de l'ordonnancement, nous utiliserons ce problème particulier pour le justifier.

Section 2 : Structure générale du processus décisionnel

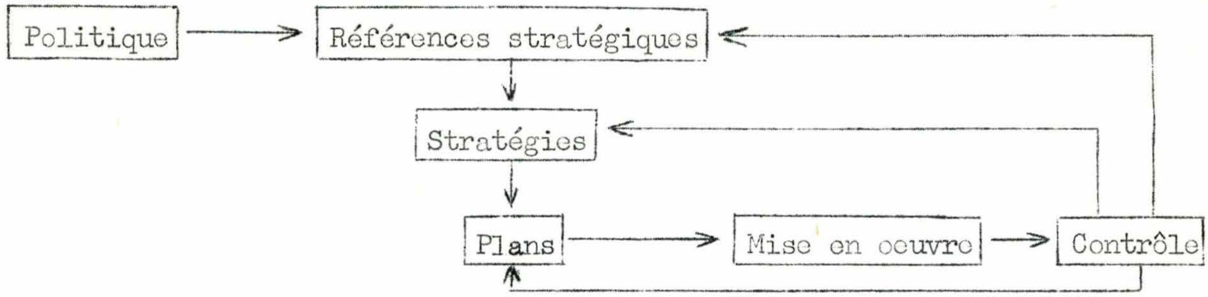
Si l'ordonnancement n'est pas un problème purement "technique" et s'il n'est pas à considérer en dehors de son environnement, qu'est-il ? C'est à cette question que nous allons tenter de répondre ?

Le problème d'ordonnancement rentre dans le cadre des problèmes de décision et plus particulièrement de ceux tenant à l'organisation de la production: sa résolution implique et dépend des décisions relatives aux stocks, aux clients, aux moyens financiers, aux investissements ... (cfr. Titre I)

Une des fonctions afférant à la direction générale d'une entreprise consiste dans l'établissement de "références stratégiques" [1] qui seraient des "éléments plus ou moins formalisés d'un certain projet de l'entreprise". Comme la fixation des états détaillés du futur est impossible, nous ne pouvons parler que de "références". A moyen terme, ces "références" se transforment en stratégies, qui se définissent comme étant des décisions conditionnelles prises au départ et permettant de répondre aussi adéquatement que possible aux situations dans lesquelles l'entreprise peut se trouver. Pour être opérationnelle, une stratégie se traduit en plans, ensembles d'actions totalement déterminées. Lorsqu'un problème est formalisé, on l'appelle "modèle", résultat de la formalisation en un langage approprié des structures d'un phénomène et d'un processus de décision s'appliquant au contrôle de ce phénomène. Qui dit contrôle, dit mise en oeuvre d'actions correctives ; ce contrôle donne lieu à un mouvement de feedback vers les références stratégiques, les stratégies et les plans. Le circuit du processus décisionnel serait donc schématisé de la façon suivante :

---

[1] Nous tenons à remercier le professeur M. Bourgeois des H.E.C. à Paris, qui nous a permis de reprendre une grande partie des idées qu'il avait émises lors d'une conférence donnée le 23 et 24 octobre 1968 à l'Ecole Supérieure de Guerre Aérienne et qui avait pour thème : "Les modèles de décision".



Si l'entreprise, définie comme système (réunion en une totalité d'un ensemble d'éléments en relation étroite les uns avec les autres) pouvait être entièrement formalisée, la démarche serait la suivante : puisqu'il appartient à la direction générale de fixer les références stratégiques, les responsables de la mise en application du système utiliseront celles-ci en les considérant comme des contraintes et/ou comme des critères. N'étant que des références, les "références stratégiques" ne seront jamais exprimées en termes absolus ; elles ne sont donc pas des contraintes au sens strict du terme. Il s'agit plutôt de la détermination d'une hiérarchie de valeur entre les critères choisis. A partir de ces références, les responsables de la mise en application du système vont pouvoir établir leurs plans : ils se fixeront des normes qui seront, momentanément considérées comme contraintes. Si le système sort de ces normes, de deux choses l'une :

- on applique un autre plan qui permet au système de rester dans les normes établies ;
- si l'application de tous les plans possibles ne permettent pas au système de rester dans les normes, il faudra les rendre moins restrictives. N'oublions pas qu'elles sont formulées dans le cadre des références stratégiques ; il faudra donc modifier ces normes ce qui n'est pas nécessairement du ressort des responsables de la résolution d'un problème particulier.

La perspective statique de ce raisonnement consiste à se fixer des normes qui soient à la limite des références stratégiques ; la perspective dynamique, par contre consiste à resserrer ces normes au maximum (tant que le système continue à évoluer à l'intérieur d'elles) pour pouvoir éventuellement modifier les références en les rendant plus contraignantes ; ces nouvelles références serviront de base à la résolution des problèmes se posant à l'avenir.

Section 3 : Modèle versus système

Comme en réalité, l'entreprise-système n'est pas formalisable, il s'avère nécessaire de décomposer le système en "modèles" qui seraient les parties formalisables d'un système. Au lieu d'avoir affaire à un problème (LE problème de l'entreprise), nous aurons affaire à de nombreux sous-problèmes. Le rôle des références stratégiques consistera alors à sauvegarder une cohérence entre les multiples composantes du système. Un problème particulier ne sera plus étudié en fonction de ses propres exigences mais en relation avec son environnement par l'intervention de contraintes ou de paramètres indépendants dont il faudra tenir compte lors de sa résolution. Il ne s'agit donc plus d'étudier l'ordonnancement en soi, comme un problème indépendant mais de le résoudre en fonction de directives coordonnatrices élaborées à un niveau supérieur de décision. Contrairement à la deuxième partie, il ne s'agit plus d'élaborer des modèles particuliers mais un modèle général d'ordonnancement, point de départ éventuel d'une théorie. Notons à ce propos que :

- la théorie vise à la généralité alors que le modèle vise à la spécificité : si un modèle "colle" à un problème particulier, la théorie vise à rendre compte du plus grand nombre de phénomènes avec le plus petit nombre de propositions (rapports formels) ;
- un modèle est plutôt orienté vers l'action tandis que la théorie est plutôt orientée vers la connaissance (En fait, il y a interaction entre modèles et théorie).

Le raisonnement exposé dans la section précédente est applicable aux modèles. Le domaine de représentation phénoménologique du modèle se définissant dans l'espace par des frontières et dans le temps par un horizon, le modèle est presque toujours un sous-modèle par rapport à un modèle dont l'espace est plus vaste et l'horizon plus long ; dans cette prespective, le système serait un "super-modèle" pouvant être formalisé par un ensemble de modèles organisés hiérarchiquement dans l'espace et le temps.

Chapitre 2 :

L'ordonnancement, problème heuristique

Section 1 : Classification des modèles [1]

Selon la nature des phénomènes dont le modèle doit rendre compte, il sera ouvert ou fermé. Un modèle est fermé lorsque toutes les éléments nécessaires à la description du phénomène sont connus avec exactitude ou en probabilité et où toutes les relations entre les éléments sont clairement établies. Les modèles ouverts sont ceux qui ne possèdent pas ces caractéristiques.

Les modèles fermés se subdivisent en trois catégories :

- Le modèle est rigide lorsque le processus de décision qu'il inclut est complètement formalisé ;
- un modèle sera adaptif lorsque le processus n'est pas complètement formalisé mais [illegible] est susceptible de modifications en fonction de la variation de certains paramètres ; dans ces modèles, les processus de décision sont prévus à l'avance ;
- un modèle basé sur l'apprentissage statistique est un modèle où le processus de décision n'est pas prévu à l'avance mais s'engendre de lui-même par inférence statistique à partir de données observées ; ces modèles ne sont applicables que dans la mesure où l'environnement observé correspond à la classe d'environnement pour laquelle la procédure de résolution a été établie et où les critères de procédure s'avèrent applica-

---

[1] Nous n'avons pas la prétention de donner une classification des modèles, qui soit universellement admise.

bles.

Tous les modèles fermés donnent lieu à une procèdure de résolution algorithmique. L'algorithme est une procédure déterminée dans la totalité de son déroulement et conduisant avec certitude à une solution unique ; un algorithme est donc toujours un processus univoque de résolution de problèmes et de prise de décision. Il en résulte qu'il est spécifique à une structure de problèmes particuliers ; les algorithmes généraux s'appliquent à la résolution de problèmes différents dans la mesure où ceux-ci peuvent être considérés comme possédant des structures analogues.

Lorsqu'un modèle est ouvert, deux procédures de résolution sont possibles : la procédure algorithmique et la procédure heuristique. Le choix de la procédure à utiliser est fonction de la nature du problème posé ("technique" ou décisionnel) ou de la manière dont ceux qui doivent le résoudre le considèrent. Cette distinction apparaîtra plus claire dans la suite de l'exposé.

Section 2 : Définition du problème heuristique

La recherche opérationnelle a jusqu'à présent montré son efficacité dans le traitement de problèmes bien structurés mais non dans le traitement de ceux qui le sont moins. Bien que certains problèmes puissent être catalogués comme bien ou mal structurés, la distinction est souvent malaisée. Tout dépend du niveau auquel on se situe (voir infra).

De façon générale, un problème sera appelé "bien structuré" lorsqu'il répond aux trois caractéristiques suivantes : [1]

- toutes les variables significatives et toutes les relations entre elles

----

[1] SIMON H.A. et NEWELL A. : "Heuristic Problem Solving : the Next Advance in Operations Research". Journal of the O.R.S.A. vol. 6 n°1 janvier 1958

peuvent être décrites en termes mathématiques et logiques,

- les objectifs (critères) sont spécifiés dans une fonction bien définie,
- il est possible d'appliquer à ces problèmes des routines de calcul permettant d'aboutir à une solution (existence d'algorithmes).

En un mot, un problème est bien structuré lorsqu'il est susceptible d'une formalisation explicite et quantitative, et résoluble par des techniques connues et applicables.

Lorsqu'une de ces caractéristiques est absente, le problème est dit "mal structuré" et devra être résolu par une procédure heuristique. Le domaine des processus heuristiques de décision est donc celui des problèmes de gestion dont la structure n'est pas exactement connue, soit parce que les valeurs prises par les différents facteurs sont incertains, soit parce que les relations entre ces facteurs ne sont pas ou mal identifiées. Il en résulte qu'un algorithme rigoureux ne peut être programmé ; le décideur devra donc adopter un comportement adaptif : sur base d'informations supplémentaires qu'il acquérra progressivement, il ajustera son modèle ou précisera sa règle de décision. De tels problèmes se rencontrent tant aux niveau des décisions stratégiques (plus général) qu'au niveau d'exécution (problème spécifique, l'ordonnancement par exemple).

Section 3 : L'ordonnancement non unitaire, problème heuristique

Voyons comment, sur base des trois caractéristiques citées ci-dessus, l'ordonnancement est un problème de type heuristique :

§ 1. L'énoncé du problème

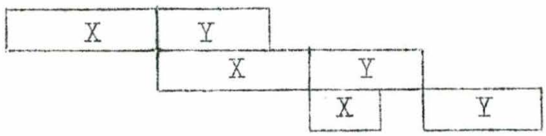
L'interdépendance entre les variables significatives n'a jamais pu être formalisée. Les variables introduites dans les modèles de la seconde partie sont essentiellement des fonctions de la durée d'occupation de l'atelier, des retards, des stocks intermédiaires, des courbes de charge de machines, du nombre de tâches en attente dans l'atelier etc... Aucune relation formelle n'a jamais pu être introduite entre ces variables.

De plus, le problème se compose essentiellement de deux types de contraintes : les contraintes de routing et les contraintes d'ordonnancement. Au début de la seconde partie, nous avons insisté sur la différence de nature existant entre ces contraintes. Les premières sont très restrictives et si elles étaient seules, le problème serait résolu (P.E.R.T.). L'introduction des secondes entraîne un grand nombre de degrés de liberté supplémentaires. La contrainte d'ordonnancement n'est pas une contrainte au sens strict du terme. Prenons le cas de la contrainte de routing : elle a pour effet de fermer et de délimiter le domaine des solutions possibles ; il en est de même dans un problème résoluble par programmation linéaire : après avoir démontré que dans un tel problème, la solution devait se trouver sur une des frontières du domaine des solutions possibles, domaine délimité par les contraintes, le Simplexe consiste dans une exploration systématique de ces frontières. La "contrainte" d'ordonnancement, au contraire a pour effet d'ouvrir le domaine des solutions. Le critère utilisé n'en gardera qu'une. Il est assez paradoxal de constater que la résolution d'un ordonnancement consiste à introduire des contraintes au sens strict du terme puisqu'elle spécifie que la tâche X doit précéder la tâche Y.

Pour illustrer ce paradoxe, inversons l'énoncé du problème : "Trouver l'ordonnancement optimal (en fonction de F_{max}) de deux tâches, possédant chacune trois opérations à traiter par trois machines ; les contraintes sont :

- la première opération d'une tâche doit précéder la deuxième qui elle même doit précéder la dernière (contrainte de routing) ;
- la tâche X doit précéder la tâche Y (nous remplaçons la contrainte d'ordonnancement par la solution d'un problème réel) ;.

L'effet de la première contrainte est de réduire le nombre de solutions possibles. Si nous utilisons le théorème de Roy (cfr. Titre II, chapitre 3, section 1), il nous reste deux ordonnancements possibles : X-Y et Y-X (sur toutes les machines). La seconde réduit ce nombre à un : le seul ordonnancement, respectant le système des contraintes est le suivant :



Ce qui dans le véritable problème apparaît comme une contrainte apparaît ici comme une conséquence et comme la solution : une machine ne peut traiter qu'une opération à la fois. Cette solution (comme dans les problèmes de programmation linéaire) est implicitement comprise dans le système des contraintes et de la fonction critère.

Nous avons donc affaire à un <u>modèle "ouvert"</u> lorsque le domaine des solutions possibles n'est pas délimité avec précision. C'est à cause de l'inexistence de contraintes suffisamment restrictives que la programmation linéaire n'est pas à même de résoudre un problème d'ordonnancement.

§ 2. La fonction-critère

Qu'elle ne soit pas bien définie découle de ce qui procède. Dans le problème "inversé", le problème ne se posait même pas, parce que les contraintes étaient tellement restrictives qu'elles n'aboutissaient qu'à un seul ordonnancement. Dans un problème résoluble par programmation linéaire, le critère a pour fonction l'établissement d'une hiérarchie de valeur entre les diverses solutions possibles : il réduit encore plus le domaine des solutions. Dans le problème d'ordonnancement réel, il est impossible de définir le critère puisque le domaine des solutions n'est pas délimité ; c'est probablement pour cette raison qu'il n'a pas de relation évidente avec le système des contraintes. Expliquons-nous.

Dans le cas d'un programme linéaire, nous avons une variable d'action: les quantités à produire de chaque bien. De nombreuses combinaisons de production étant admissibles, on les classe selon le critère utilisé ; celui-ci reflète un objectif général (maximisation du chiffre d'affaires, du bénéfice net, du profit ...). En matière d'ordonnancement, par contre, si la variable d'action est "l'ordre des tâches", il n'y a pas de relation stricte (voir §1) entre cette variable et une fonction-critère ; il s'avère impossible d'établir un critère général débordant le cadre de l'ordonnancement proprement dit : il reste situé au niveau de l'ordonnancement lui-même.

La difficulté d'énoncer une fonction-critère claire et générale nous a obligé à n'en retenir qu'un aspect, en supposant qu'il correspondait aux désirs ou aux besoins de l'entreprise. Dans l'approche algorithmique de la seconde partie, le critère était choisi en fonction de sa simplicité, de sa faculté d'être énoncé en termes quantifiables, des résultats auxquels il aboutit etc... En un mot, il était défini en fonction des possibilités et non pas en fonction des besoins. Cette manière de voir peut entraîner des incohérences et des contradictions lorsque la solution de l'ordonnancement est remise dans son contexte. Il suffit de se rappeler ce que nous disions en terminant la seconde partie : trouver un ordonnancement optimal n'est pas un but en soi ; même si l'objectif de la firme consiste à garder son goodwill en éliminant les retards, il est jusqu'à présent impossible de trouver un ordonnancement qui soit complètement satisfaisant à cet égard (certaine combinaison entre le retard moyen et le retard maximum).

La démarche heuristique change la nature du critère grâce à l'intervention des références stratégiques. Ce n'est plus en fonction des possibilités mais en fonction de ces références (qui ne sont pas définies par les responsables de l'ordonnancement) qu'un problème doit être résolu. De plus, définies à un niveau supérieur de décision, elles donnent des orientations à toutes les parties de l'entreprise et maintiennent par conséquent une certaine cohérence entre ces parties.

Elles permettent enfin de tenir compte de plusieurs critères. Un bon ordonnancement (ordonnancement satisfaisant) ne sera plus celui qui minimise les retards ou la durée d'occupation de l'atelier, mais celui qui reste et dans certaines limites de retards, et dans certaines limites d'occupation de l'atelier, et dans certaines limites d'attente des tâches etc ... (voir infra).

§ 3. Il n'existe pas d'algorithme rigoureux

Puisque la démarche algorithmique revient à établir des relations univoques entre les contraintes et la fonction-critère, encore faut-il que les contraintes , les variables et le critère soient bien définis. La démarche heuristique n'aboutit pas nécessairement à un ordonnancement unique car il est possible d'en trouver plusieurs qui restent dans les limites définies par les références stratégiques. Dans une perspective dynamique et par l'examen des différents ordonnancements obtenus, on resserre les limites ce qui en fin de compte donne un seul ordonnancement.

Section 4. L'ordonnancement, modèle ouvert

Nous avons dit que les modèles ouverts pouvaient être résolus de façon algorithmique ou heuristique et qu'il pouvait s'avérer difficile de se prononcer sur la bonne ou mauvaise structuration d'un problème. Il est temps de nous expliquer sur ces deux points.

Il faut distinguer entre le type de problème et le type de procédure. Si une procédure heuristique suppose que le problème soit considéré comme tel, il semble qu'un doute puisse subsister quant à la véritable nature d'un problème, au simple vu de son énoncé. Prenons le cas de l'ordonnancement. On peut affirmer que d'algorithmique le problème est devenu heuristique. Ce changement a eu lieu lorsque de trois machines, nous avons passé au modèle plus général. Lorsque le critère est F_{max} et que le modèle est du type $n/2/F/F_{max}$, la procédure était algorithmique ; la raison réside dans les caractéristiques particulières de ce modèle : nous avons, grâce au théorème de Roy introduit des contraintes supplémentaires (l'ordre de passage sur les deux machines est identique) ; de plus l'atelier était uniforme. Pour F_{max}, la fonction objectif consistait à minimiser les temps morts que se présentent sur la deuxième machine. Ces particularités supplémentaires ont permis à Johnson de disposer de relations très précieuses :

$$F_{max} \geqslant \sum_{i=1}^{n} A_{[i]} + B_{[n]}$$

$$F_{max} \geqslant A_{[1]} + \sum_{i=1}^{n} B_{[i]}$$

Si on se réfère à l'annexe III, section 5, on se rendra compte que c'est grâce aux contraintes supplémentaires qu'une relation univoque a pu être trouvée (entre la durée de traitement la plus courte et la minimisation des temps morts) et que la procédure algorithmique est possible. Ce modèle est fermé et rigide. Il s'ouvre en se généralisant soit par l'augmentation du nombre de machines, soit par la disposition quelconque de l'atelier. Il n'y a plus de contraintes supplémentaires permettant de trouver un algorithme. Deux possibilités de résolution s'offrent à nous : la procédure heuristique (voir infra) et la simulation ; cette dernière s'applique lorsque le processus de décision est indépendant de la nature du problème. Elle forme le pont entre la procédure heuristique et algorithmique et appartient à l'une ou à l'autre selon la manière dont elle est considérée :

- la simulation, outil de recherche : on cherche à savoir si certaines règles d'affectation (de décision ou de priorité) sont meilleures que d'autres ; dans ce cas, elle est algorithmique, car elle présente un caractère routinier, automatique et facile à programmer ;
- la simulation, procédure opératoire : le phénomène est simulé sur un certain horizon de façon à trouver un bon ordonnancement. Cette simulation débute avec l'état global du système de planning de production ; ensuite, on identifie les délais critiques, les goulots d'étranglement, les niveaux de rupture de charge, ... En fonction de ces éléments, on réajuste l'ordonnancement en modifiant les périodes, les règles locales d'affectation jusqu'au moment où l'on ne trouve plus d'améliorations possibles. Cette démarche est heuristique car elle consiste à adapter les

règles de priorité en fonction de la situation particulière et non à les appliquer de façon automatique.

Nous voudrions terminer ce chapitre en soulignant qu'un problème particulier quel qu'il soit, n'est pas, par définition bien ou mal structuré. La nature algorithmique ou heuristique du problème dépend du niveau de généralité auquel il se situe, des hypothèses implicites du décideur quant à la nature du problème et des hypothèses (contraintes) supplémentaires (explicites ou non).

Chapitre 3 :

Procédures algorithmiques et heuristiques

Section 1 : Place de la procédure dans un modèle de décision

Si des difficultés quant à la distinction de la nature d'un problème peuvent subsister, elles s'estompent lorsqu'on compare les procédures. Pour bien situer la place de la procédure méthodologiques d'une décision, nous voudrions rappeler les étapes méthodologiques d'une démarche décisionnelle s'appuyant sur des modèles mathématiques et sur l'utilisation d'ensembles électroniques :

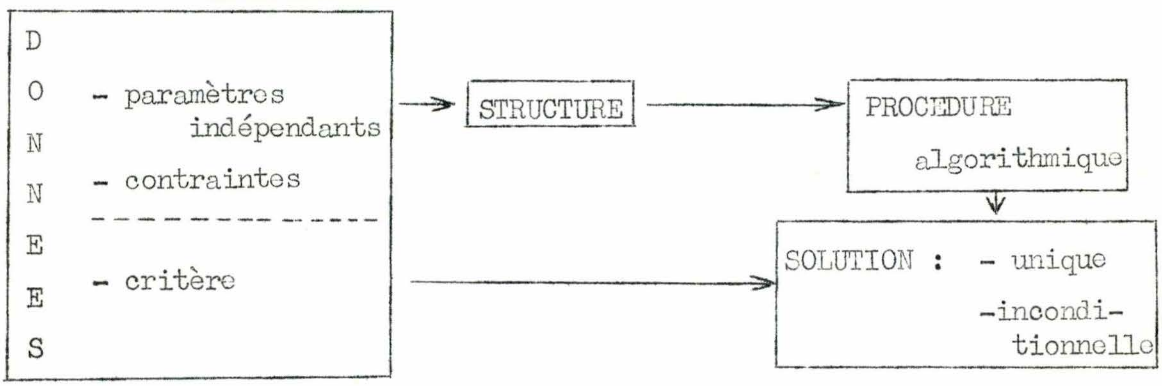
- poser le problème : énoncé et nature du problème ; rassemblement des données significatives, définition du domaine, dans l'espace et le temps : les frontières et l'horizon ; définition de l'objectif.
- définition des données : cela suppose :
  - . le choix d'un critère : un objectif multidimensionnel peut donner lieu à la définition d'un critère d'un ordre supérieur (synthétique ou composite) ou au choix d'un critère unique, les autres étant considérés comme des contraintes) ;
  - . la définition des contraintes ;
  - . la définition des paramètres d'action ou/et indépendants

- l'investigation des structures du phénomène : étude de l'ensemble des rapports existant entre les données (similitude, différence, corrélation);
- choix d'une procédure ou d'un procédé de calcul : algorithmique ou heuristique ;
- inventaire des solutions possibles : hypothèses sur les variations des paramètres indépendants et options concernant les paramètres d'action ;
- construction du modèle proprement dit ;
- programmation du modèle ;
- résolution et évaluation des solutions : validation du modèle et analyse de sensibilité ;
- applications : si la décision n'est pas répétitive, établissement d'un plan d'action et de contrôle ; si elle l'est, mise en route d'une routine.

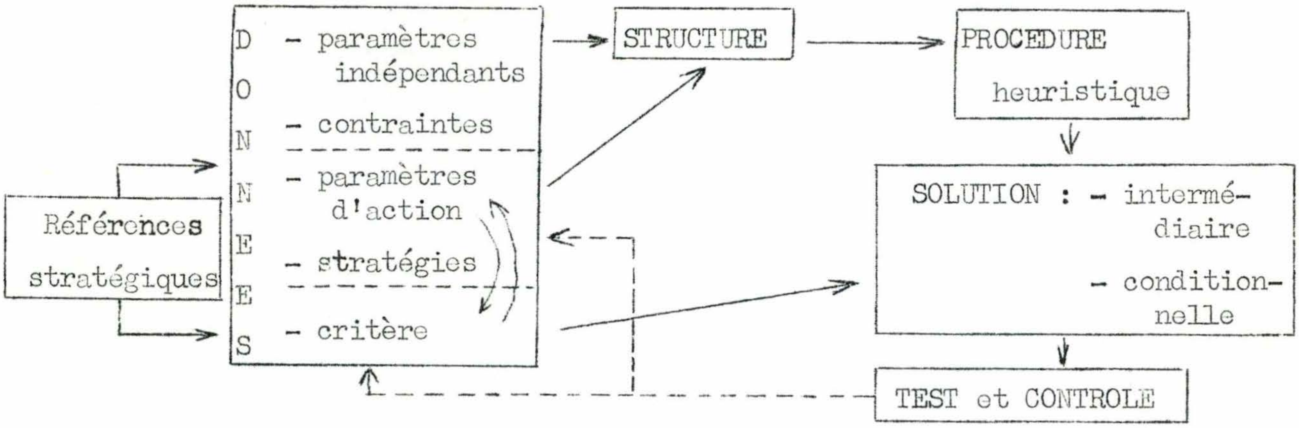
Section 2 : Schéma de la démarche décisionnelle selon le type de procédure

Suivant le type de procédure - algorithmique ou heuristique - le schéma de la démarche décisionnelle varie ; on peut grossièrement les représenter de la manière suivante :

- si elle est algorithmique :



- si elle est heuristique



Dialogue Homme-Machine

La présence ou l'absence des paramètres d'action différencient la procédure heuristique et algorithmique. Dans un problème bien structuré, le problème se résoud de lui-même, sans l'intervention d'un décideur (une fois le critère défini). Si par contre, les variables ne sont pas ou mal connues ou/et si les relations entre ces variables ne sont pas claires, la procédure heuristique apparaît indispensable. La ou les solutions vont dépendre de la valeur accordée aux paramètres d'action et des stratégies utilisées ; ces valeurs sont définies selon les références stratégiques, qui doivent tenir compte de l'environnement du problème et de la manière dont le problème s'insère dans l'ensemble. Le critère utilisé sera passif ou actif suivant qu'il est défini en fonction des stratégies utilisées ou qu'il sert de cadre de référence à l'établissement des stratégies.

Au fur et à mesure de la résolution du problème, les solutions font l'objet de tests :

- quelle est la valeur prise par la fonction-objectif ?
- cette valeur est-elle satisfaisante et acceptable ?
  - dans l'affirmative, la solution de conditionnelle devient inconditionnelle ; si la solution finale est acceptée, la perspective dynamique consistera à resserrer les valeurs des paramètres d'ac-

tion pour approcher la solution optimale ;

. dans la négative, il y a lieu d'analyser les raisons d'incompatibilité de la valeur de la fonction-objectif et celles des paramètres d'action. Selon le résultat de l'analyse, il faudra modifier les stratégies, les plans, les paramètres ou/et revoir le critère.

Section 3 : Stratégies heuristique et dialogue Homme-Machine

Le dernier point soulevé mérite quelques explications. Dire qu'un problème est mal structuré (et doit donc être soumis à une procédure heuristique) signifie qu'il est impossible d'établir

- soit l'inventaire complet des situations dans lesquelles le système peut se trouver
- soit une liste exhaustive des remèdes adéquats à appliquer en face d'une situation particulière. (En fait ces deux points vont la plupart du temps de pair).

Par suite de l'impossibilité de dresser une "matrice" reliant les causes (situations) aux effets (par l'intermédiaire de stratégies) une procédure heuristique DOIT être utilisée. D'un point de vue pratique, il existe des situations-type, auxquelles des stratégies particulières peuvent être appliquées et programmées. Si une situation non programmée survient, le décideur doit intervenir pour y apporter une solution : en supposant l'utilisation d'ensembles électroniques, un véritable dialogue Homme-Machine s'instaure.

- Si un problème consiste à rechercher, parmi un ensemble de possibilités la meilleure solution, la puissance de calcul d'un ordinateur permet une plus large exploration de ces possibilités. Lorsqu'un problème semble trop complexe, on le simplifie en agrégeantles données ou en négligeant certaines contraintes.
- La majorité des problèmes de planning sont dynamiques : la puissance de mémoire et de calcul des ensembles électroniques permet de tenir compte

des interactions dynamiques des décisions, des nouvelles informations, des distorsions dans celles-ci et de leur impact sur le système, sans compter la réduction du délai de réponse.

En fonction de ce qui précède, le rôle de la Machine dans la résolution de problèmes heuristiques se décrit de la manière suivante :

- se substituer à l'homme lors du contrôle du processus, lorsque les éléments de contrôle et de décision sont connus,
- contrôler l'évolution des paramètres identifiés,
- faire appel à l'homme dès qu'une situation non programmée apparaît,
- procurer à l'homme des moyens de calcul et de simulation,
- jouer le rôle d'accumulateur de connaissances : lorsqu'une situation non programmée se présente et qu'une solution lui est donnée par le décideur, celle-ci est programmée ; la réapparition d'une situation analogue ne nécessitera plus l'intervention du décideur.

Avant d'examiner le rôle de l'Homme dans ce dialogue, il nous faut répondre à la question suivante : Pourquoi l'intervention d'un décideur est-elle indispensable ?

- "If planning is just unstructured problem solving, and problems solving is recursive recognition of goal-situation differences and application of operators to reduce the differences, and humans are better recognizers than computers, then effective planning calls for the presence of humans within the process" [1]
- L'homme apparaît donc plus apte à reconnaître la structure d'ensemble d'un problème ; cette "vision" globale et synthétique s'oppose à la démarche purement séquentielle et logique d'un ordinateur.

---

[1] CARROLL D.C. : "Man-Machine Cooperation on Planning and Control Problems" presented at the International Symposium on Long Range Planning for Management, held at UNESCO , Paris, September 20-24 1965.

- Certaines informations sont parfois tellement rarement utilisées, si difficiles à quantifier ou si subjectives, qu'il ne serait pas économique de les insérer dans un programme ou/et de les inclure dans une mémoire.
- Toutes les informations nécessaires, utiles ou significatives ne sont pas nécessairement reconnues comme telles par le modélisateur. La possibilité d'accès à un modèle simple par plusieurs utilisateurs permettra une meilleure adaptation du modèle à la situation réelle ; ceci devient d'autant plus évident lorsqu'on considère les interdépendances entre les diverses fonctions d'une entreprise, fonctions remplies par des personnes différentes et n'ayant pas les mêmes objectifs.

En fonction de ce qui précède, le rôle de l'Homme dans la résolution des problèmes heuristiques se décrit de la manière suivante :

. assister la machine lorsque celle-ci rencontre une situation non programmée,
. adapter les valeurs des paramètres du problème et le critère de décision aux données nouvelles qui ne peuvent pas être prises en charge automatiquement par le système de traitement de l'information,
. consulter les données enrégistrées par le système de traitement de l'information et utiliser ses moyens de calcul pour étayer une prévision ou une décision, lorsque celles-ci ne sont pas programmables.

Section 4 : Nature de la démarche heuristique

La procédure heuristique a pour objectif la restitution de la démarche intellectuelle humaine. Il ne s'agit pas d'établir un modèle intellectuel global ; il s'agit de modéliser certaines propriétés spécifiques et limitées du comportement d'un système intelligent. Suivant que le système tente de modéliser le processus humain de prise de décision, ou qu'il se veuille essentiellement logique (non humain), l'approche sera descriptive ou normative [1].

Les procédures heuristiques fondées sur le comportement humain sont établies à partir de l'observation des mécanismes mentaux mis en jeu par le décideur dans la résolution de problèmes concrets dans des situations concrètes :

- Lorsqu'il ne s'agit que de reproduire les mécanismes de traitement humain de l'information pour que la procédure formalisée réponde comme le décideur à des stimuli identiques, la procédure heuristique est spécifique.
- Lorsqu'il s'agit de formaliser des procédures de décision assez générales pour qu'elles se comportent avec "intelligence" face à une variété de problèmes nouveaux dans des situations nouvelles, les procédures heuristiques sont générales.

En d'autres termes la procédure heuristique est générale lorsque les contours du problème sont circonscrits et indépendants de l'environnement : le jeu d'échecs, tous les jeux de société complexes, la démonstration de théorèmes peuvent être (et ont été) soumis à ce genre de procédures [2]. Lorsque, par contre la procédure est soumise aux caractéristiques de l'environnement et que celui-ci intervient soit sous la forme de contraintes soit par la nécessité de modifier des stratégies, la procédure sera spécifique à un problème déterminé.

Parmi les mécanismes mentaux qui ont déjà été sélectionnés et modélisés

---

[1] Partant de l'hypothèse que l'homme n'est pas nécessairement l'être le plus intelligent qui soit, des chercheurs ont tenté de modéliser le processus d'une intelligence abstraite ; les procédures qui résultent d'une telle approche sont normatives parce qu'elles correspondent au comportement que le décideur devrait adopter et non à celui qu'il manifeste en réalité.

[2] NEWELL A., SHAW J.C. et SIMON H. : "A General Problem Solving Program for a Computer" ; Computers and Automation VIII, juillet 1959, p. 10 - 17.

on peut citer :

- l'analyse des liens de cause à effet : par le cheminement dans un arbre de décision, par évaluation successive des branches à chaque noeud,
- la formation de concepts,
- la reconnaissance des formes,
- la magasinage de l'information (organisation associative en mémoire).

Nous terminerons cette section par deux remarques :

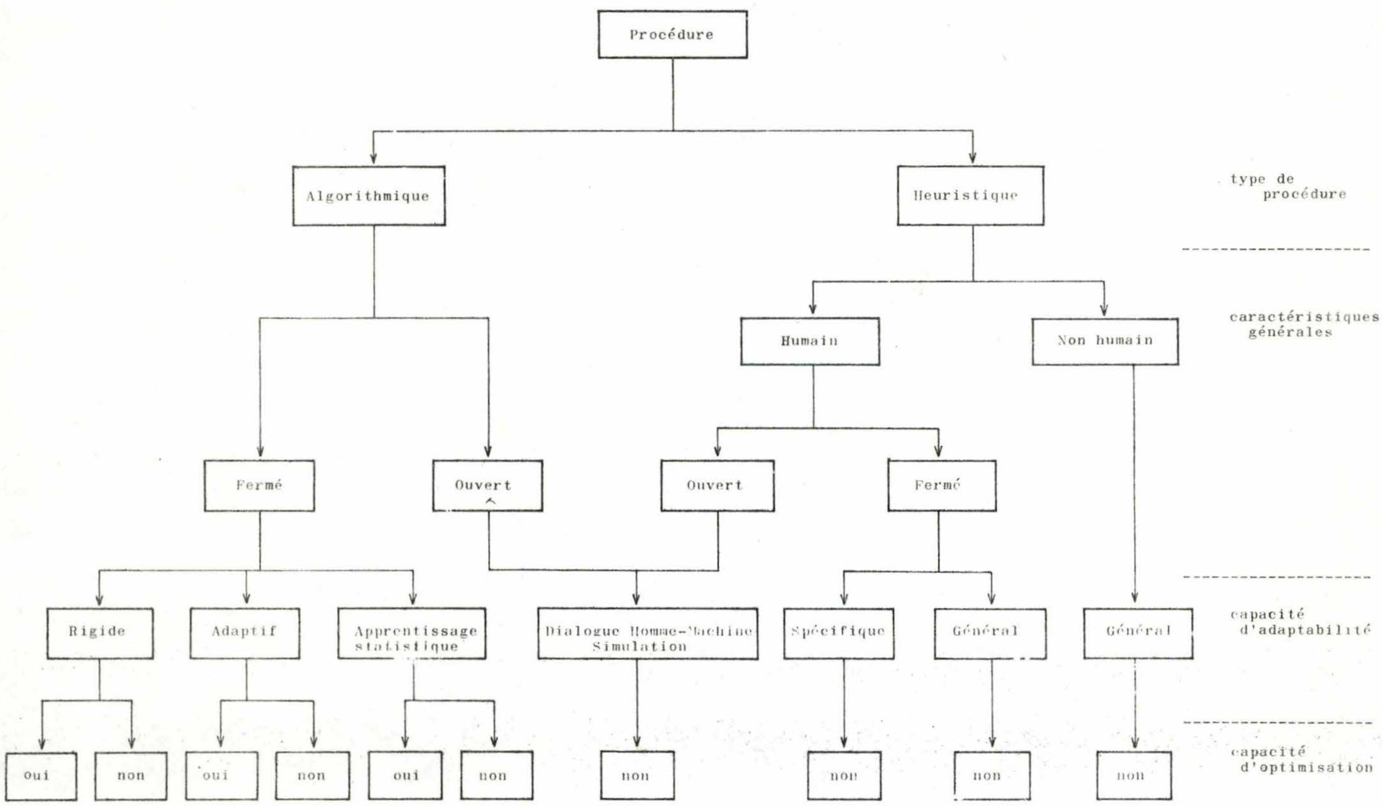
- Si la formalisation des procédures simulant le comportement humain permet d'atteindre des vitesses de traitement plus grandes, d'obtenir une plus grande exactitude, d'assurer une plus grande cohérence et stabilité dans le raisonnement, il est néanmoins important de souligner que leurs performances ne sont pas supérieures à celles du décideur dont on a observé le comportement.
- Pour apaiser les craintes des uns et modérer les transports des autres, il n'est pas question de supposer - même à longue échéance - l'existence de "cerveaux"électroniques. Qu'il s'agisse de tests logiques ou de modèles plus complexes, les modèles électroniques ne peuvent prétendre décrire nos délibérations cérébrales. La raison en est fort simple : un modèle suit "un raisonnement" à partir d'une situation donnée, fixée dans le temps et l'espace. L'esprit humain semble procéder de façon plus synthétique en portant d'abord un jugement d'ensemble sur la situation. Il en retire une impression, décide empiriquement d'une première solution quitte à étudier ensuite quelques variantes de celle-ci". [1] Il n'y aura pas plus dans un modèle électronique que ce que l'homme aura bien voulu y mettre, si ce n'est dans l'explicitation de liens logiques entre des éléments qui apparemment n'en révélaient pas.

---

[1] STENGEL J. : "Recherche dans le domaine des processus heuristiques de décision", article paru dans les "Actes du 2ème Congrès international de Recherche Opérationnelle, organisé à Aix en Provence en 1960 par The International Federation of Operational Research Societies" ; Dunod 1961, p. 72-81.

Section 5 : Schéma des modèles selon la procédure utilisée

Nous avons défini un grand nombre de modèles et nous les avons classés selon la procédure à mettre en oeuvre (heuristique ou algorithmique). Nous voudrions synthétiser tout cela en un seul tableau : au premier niveau de la hiérarchie, la nature de la procédure, au deuxième, les caractéristiques générales des modèles, au troisième, leur capacité d'adaptabilité et au dernier leur capacité d'optimisation.



Chapitre 4 :

Application de la procédure heuristique aux problèmes d'ordonnancement

Section 1 : Critique de procédures, considérées comme heuristiques

Si la plupart des auteurs sont d'accord sur la définition des problèmes heuristiques, la manière de considérer les méthodes heuristiques varient, du moins en matière d'ordonnancement.

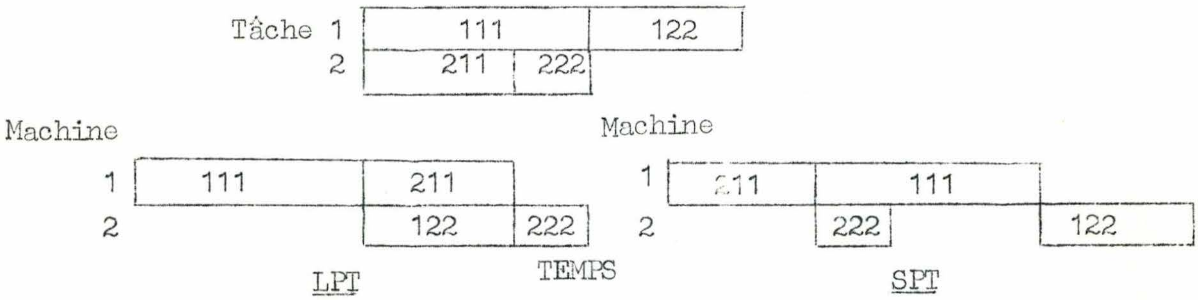
Certains estiment que la difficulté de trouver des résultats satisfaisants provient des règles utilisées. Ils utilisent donc des règles de décision complexes, combinaisons de règles simples affectées de coefficients de pondération ou/et de probabilité. D'autres comme Fisher et Thompson [1] se base sur le fait que dans certaines situations une règle est préférable à une autre. Partant des règles SPT et MWKR, ils développent un programme auquel est associé un système de probabilités : ce système se modifie selon le succès relatif rencontré dans le passé par l'application de l'une ou de l'autre règle. Nous ne considérons pas ces démarches comme heuristiques et ce pour les raisons suivantes :

- Dans l'annexe V, inventoriant certains travaux effectués dans le domaine de la simulation, nous avons abordé le modèles probabilistes (Bakhru, Rao...). Cette simulation ne fait que tempérer les effets de l'emploi univoque et automatique d'une règle de décision ; nous admettons l'intervention de probabilités mais nous réfutons l'argument selon lequel une règle doit être appliquée d'autant plus qu'elle s'est avérée, par ailleurs et dans le passé, plus efficace.

---

[1] FISCHER H. et THOMPSON G.L. : "Probabilistic learning Combinations of local Job-Shop scheduling Rules" dans MUTH J.F. et THOMPSON G.L. op. cit. chapitre 15.

- Le choix des règles est laissé à la discrétion du décideur. S'il est exact que, suivant le critère utilisé, certaines règles s'avèrent "statistiquement" meilleures que d'autres, nous prétendons qu'il est arbitraire de se limiter à la combinaison de deux règles. La règle SPT n'a de bonnes références qu'utilisée globalement (c-a-d pour l'établissement d'un ordonnancement dans son entièreté) et non pas lorsqu'elle est utilisée localement. Même lorsque le critère est min F_{max}, la règle devant statistiquement donner le plus mauvais résultat, la règle LPT en l'occurence peut dans certains cas s'avérer optimale, même utilisée globalement. Exemple :



- Ces critiques veulent aboutir à celle qui nous semble fondamentale : la démarche heuristique doit se développer au fur et à mesure que le problème se résoud et en fonction du problème particulier à résoudre. La démarche heuristique doit être entreprise sans préjuger de la valeur d'une règle ; elle consiste à porter toute son attention sur la situation particulière qui se présente.

Or, la simulation, même tempérée par des règles probabilistes est appliquée quel que soit le problème. La démarche heuristique consiste essentiellement dans une analyse de situation. Les auteurs cités le reconnaissent : le modèle psychologique de "récompense-punition" résulte par essence de l'expérience; lrosqu'on est confronté avec une nouvelle situation, une autre action est entreprise et le résultat gardé en mémoire. Quand une situation similaire apparait, une autre action est prise et évaluée en fonction des résultats antérieurs. Les deux actions sont identiques ou différentes, mais si la dernière action est plus efficace, on

augmente

la probabilité de répéter cette action dans l'avenir ; sinon cette probabilité baisse. Les hommes tendent à renforcer les actions associées à des résultats satisfaisants et s'écarter de celles dont les résultats ne le sont pas.

Cette manière de voir l'heuristique est fort fréquente et se présente sous le nom d'apprentissage ou de "learning process". Nous ne nions pas son intervention mais nous estimons d'une part qu'il est mal appliqué par l'emploi de règles probabilistes (du moins en matière d'ordonnancement) et d'autre part qu'il est trop présomptueux :

- les situations sont souvent non comparables ; nous le verrons plus tard ; cette théorie suppose que l'on puisse affirmer que deux situations sont identiques ou similaires ; or, nous avons dit que la démarche heuristique était indispensable du fait de l'impossibilité de tenir compte de toutes les situations particulières et de tous les éléments qui sont à l'origine de cette situation.
- les problèmes à résoudre sont presque tous de nature dynamique ; l'introduction du facteur temps rend cette description des situations encore plus aléatoire.
- d'après Fisher et Thompson, la situation se juge au niveau de l'ordonnancement lui-même. Dans la perspective de l'entreprise, système intégré, un ordonnancement se juge par son impact sur l'environnement : minimiser la durée totale d'occupation de l'atelier peut s'avérer un objectif valable au niveau de l'ordonnancement mais secondaire ou incompatible lorsqu'on s'aperçoit qu'il entraîne des retards importants, qu'il nécessite des capacités de stockage supplémentaires, qu'il dépasse la courbe de charge des moyens de production etc...

Section 2 : Vers une conception de l'heuristique appliquée à l'ordonnancement

Quels sont les apports de l'approche précédente ?
Elle nous a permis de mettre l'accent sur les points suivants, sur lesquels

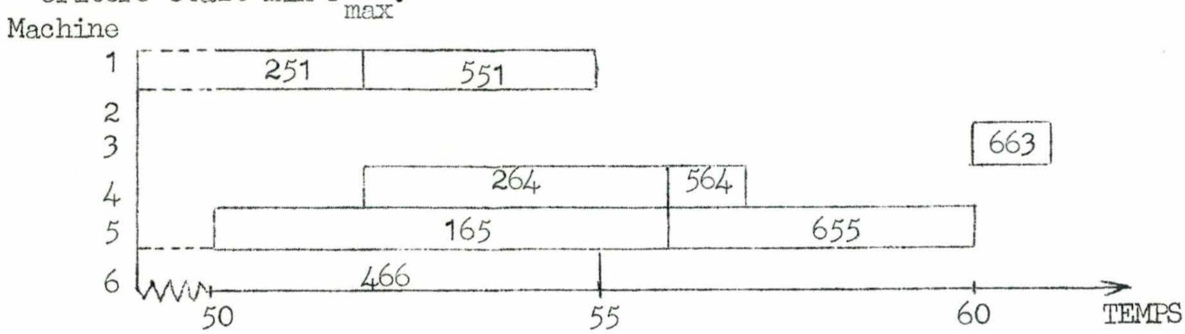
nous nous baserons dans la suite :

- une règle ne peut pas être considérée comme bonne ou mauvaise, lorsqu'elle est utilisée localement.
- la valeur d'un ordonnancement doit se juger non au niveau de l'ordonnancement (en fonction des possibilités) mais au niveau de l'entreprise (en fonction des besoins).
- s'il existe une confusion largement répandue entre la procédure heuristique et probabiliste (apprentissage), cela ne signifie pas que nous réfutions l'intervention de probabilités ; cependant ce n'est pas parce qu'une règle a donné de bons résultats dans le passé qu'il faut l'appliquer presque de façon automatique dans le futur : en matière d'ordonnancement les situations identiques ou même analogues n'existent pas, du moins lorsque le problème n'est pas trop simple.
- La distinction entre l'emploi automatique d'une règle et son utilisation suivant la situation particulière est la même que celle qui existe entre le "prêt à porter" et le "fait sur mesure". Notre attitude devant un problème d'ordonnancement doit donc être neutre et objective. Citons à ce propos le travail de Burstall [1] qui a formalisé le comportement d'un homme résolvant un problème d'ordonnancement ; il n'a pas cherché à savoir pourquoi telle opération devait être placée devant une autre (rapports logiques) mais ce qui avait amené l'homme observé à ordonnancer telle opération à tel endroit. Il arrive souvent qu'une situation s'améliore à la suite d'une permutation sans qu'il soit possible de suivre le raisonnement logique aboutissant à cette permutation. L'exemple qui suit témoigne de la quasi-impossibilité de décrire toutes les situations, de trouver pour chacune d'elles une solution adéquate et enfin de les programmer.

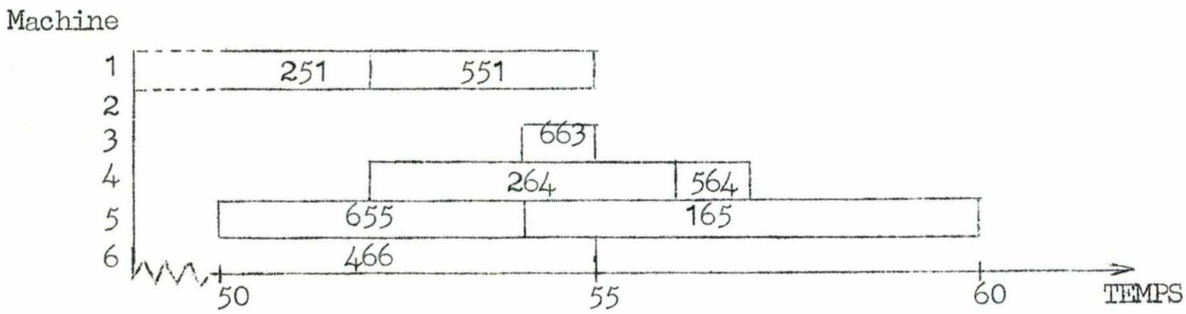
---

[1] BURSTALL R.M. :"A Heuristic Method for a Job-Shop Scheduling Problem" Operational Research Quarterly; septembre 1966 p. 291.

Nous avons résolu un problème cité par Fisher et Thompson. L'application de la règle MWKR donnait l'ordonnancement suivant (à partir du moment 50). Le critère était min F_{max}.



Nous nous sommes aperçus (par hasard) que la permutation de l'opération 165 et 655 avait pour effet de réduire F_{max} d'une unité de temps. Or, il n'y avait aucune raison apparente de le faire puisqu'il reste à la tâche 1,6 heures de traitement à subir et 5 heures pour la tâche 6. La tâche 1 méritait donc bien une priorité légèrement supérieure.



En cherchant les raisons de cette amélioration, nous avons décelé certaines circonstances favorables à la permutation, sans pour autant être certain qu'elles soient les seules ni que ce soit effectivement l'une ou plusieurs d'entre elles qui nous ont poussé à intervertir ces deux opérations :

- la machine 3 est inoccupée, ce qui permet de fixer 663 n'importe où du moment que cette opération suive 655 ;
- bien qu'elle soit plus longue, l'opération 165 est la dernière de la tâche 1, ce qui n'est pas le cas de la tâche 6;
- la permutation de ces deux opérations ne modifie pas l'ordonnancement

des autres opérations ;

- la valeur de F_{max}, quelle quesoit la permutation choisie, sera déterminée par la tâche 1 ou 6. Il aurait suffi que 564 ait une durée de traitement de 5 heures pour qu'il n'en soit plus ainsi.

La seule chose que nous puissions dire, c'est que la permutation a été effectuée par notre faculté de "visualisation" d'un ordonnancement dans son entièreté. Cet exemple illustre aussi le fait qu'aucune règle n'est bonne ou mauvaise en soi (à supposer que l'on puisse trouver la règle auquel obéit le dernier ordonnancement).

S'il semble impossible de disséquer en une suite de rapports logiques le processus du raisonnement humain (principe de base de la démarche heuristique), il est toutefois possible d'utiliser certaines caractéristiques de celui-ci. Les principales en sont la :

- projection : effets à plus ou moins long terme de l'utilisation d'une stratégie ou d'une règle ;
- mémoire ou association d'idées : faculté de l'esprit humain de rapprocher une situation d'une autre, rencontrée antérieurement ;
- simplification et décomposition : lorsqu'un problème est trop complexe, on le subdivise en problèmes plus simples. Simon [1] y voit une raison supplémentaire d'instaurer le dialogue Homme-Machine. C'est ainsi qu'il décrit le modèle de résolution d'un problème : "Problem solving by erecting goals, detecting differences between present situation and goals, finding in memory or by search, tools or processes ("operators") that are relevant to reducing differences of these particular kinds, and applying these tools or processes. Each problem generates subproblems until we find a subproblem we can solve - for which we already have a pro-

---

[1] SIMON H.A. : "The New Science of Management Decision" Harper and Bros., New York, 1960, p. 27 cité par CARROLL D.C., op. cit.

gram (i.e. algorithm) stored in memory. We proceed until, by successive solution of such subproblems, we eventually achieve our over-all goal - or give up".

- Reconnaissance des formes ou visualisation : faculté synthétique de l'esprit humain, permettant de juger globalement d'une situation. Cette faculté est très importante en matière d'ordonnancement et comme nous le signalions, c'est sans doute elle qui nous a poussé à permuter les deux opérations.
- abstraction temporaire : cette faculté consiste à faire abstraction ou à transformer fictivement et temporairement des données, des contraintes ou le valeur des paramètres, dans le but de simplifier le problème ou de juger de l'impact des contraintes ou paramètres sur le système.
- ...

Malgré tout ce qui précède quant à la démarche et à la procédure heuristique, il pourrait subsister des doutes quant à leur véritable signification. S'agit-il d'une analyse de situation ou de la transposition formalisée de caractéristiques de l'esprit humain ? Nous ne croyons pas que ces deux définitions soient fort distantes l'une de l'autre ; c'est ce que nous allons essayé de montrer dans les sections suivantes, sur base d'un exemple. Qu'il nous suffise, à ce stade de dire ceci :

- l'aspect "analyse d'une situation" met davantage l'accent sur le côté conceptuel et théorique d'un problème mal structuré, tandis que l'aspect "formalisation de caractéristiques de l'esprit humain" insiste plus sur les moyens pratiques de résolution de ce problème. Autrement dit, dans le dialogue Homme-Machine impliqué par la nature du problème heuristique, le premier aspect insiste sur l'intervention de l'homme, le second sur celui de la machine ;
- si, la formalisation de toutes les caractéristiques de l'esprit humain était possible, une situation pourrait être complètement analysée et l'intervention de l'homme dans le dialogue inutile ;

- la pauvreté des recherches dans ce domaine nous force à ne pouvoir appliquer qu'un nombre restreint (second aspect) de règles heuristiques (section suivante).

Section 3 : Application de l'heuristique, formalisation de caractéristiques de l'esprit humain.

Les seuls résultats d'application de règles heuristiques aux problèmes d'ordonnancements, nous les avons trouvé chez Gere [1]. Malgré nos efforts nous n'avons pas pu prendre connaissance des données et du programme qu'il avait élaboré ; aussi, est-ce sur base de sa description de certaines règles heuristiques, que nous avons établi un programme (chapitre suivant).

§ 1. Principe

Il y a deux manières d'approcher un problème d'ordonnancement : l'approche générale consistant dans une application systématique d'une règle de priorité et l'approche spécifique (le "fait sur mesure") tenant compte des particularités du problème à résoudre. Gere utilise une approche intermédiaire consistant dans l'utilisation de bonnes règles de priorité en les "modelant" en fonction du problème particulier et en faisant appel à des règles heuristiques.

Le critère choisi par Gere est la minimisation de la somme des retards. Comme il n'existe pas de résultats sur les mérites relatifs des règles de priorité, lorsque le critère consiste à rencontrer les dates de livraison, le choix des règles est arbitraire. Cependant, il est vraissemblable que des règles tenant compte, d'une manière ou d'une autre, des dates de livraison et/ou les durées de traitement aboutissent à de meilleurs ordonnancements que les règles aléatoires. Nous verrons - et

---

[1] GERE W.S. : op. cit.

cette conclusion est très importante - que la détermination d'une "bonne" règle de base est relativement peu importante.

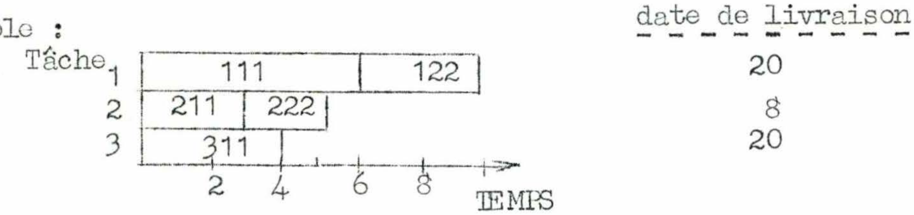
§ 2. Choix des règles heuristiques

Gere les choisit en prenant note des règles intuitivement utilisées lorsqu'on essaie de tracer un ordonnancement et en demandant aux praticiens leur avis. Il décrit 8 règles et il a testé 4 d'entre elles ; ce sont ces quatre dernières que nous avons choisies.

a. L'heuristique d'alternance

Le raisonnement à la base de cette heuristique est le suivant : la règle de priorité sélectionne, en cas de conflit de passage, une seule opération. L'heuristique d'alternance effectue un contrôle pour s'assurer que l'ordonnancement de cette opération n'entraîne pas le retard d'une autre tâche dont une des opérations est ordonnançable. S'il en ainsi, on ordonnance cette dernière opération à la place de la première et un nouveau contrôle des retards est effectué. Selon le résultat (somme des retards) on ordonnancera l'une ou l'autre opération. Dans le cas où plusieurs opérations ordonnançables sont en retard, on choisit celle dont le retard est le plus important.

Exemple :



Si la règle de priorité consiste à choisir l'opération dont le numéro d'identification de la tâche est le plus petit (règle aléatoire), l'opération 111 sera ordonnancée au moment 0. Comment avons-nous estimé les retards minima entraîné par cette décision ? :

- tâche 1 : l'opération suivante à ordonnancer (122) est à traiter sur la machine 2, qui est à priori inoccupée. (elle pourrait le devenir avant la fin du traitement de l'opération 111) au moment où l'opération 111 s'achève (au moment 6). Il reste, si 111 est ordonnancé, 3 heures de traitement à subir par la tâche 1. Le retard minimum (l'avance maximum) de cette tâche est donc de :

$$20 - \max(6,0) - 3 = 12$$ [1]

- tâche 2 : la contrainte de routing s'exprime par 0, puisqu' aucune opération de cette tâche n'a encore été ordonnancée ; par contre 211 ne pourra commencer au plus tôt qu'au moment 6, puisque 111 est traité pendant cette période (contrainte disjonctive). La tâche 2 doit encore subir 5 heures de traitement. Le retard minimum de cette tâche s'élève donc à :

$$8 - \max(0,6) - 5 = -3$$

- tâche 3 : même raisonnement que pour la tâche 2 : l'avance (retard) s'élève à

$$20 - \max(0,6) - 4 = 10$$

Il ressort de ces estimations que l'ordonnancement de l'opération 111 entraîne le retard de la tâche 2 ; somme des retards : 3 ; somme des avances : 21 ; somme des écarts : 18. Appliquons l'heuristique d'alternance sur base de la tâche 2 (en retard) : si 211 est ordonnancé les avances (retards) sont les suivantes :

- tâche 1 : $20 - \max(0,3) - 9 = 8$
- tâche 2 : $8 - \max(3,0) - 2 = 3$
- tâche 3 : $20 - \max(0,3) - 4 = 13$

---

[1] Il s'agit en fait d'une variante de l'application de la règle du temps restant (qui ne tient compte que de la contrainte de routing). Le premier terme du maximum exprime cette contrainte : date au plus tôt à laquelle une opération ordonnançable d'une tâche peut commencer ; le second terme exprime la contrainte disjonctive : date à partir de laquelle la machine qui doit traiter l'opération ordonnançable d'une tâche est inoccupée

La somme des retards est nulle, la somme des avances de 24 et la somme des écarts de 24. Puisque l'ordonnancement de l'opération 211 n'entraîne aucun retard, et que celui de l'opération 111 (choisie par la règle de priorité) en entraîne , l'heuristique s'applique et 211 sera effectivement ordonnancé.

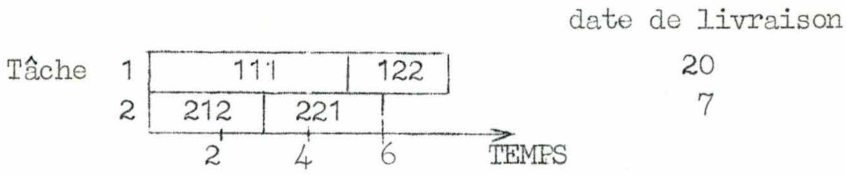
Supposons un instant que la date de livraison de la tâche 2 était de 12. L'ordonnancement de 211 n'aurait entraîner aucun retard et la somme des avances (écarts) s'élevait à 21. D'après Gere, 111 devait être ordonnancé. Nous avons essayé d'élargir l'heuristique d'alternance en l'appliquant également lorsqu'aucun retard n'était subi par l'ordonnancement de l'opération choisie par la régle de priorité, en effectuant un contrôle sur base de l'opération dont l'avance était la moins élevée (211). Nous obtenions une somme des avances (écarts) de 28 ; dans ce cas là, nous avons ordonnancé 211. Nous reviendrons sur ce point lorsque nous examinerons les résultats de l'exemple traité. (chapitre suivant)

b. L'heuristique de projection

Dans tout ordonnancement, la question qui se pose est la suivante : lorsqu'il y a conflit de passage entre plusieurs tâches sur une même machine, que se passe-t-il si telle opération est ordonnancée ? Si on pouvait anticiper tous les conflits possibles relatifs à l'ordonnancement d'une opération particulière, on aboutit rapidement à l'ordonnancement optimal. Malheureusement, la perception de tous les conflits entraînerait des itérations presque exhaustives de tous les ordonnancements possibles. A défaut de pouvoir faire de la projection à long terme, nous effectuerons de la projection à court terme.

Lorsqu'une opération est choisie par la règle de priorité, on se pose la question suivante : existe-t-il une opération plus critique [1]. que celle choisie par la règle, susceptible d'atteindre la même machine avant que l'opération prévue ne soit terminée ? (s'il y a plusieurs opérations critiques, on prendra celle qui l'est le plus). Si la réponse est affirmative, on ordonnance cette opération et on contrôle l'impact de cet ordonnancement sur les retards. Selon les résultats, on ordonnancera l'opération prévue par la règle de priorité ou celle prévue par l'heuristique.

Exemple :



Quel que soit la règle de priorité choisie, on aboutira au même ordonnancement : 111 - 221 sur la machine 1 et 212 - 122 sur la machine 2.[2] Les avances des tâches sont respectivement de 12 et de - 1. Puisque la tâche 2 est en retard, appliquons l'heuristique ; vérifions d'abord les conditions d'application :

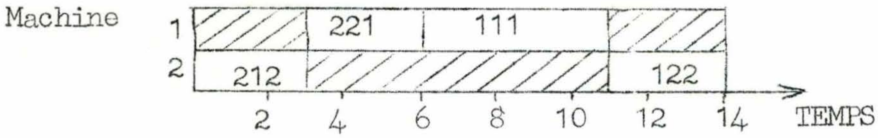
- il existe une tâche (2) dont l'opération suivante non ordonnancée (221) pourrait commencer avant que l'opération en traitement (111) sur la machine 1 ne soit terminée ;
- l'avance de la tâche (1), dont une opération (111) est ordonnancée selon la règle est plus importante (12) que celle de la tâche (2) dont une opération (221) est susceptible d'atteindre cette machine (-1)

L'effet de l'ordonnancement de l'opération 221 est le suivant : la tâche 1 aura une avance de 6 et la tâche 2 de 1. L'heuristique de projection doit donc s'appliquer et donne le résultat suivant :

---

[1] Une opération est "critique" lorsqu'elle est en retard ou que son avance est inférieure à un niveau donné. Notre niveau critique est zéro, ce qui signifie que nous ne nous sommes occupés que des tâches en retard.

[2] Cette affirmation est exacte dans la mesure où les ordonnancements engendrés sont de type non-retardé.



Remarques :

- Comme pour l'heuristique d'alternance, nous avons élargi cette heuristique en effectuant également un contrôle lorsqu'à la suite de premier ordonnancement, aucun retard n'était subi ; dans ce cas, il suffit qu'une des tâches ait une avance moins élevée que celle prévue par la règle, pour que l'heuristique soit testée.
- L'utilisation de cette heuristique change le type d'ordonnancement : de non-retardé, il devient actif, des machines pouvant rester inoccupées alors que des opérations sont ordonnançables sur ces machines.
- Malgré son efficacité, cette heuristique doit s'appliquer avec une très grande prudence, car elle peut entraîner le retard d'autres tâches (nous nous sommes occupés d'une tâche "critique" mais non pas de la somme des retards ; il suffit de supposer, dans notre exemple que d'autres opérations étaient ordonnançables sur la machine 1, au moment 0). C'est pour cette raison, que l'heuristique de projection ne sera efficace que si une double condition est remplie :
    - l'opération de la tâche différée (111) n'est pas en retard ;
    - la nouvelle valeur de l'avance de la tâche différée (1) doit être supérieure à deux fois celle de la tâche dont une opération est ordonnancée par l'heuristique (6 est supérieur à 2x1)

c. L'heuristique d'insertion

Si l'heuristique de projection tient compte d'un retard éventuel de la tâche dont une des opérations est différée, comment tient-elle compte des autres tâches dont certaines opérations sont en attente devant cette machine (c-a-d sont ordonnançables au même moment) ?

La seconde condition ci-dessus répond d'une certaine manière à la question, en donnant une certaine idée de la durée pendant laquelle une opération sera différée. Cependant elle ne suffit pas et il est même possible que cette heuristique n'aboutisse pas à de meilleurs ordonnancements. Heureusement elle entraîne toujours un laps de temps pendant laquelle une machine est inoccupée (la machine 1 est libre du moment 0 au moment 3). L'heuristique d'insertion revient à examiner si une opération ordonnançable n'est pas susceptible de remplir cet intervalle de temps. Si dans l'exemple ci-dessus, nous avions eu une opération (311) dont la durée de traitement était inférieure à 4, elle aurait pu être insérée ; si plusieurs opérations sont dans le même cas, on choisira celle dont le retard est le plus important ou celle dont la durée de traitement est la plus longue.

d. L'heuristique de réajustement des dates de livraison

Lorsqu'un ordonnancement est terminé et qu'il est insatisfaisant, on peut en établir un autre basé soit sur une autre règle de priorité, soit sur la même en modifiant les priorités des tâches. Cette modification se fait de la manière suivante : lorsqu'un ordonnancement est terminé et qu'une ou plusieurs tâches sont en retard, or recommence l'ordonnancement en attribuant à ces tâches une date de livraison fictive, égale à la date de livraison réelle diminuée du montant du retard. Cette façon de faire a pour effet d'augmenter les priorités des tâches en retard. Cette heuristique ne peut pas être recommencée plusieurs fois car une situation de "Rob Peter to Pay Paul" pourrait s'ensuivre.

Nous n'avons pas testé cette heuristique car elle se rapproche de la seconde partie de notre programme (section suivante). Signalons cependant que, d'après Gere, le second ordonnancement est légèrement meilleur que le premier mais la différence n'est pas significative.

§ 3. Résultats et conclusions de Gere

Gere testa 25 problèmes statiques : le nombre de tâches variait de 6 à 60, le nombre de machines de 4 à 6, le nombre d'opérations par tâche de 1 à 16 et les durées de traitement de 1 à 10.

Il utilisa les règles de priorité suivantes :

1. Temps restant : le temps restant d'une tâche est la différence, à un moment donné entre sa date de livraison et la somme des durées de traitement des opérations restant à effectuer de cette tâche ; si cette différence est négative, la tâche sera certainement en retard.
2. Temps restant par opération : il s'agit du nombre d'heures disponibles avant la date de livraison, divisé par le nombre d'opérations restant à effectuer ; si la tâche est nécessairement en retard (temps restant négatif), la valeur de priorité de cette tâche est égale au montant du retard.
3. Rapport de temps restant : il s'agit du nombre d'heures disponibles avant la date de livraison divisé par le nombre d'heures restant jusqu'à la date de livraison; si elle est nécessairement en retard (temps restant négatif), la valeur de priorité de cette tâche est égale au montant du temps restant.
4. Combinaison de la règle du rapport de temps restant et de la règle de la plus courte opération (SPT) : si une tâche est en retard, sa valeur de priorité est égale au montant du temps restant ; l'ensemble des autres tâches est divisé en deux sous-ensembles :
   - le sous-ensemble des tâches dont le rapport de temps res-

tant est inférieur à deux fois celui de la tâche ayant le rapport de temps restant minimum ; les tâches appartenant à ce sous-ensemble seront classées selon la règle SPT ;

- le sous-ensemble des autres tâches dont le classement est sans importance.

5. SPT : règle de la plus courte opération.
6. Premier arrivé, premier servi (FCFS) : est prioritaire l'opération dont la durée d'attente devant une machine est la plus grande.
7. Règle aléatoire : est prioritaire l'opération de la tâche dont le numéro d'identification est le plus petit (par exemple).

Gere aboutit aux conclusions suivantes, lorsque le critère consiste dans la minimisation de la somme des retards :

1. Une règle qui n'utilise pas les données du problème (dates de livraison, durées de traitement, nombre d'opérations...) est moins efficace qu'une règle qui en tient compte. Les deux dernières règles citées sont de ce type : bien que la règle du premier arrivé, premier servi soit légèrement supérieure à la règle aléatoire, la différence d'efficacité entre elles n'est pas significative ;
2. Malgré les apparences, la règle 1 est légèrement supérieure à la règle 2, même lorsque le nombre d'opérations varie d'une tâche à l'autre.
3. Malgré les apparences, la règle 1 est légèrement supérieure à la règle 3, même lorsque les dates de livraison varient d'une tâche à l'autre ;
4. La règle 1 est significativement plus efficace que la règle 5. SPT s'avère supérieur lorsqu'on désire que les tâches soient terminées le plus rapidement possible mais lorsque des dates de livraison sont imposées, les tâches possédant des opérations lon-

gues peuvent être différées pendant très longtemps.

5. La règle 4 est significativement plus efficace que la règle SPT mais elle est inférieure à la règle 1 ;
6. L'heuristique d'alternance utilisée seule : cette heuristique s'est avérée efficace ; dans un cas seulement l'application systématique de la règle aboutissait à un meilleur ordonnancement ;
7. L'heuristique de projection et d'insertion utilisées seules : en comparant les ordonnancements obtenus par ces heuristiques avec ceux obtenus grâce à l'heuristique d'alternance, les retards furent réduits dans 15 cas, égaux dans 7 autres et légèrement supérieurs dans les trois derniers. L'hypothèse d'efficacité de cette heuristique est donc fortement confirmée ;
8. L'utilisation conjointe de plusieurs heuristiques s'avère toujours meilleure que l'utilisation de l'une d'entre elles.

Un point intéressant soulevé par Gere concerne le comportement des règles de priorité lorsqu'elles sont renforcées par des heuristiques :

- si on effectue une hiérarchie d'efficacité entre les règles lorsqu'elles ne sont pas accompagnées d'heuristiques, on s'aperçoit qu'il n'y a pas de corrélation entre ce classement et celui qui serait effectué dans le cas où règles de priorité sont accompagnées d'heuristiques. Cela signifie qu'une règle utilisée seule peut s'avérer "mauvaise" et qu'associée à une ou plusieurs heuristiques, elle s'avère "bonne" ;
- parallèlement, Gere vérifia l'hypothèse selon laquelle les règles sont approximativement d'efficacité égale lorsqu'elles sont soutenues par les heuristiques d'alternance et/ou de projection.
- chaque règle devient plus efficace lorsqu'elle est accompagnée d'heuristiques.

§ 4. Romarques

- L'emploi de règles heuristiques ne se limite pas au cas où le critère concerne les retards ; bien que certaines soient spécifiques à ce critère, il aurait suffi d'introduire de légères modifications pour les appliquer à F_{max} par exemple.

- L'emploi d'heuristiques diminue sensiblement le coût d'établissement d'un ordonnancement. D'après Gere, l'introduction dans un programme des heuristiques décrites dans un programme accroît la durée de calcul de 10 à 20 %. Ce coût supplémentaire est largement compensé par la valeur de l'ordonnancement lui-même (dans notre cas, minimisation des retards). D'autre part, comme il semble prouvé qu'une règle "pauvre" avec heuristiques est meilleure qu'une "bonne" sans heuristiques et que les règles heuristiques réduisent les différences d'efficacité entre les règles de décision, il suffirait d'établir un seul ordonnancement, basé sur une règle simple, non aléatoire, qui tienne compte du critère utilisé et qui soit accompagnée de toutes les heuristiques.

- Nous avons résolu un problème d'ordonnancement simple, dont les résultats se trouvent dans le chapitre suivant. Certaines conclusions de Gere sont vérifiées ; si elles ne sont pas toutes confirmées, cela provient de ce que notre exemple est unique d'une part et qu'il est trop simple de l'autre (le nombre de conflits n'est pas suffisant).

- Il reste un problème à résoudre, celui du critère : Comment résoudre un problème d'ordonnancement lorsque plusieurs critères sont utilisés ? Nous allons tenté de répondre à cette question dans la section suivante. Au lieu de mettre l'accent sur les heuristiques en tant que méthodes de calcul, nous insisterons sur l'heuristisque en tant que nature d'un problème particulier : la notion d'heuristique y apparaît comme une analyse de situation, devant tenir compte des multiples aspects affectant le bon fonctionnement d'une entreprise (la production, plus particulièrement).

Section 4 : Application de l'heuristique, analyse de situation

Dans la section précédente, nous insistions sur l'aspect des procédures heuristiques applicables à l'ordonnancement. Nous voudrions insister ici sur l'heuristique en tant que nature particulière d'un problème : c'est parce qu'il est mal structuré qu'il est heuristique. La limite essentielle des règles heuristiques utilisées dans la section précédente résidait dans la fonction-critère : dans les deux premiers chapitres, nous critiquions "la définition du critère en fonction des possibilités et non en fonction des besoins". Nous allons essayer de sortir de cette impasse par une méthode qui nous a été suggérée par le professeur F. Bodart.

§ 1. Principe

La fonction-critère doit être définie dans le cadre des références stratégiques et des stratégies (cfr. chapitre 1 et 2) qui sont édictées à un niveau supérieur de décision. Ce critère doit refléter un objectif général, tenir compte de l'entreprise dans son ensemble. Pour être efficace, un ordonnancement doit dépendre et rejaillir sur tout son environnement. L'impossibilité de formaliser les interdépendances entre les variables (caractéristiques du problème heuristique) empêche la définition d'un critère. On pourrait éviter cette difficulté de la manière suivante : il suffit que la direction générale (ou le niveau supérieur) exprime des souhaits plus ou moins stricts, que ces souhaits soient considérés par les responsables de l'ordonnancement comme des contraintes et que ces responsables trouvent un ordonnancement satisfaisant ces souhaits. [1]

---

[1] Cette approche par système doit être mise en relation avec la première partie (particulièrement le chapitre 3) ainsi qu'avec les trois premiers chapitres de la troisième partie.

§ 2. Procédure pratique

a. Les souhaits, reflets des objectifs de l'entreprise peuvent, entre autres affecter les points suivants :

- la clientèle : garder la satisfaction du client en ne subissant aucun retard ou en lui livrant sa commande plus tôt que prévu ;
- la production : garder les stocks intermédiaires dans certaines limites, équilibrer les courbes de charge, garder le nombre d'arrêts-machine ou/et le nombre d'heures d'inoccupation-machine en dessous d'un certain niveau ... ;
- le temps : spécifier que toutes les tâches doivent être terminées avant une date fixée à priori, limiter la durée moyenne d'occupation de l'atelier à une certaine valeur ... ;
- prix de revient : il peut exister des écarts importants entre le prix de revient standard, déterminé à priori, en fonction de la filière utilisée, des conditions de l'ordonnancement (et de l'environnement en général) et le prix de revient réel.

b. Ces souhaits, exprimés sous la forme de limites à ne pas enfreindre sont reprises par les responsables de l'ordonnancement qui devront les prendre comme des contraintes. Le problème d'ordonnancement revient alors à en trouver un (ou plusieurs) qui satisfait ces normes. Remarquons qu'il ne s'agit pas d'un ordonnancement optimal parce que :

- s'il l'était, la solution serait unique ; s'il y en avait plusieurs, elles devraient avoir exactement la même valeur ;
- il supposerait une commune mesure entre les limites fixées à priori (Nous reviendrons sur cette remarque).

c. Une fois les souhaits introduits comme limites on comme paramètres de contrôle, l'ordonnancement commence sur base d'une règle quelconque, accompagnée ou non de règles heuristiques. Avant de prendre une décision définitive concernant l'ordonnancement d'une opération on effectue un contrôle pour s'assurer que toutes les variables respectent ces paramètres. De deux choses l'une :

- le contrôle est positif : les opérations sont effectivement ordonnancées et l'ordonnancement continue ;
- le contrôle est négatif en ce sens qu'on sort des limites fixées ; dans ce cas, on change de règle ou/et d'heuristique et on effectue un nouveau contrôle : si les paramètres sont respectés, on ordonnance les opérations selon cette dernière règle et/ou heuristique ; l'ordonnancement continue en appliquant la règle ou/et l'heuristique de base.

  S'il est impossible de trouver une solution "satisfaisante", on recommence tout l'ordonnancement, sur base d'une autre règle et/ou heuristique.

d. Dans une optique dynamique, on resserre les limites jusqu'à ce qu'il ne soit plus possible de trouver un ordonnancement qui les respecte.

§ 3. Remarques

Une solution sera d'autant plus rapidement acceptée ou rejetée que les projections sont bonnes. Il ne s'agit pas de dire: les retards subis jusqu'à présent étant nuls, on continue ; il faut au contraire se poser la question suivante : si j'ordonnance cette opération maintenant et que j'effectue un contrôle concernant l'impact actuel et futur de cet ordonnancement sur les retards, quel est le retard minimum entraîné ? Si ces projections sont bonnes, on peut souvent arriver très rapidement à la conclusion : continuer sur base de la règle X et l'heuristique Y ne donnera pas de solutions satisfaisantes. Exemple : dans le problème traité dans le chapitre suivant, nous avons pu affirmer, au moment 18 que l'ordonnancement utilisant la règle de base 1 (sans heuristiques) ne pourra pas donner de solution satisfaisant les paramètres (l'ordonnancement s'effectue sur plus ou moins 65 unités de temps).

Les règles et heuristiques doivent être simples et refléter un ou plusieurs aspects des limites de contrôle ; prenons trois exemples :

- pour les retards : une règle tenant compte du "temps restant" doit être inclue (exemple: règle SLACK) ;
- pour le temps : une règle basée sur les durées de traitement des opérations doit être inclue (exemple : règle SPT) ;
- pour les durées d'attente d'une tâche : exemple : règle FCFS.

Il est possible qu'aucune solution ne soit satisfaisante. Cela ne signifie pas nécessairement qu'il n'y a aucun ordonnancement qui vérifie ces limites de contrôle (pour arriver à cette conclusion, il aurait fallu examiner tous les ordonnancements possibles). On peut remédier à cette situation de deux façons :

- augmenter le nombre de règles ou/et d'heuristiques ; cette politique accroît le nombre de combinaisons possibles à explorer ;
- vérifier les causes pour lesquelles aucune solution satisfaisante n'a pu être trouvée. Il peut en effet s'avérer que les limites de contrôle soient trop restrictives.

La méthode telle que nous l'avons décrite est statique. De façon dynamique, on pourrait se fixer, au départ des limites relativement lâches ; lorsqu'un ordonnancement satisfait ces limites, on resserre les limites (après examen des résultats obtenus) et on cherche un nouvel ordonnancement jusqu'à ce qu'il ne soit plus possible d'en trouver un qui vérifie ces paramètres. Dans cette perspective, on pourrait peut-être trouver un ordonnancement proche de l'optimum mais notons qu' :

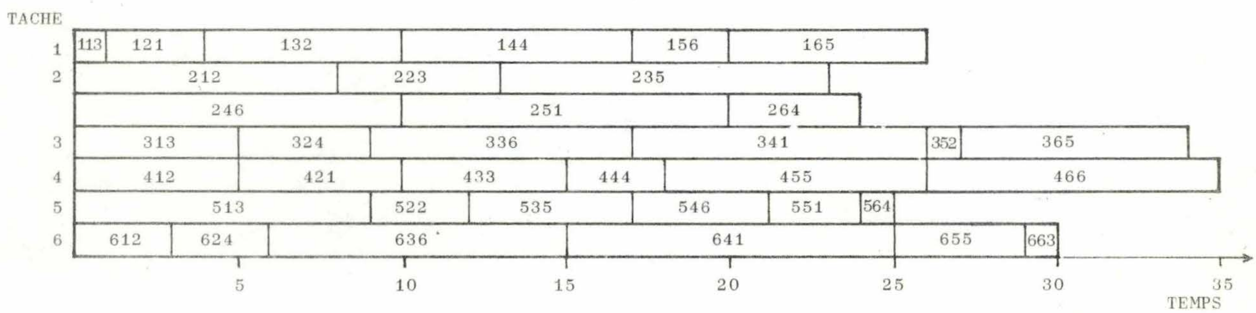
- une solution optimale supposerait que nous délimitions l'espace de toutes les solution possibles (programmation linéaire). Or, cette délimitation n'est possible que dans le cas d'une énumération complète de tous les ordonnancements.

- il faudrait établir soit une hiérarchie de valeur entre les paramètres de contrôle, soit les ramener à une commune mesure. Dans le premier cas, on attribuerait une côté à tous les paramètres et on choisirait la solution dont la côté est la plus élevée. Remarquons cependant que cette politique n'est pas optimale, car nous supposons à nouveau connaître toutes les solutions satisfaisantes.

Chapitre 5 - Résultats empiriques

Section 1 - Données du problème

Nous avons voulu appliquer les deux formes d'heuristiques que nous venons de décrire. A défaut du programme de Gere, nous avons élaboré nous-même un programme Fortran et nous l'avons appliqué à un seul problème. Le diagramme-tâche de ce problème se présente de la manière suivante :



Les dates de livraison de ces tâches sont respectivement de 56, 56, 50 55, 57 et 61

Les 15 règles testées sont les suivantes :

1. RANDOM : cette règle se base sur le numéro d'identification de la tâche
2. FDCS : premier arrivé, premier servi
3. LPT : durée de traitement la plus longue
4. MWKR : somme des durées de traitement restant à effectuer ; on choisit la tâche dont la somme est la plus élevée
5. LWKR : même concept, mais on choisit la tâche dont la somme est la moins élevée

6. MWKR-P : même concept, mais on en soustrait la durée de traitement de l'opération ordonnançable

7. MWKR/P : même concept, mais on la divise par la durée de traitement de l'opération ordonnançable

8. MOPNR : on choisit la tâche dont il reste le plus grand nombre d'opérations à traiter

9. MWKR/OPNR : on choisit la tâche dont le rapport entre la somme des durées de traitement restant à effectuer et le nombre d'opérations restantes est le plus élevé

10. DATE : règle des dates de livraison (on se réferera au chapitre 4 section 3 pour la définition des 4 règles suivantes : il s'agit de règles utilisées par Gere (se basant d'une façon ou d'une autre sur les dates de livraison)

11. SLACK : règle du temps restant

12. SLACK/OPNR : règle du temps restant par opération

13. SLACK ratio : règle du rapport de temps restant

14. SLACK-SPT : combinaison de la règle du temps restant et de la plus courte opération

15. SPT : règle de la plus courte opération

Les heuristiques associées à ces règles sont celles que nous avons décrites dans le chapitre 4, section 3 et qui furent programmées par Gere : heuristique d'alternance de projection (plus insertion) et combinaison de toutes ces heuristiques.

Remarque :

Tous les ordonnancements sont de type "non-retardés" [1] : une décision est prise au moment où devant une machine inoccupée, une ou plusieurs

---

[1] Excepté les ordonnancements utilisant l'heuristique de projection ; ils sont de type "actifs".

opérations sont susceptibles d'être ordonnancées (c-a-d respectent les contraintes de routing). De plus, toutes les tâches sont disponibles au moment où l'ordonnancement commence.

Section 2 : Critères retenus

- critères, fonctions de la durée d'occupation de l'atelier (F) :

  - Date à laquelle chacune des tâches quitte l'atelier (F_i)
  - Date à laquelle l'atelier sera libre (aura terminé le traitement de toutes les tâches) : F_{max}.
  - Durée moyenne d'occupation de l'atelier par les tâches : $\bar{F}$.

- critères, fonctions des avances ou des retards des tâches (L) :

  - avance (retard) de chacune des tâches : $-L_i$
  - avance (retard) minimum (maximum) subi par une des tâches : $-L_{max}$
  - avance (retard) moyenne de toutes les tâches : $-\bar{L}$
  - Somme des retards : $\sum_{i=1}^{n} T_i$
  - Somme des avances : $\sum_{i=1}^{n} E_i$

- critères, fonctions des attentes subies par les tâches (W_i) :
  - nombre maximum de tâches en attente devant une machine à un moment donné,
  - durée totale pendant laquelle une tâche est restée en attente : W_i,
  - durée maximum, séparant le traitement de deux opérations consécutives d'une même tâche,
  - durée totale pendant laquelle toutes les tâches sont restées en attente : $\sum_{i=1}^{n} W_i$; ce critère ainsi que le premier donne une idée du volume et de l'importance des stocks intermédiaires nécessaires.

- critères, fonctions de l'inoccupation des machines :

  - durée totale pendant laquelle une machine est restée inoccupée ; calculée à partir du moment 0, cette durée s'étend jusqu'au moment où la machine en question a terminé la dernière opération qu'elle doit traiter ; nous supposons, en effet qu'un autre ordonnancement (utilisant cette machine) pourrait commencer dès ce moment,

. durée maximum séparant le traitement de deux tâches, ordonnancées consécutivement sur la même machine,

. durée totale pendant laquelle l'ensemble des machines sont restées inoccupées.

Remarque : nous avons estimé la valeur minimum que pourrait prendre F_{max} ; cette estimation se base sur :

- la durée totale de traitement d'une tâche,
- la somme des durées de traitement à effectuer par chaque machine,
- la date au plus tôt à laquelle une machine pourra commencer la première opération qu'elle doit traiter.

Après avoir ajouté pour chaque machine, cette date au plus tôt et la somme des durées de traitement à effectuer par cette machine, nous avons pris la valeur maximum de ces 6 termes et des durées totales de traitement d'une tâche ; nous avons obtenu 52. Cela signifie qu'un ordonnancement réalisant cette valeur est optimal en fonction de F_{max} mais cela ne signifie pas qu'un ordonnancement ne réalisant pas cette valeur n'est pas optimal (il semble que la valeur optimale de F_{max}, dans notre exemple s'élève à 58).

Section 3 : Résultats d'application de REGLES heuristiques

Sur base des 15 règles, associées ou non avec une ou plusieurs heuristiques, nous avons abouti à créer 60 ordonnancements. Notre critère étant la minimisation de la somme des retards, le tableau suivant reprend ces retards en fonction de la règle et/ou de l'heuristique utilisée.

TABLEAU DE LA SOMME DES RETARDS

Heuristique utilisée

Règle N°	Aucune	Alternance	Projection + Insertion	Toutes	Alternance élargie
1	21	20	21	20	10
2	10	0	8	1	12
3	48	27	48	27	12
4	0	0	3	1	12
5	27	12	14	14	12
6	10	0	11	1	12
7	21	12	13	5	12
8	9	6	11	1	12
9	27	18	27	18	12
10	28	16	13	15	12
11	5	5	5	5	12
12	0	0	1	1	12
13	0	0	3	1	Pas testé
14	6	6	1	1	Pas testé
15	36	12	40	14	Pas testé

Commentaires :

a. Heuristique d'alternance

En appliquant strictement la démarche de Gere, nous aboutissons dans tous les cas à de meilleurs ordonnancements (ou égaux). L'hypothèse d'efficacité de cette heuristique est donc confirmée. Nous avons voulu, comme nous le signalions dans le chapitre précédent élargir cette heuristique en l'appliquant également lorsque l'opération susceptible de remplacer celle donnée par la règle n'est pas critique mais entraîne une somme des avances plus élevée. Malheureusement les résultats ne se sont pas avérés concluants, puisque dans

6 cas seulement (sur 12) les retards furent réduits. Notons que ces 6 ordonnancements (règle 1, 3, 5, 7, 9, et 10) sont meilleurs (quant aux retards) que ceux obtenus par application stricte de l'heuristique d'alternance (le retard moyen s'élève respectivement à 10,7 et à 17,5). En recherchant les raisons pour lesquelles l'heuristique d'alternance "élargie" ne donne pas plus souvent de meilleurs résultats, nous avons remarqué ceci : l'application de l'heuristique d'alternance en diminuant l'avance de la tâche, dont une opération est ordonnancée selon la règle et en augmentant l'avance de la tâche, dont une opération est choisie par l'heuristique, augmentait l'écart séparant ces deux avances. En d'autres termes l'avance de la tâche (choisie par l'heuristique) avait trop d'avance. Or, comme le calcul des avances se fait par projection et ne donne que des avances maxima (retards minima), la tâche (choisie par la règle) pouvait subir des retards au cours de l'ordonnancement. Pour pallier ces inconvénients, nous aurions dû introduire des hypothèses supplémentaires :

- la tâche différée doit garder une avance suffisante (ce qui pose le problème du "niveau" de cette avance)
- l'écart entre les avances des 2 tâches ne peut dépasser une certaine valeur.

b. Heuristique de projection (insertion)

Elle s'est avérée supérieure ou égale dans 9 cas, lorsqu'on compare les ordonnancements obtenus avec ceux n'utilisant aucune heuristique Les ordonnancements utilisant l'heuristique d'alternance se sont avérés meilleurs que ceux utilisant la projection dans 13 cas. On vérifie donc l'hypothèse que l'heuristique de projection est efficace mais on ne vérifie pas celle (voir conclusions de Gere, supra) selon laquelle elle est meilleure que l'heuristique d'alternance.

Nous avons recherché les raisons pour lesquelles cette heuristique ne s'avère pas aussi efficace que le prétend Gere :

- nous avons expliqué que le principe de cette heuristique est différent de celui de l'alternance en ce sens qu'il s'appuie sur une tâche en retard et non pas sur l'ensemble des tâches

- Gere palliait cet inconvénient de deux manières :

  - utilisation de l'heuristique d'insertion
  - la tâche différée devait être en avance et cette avance devait dépasser deux fois celle de la tâche choisie par l'heuristique.

Bien que l'heuristique d'insertion était programmée, nous ne l'avons jamais rencontrée. N'oublions pas que les problèmes étudiés par Gere étaient plus complexes que le nôtre : de ce fait le nombre de conflits était accrû. On peut également se demander si la constante 2 n'est pas par trop arbitraire. [1]

- Nous avons aussi élargi cette heuristique en l'appliquant aux tâches qui n'étaient pas "critiques" : il suffisait que l'avance de la tâche susceptible de projection soit inférieure à celle déterminée par la règle. Nous pensions que le contrôle des retards, effectué en projetant la tâche choisie par l'heuristique est suffisant (2 conditions ci-dessus). Le tableau suivant reprend certaines données relatives aux ordonnancements où l'heuristique de projection ne s'avère pas efficace.

Règle	N° de la tâche : Projetée	Différée	Avance (retard) une fois l'ord. terminé des tâches projetées et différées sans et avec heuristique — Projetée : Sans	Avec	différée : Sans	Avec	Avance (retard) des autres tâches : avance d'une tâche avec heuristique moins avance sans heuristique ; les tâches non signalées sont inchangées
4	2	1	0	2	0	0	Tâche 5 : -3 ; Tâche 4 : +3
6	2	1	0	2	3	3	Tâche 4 : -1 ; Tâche 5 : +2
8	2	5	-3	2	15	2	Tâche 1 : -6 ; Tâche 3 : -19 Tâche 4 : -2
12	2	1	0	2	0	0	Tâche 4 : -1 ; Tâche 5 : +2
13	2	1	0	2	0	0	Tâche 4 : -1 ; Tâche 3 : +2
15	2	4	-32	-5	14	-21	Tâche 1 : -5 ; Tâche 3 : -5 Tâche 5 : -13 ; Tâche 6 : -5

[1] cfr. dernier alinéa de la page 183

Si nous regardons non seulement du côté de la somme des retards mais également du côté de la somme des avances on s'aperçoit que la différence entre les <u>écarts</u> (avec ou sans l'heuristique) est en défaveur de l'application de projection dans deux cas seulement : règles 8 et 15; pour les autres règles que nous venons de signaler, la différence entre les écarts sont respectivement (une valeur positive signifie que l'ordonnancement, fonction de la somme des écarts est préférable lorsque l'heuristique est appliquée) : +2, +3, -18, +3, +2 et -36.

De tout ceci, nous retiendrons :

- que l'heuristique de projection est efficace
- qu'elle aurait été plus efficace si des occasions d'insertion s'étaient présentées
- que l'élargissement de la procédure est justifiée
- que cette heuristique est dangereuse en ne s'occupant que d'une tâche; il faudrait probablement, en plus des conditions d'application énumérées ci-dessus, effectuer un contrôle sur la somme des retards entraînés par son utilisation.

c. <u>Utilisation conjointe de toutes les heuristiques</u>

Comparés aux ordonnancements n'utilisant aucune heuristique, les ordonnancements les appliquant <u>toutes</u> sont meilleurs ou équivalents dans 12 cas; l'utilisation de <u>toutes</u> les heuristiques s'est avérée meilleure que l'heuristique d'<u>alternance</u> seule dans 8 cas, et que l'heuristique de <u>projection</u> seule dans 14. Les hypothèses de Gere sont donc confirmées; les remarques faites à propos de l'heuristique de projection sont valables ici et justifient que l'utilisation de toutes les heuristiques ne s'est pas avérée plus souvent efficace (lorsqu'on compare avec l'utilisation de l'heuristique d'alternance seule).

d. <u>Remarques générales</u>

- Du point de vue des règles, il se confirme que la règle aléatoire.

(n° 1) et la règle SPT (n° 15) ne donnent pas de bons ordonnancements lorsque le critère est fonction des dates de livraison. Par contre les règles qui s'appuient sur ces dates (de façon dynamique) les règles 11 à 14 en l'occurence, donnent de bons résultats.

- Si nous avions groupé sur un même graphique, la somme des retards en fonction de la règle et/ou de l'heuristique utilisée, nous aurions remarqué que les lignes sont loin d'être parallèles : si on peut affirmer qu'une règle donne un meilleur ordonnancement lorsqu'elle est accompagnée d'heuristique(s), on ne vérifie pas que la hiérarchie d'efficacité entre les règles est identique suivant qu'elles sont ou non accompagnées d'heuristiques : exemple : prenons les règles 1, 9 5, 10 ; si nous les mettons par ordre d'efficacité, on obtient, si elles sont accompagnées :

  - d'aucune heuristique : 1 - 9 - 5 - 10
  - de l'heuristique d'alternance : 5 - 10 - 9 - 1
  - de l'heuristique de projection : : 10 - 5 - 1 - 9
  - de toutes les heuristiques : 5 - 10 - 9 - 1

- Sur les 72 ordonnancements recherchés (les 15 règles associées ou non avec l'une des heuristiques, plus les 12 de l'heuristique d'alternance élargie) nous avons obtenu 42 ordonnancements différents ils ne diffèrent pas nécessairement du point de vue de la valeur prise par les critères ; nous les reprenons en annexe (annexe VI). Le tableau qui suit reprend le n° de l'ordonnancement en fonction de la règle et ou de l'heuristique utilisée.

	heuristique				
Règle	Aucune	Alternance	Alternance élargie	Projection	Toutes
1	1	2	3	1	2
2	3	4	15	29	18
3	5	6	16	5	6
4	7	7	15	30	18
5	8	9	17	31	19
6	10	7	15	32	18
7	11	12	17	33	20
8	13	14	15	34	21
9	35	22	15	35	22
10	8	39	14	36	23
11	40	40	15	24	24
12	7	7	15	25	25
13	7	7	--	30	26
14	41	41	--	27	27
15	42	12	--	37	28

Si plusieurs valeurs sont identiques sur la même ligne, cela signifie que les ordonnancements basés sur la même règle mais sur une heuristique différente, sont équivalents.

Section 4 - Résultats d'application de l'heuristique, analyse de situation

Les paramètres de contrôle sont les suivants :

- en fonction du temps :
  - F_{max} ne peut pas être supérieur à 61 ;
  - $\bar{F}$ ne peut être supérieur à 56 ;
- en fonction des retards
  - aucune tâche ne peut être en retard ;
  - le retard maximum ne peut être supérieur à 0 ;
  - l'écart moyen ne peut être négatif ;

  Remarque

  En fait, les deux dernières limites sont inutiles car elles découlent de la première (aucune tâche en retard) ; rien ne s'oppose à les rendre non-redondantes.
- en fonction des durées d'attente des tâches
  - 18 périodes de temps peuvent séparer au maximum la fin d'une opération suivante d'une même tâche ;
  - la durée totale d'attente d'une tâche ne peut dépasser 33 périodes ; remarquons que pour ces deux derniers critères, nous aurions pu attribuer des limites pour chacune des tâches ;
  - quel que soit la machine, il ne peut pas y avoir plus de 3 tâches en attente au même moment devant l'une d'entre elles ;
- en fonction des durées d'inoccupation machine
  - une machine ne peut pas rester inoccupée pendant plus de 35 périodes de temps consécutives ;
  - la durée totale d'inoccupation ne peut pas dépasser 38 ; remarquons comme pour les tâches que nous aurions pu fixer une limite pour chacune des 6 machines.

Ces paramètres, à défaut d'un cas pratique, sont basés sur les meilleurs ordonnancements obtenus par l'application des règles heuristiques (section précédente) ; il s'agit des ordonnancements n° 4 et 7.
45 solutions ont été envisagées ; elles varient selon la règle et l'heuristique de base utilisées. Faisons rapidement l'examen de certaines solutions mais notons auparavant que les limites étant très restrictives, peu d'ordonnancements sont apparus satisfaisants :

- règle de base n° 1 :

  qu'elle soit accompagnées ou non d'heuristiques (4 possibilités), la solution est rejetée au moment 18 (grâce aux projections effectuées quant à la valeur des variables, on peut conclure, au moment 18 : "il est impossible de trouver une solution") : 19 périodes de temps séparaient le traitement de deux opérations consécutives de la tâche 6.

- règle de base n° 2 :

  - la solution est satisfaisante, lorsqu'aucune heuristique n'est utilisée ; la règle de base a été appliquée tout au long de l'ordonnancement, excepté au moment 35, où l'opération 455 a été ordonnancée selon la règle n° 1, accompagnée de l'heuristique d'alternance ; l'ordonnancement obtenu porte le n° 4
  - la solution est satisfaisante lorsque l'heuristique d'alternance est utilisée ; cette règle et heuristique fut appliquée automatiquement tout au long de l'ordonnancement (ordonnancement n° 4)
  - la solution est impossible lorsque cette règle est associée à l'heuristique de projection ou à toutes les heuristiques : le rejet de la solution se fait au moment 43 et est dû aux limites concernant les durées d'attente des tâches. Notons que dans le premier cas (projection) l'opération 455 a été ordonnancée au moment 35 grâce à la règle 1, accompagnée de l'heuristique d'alternance ce qui suppose que l'application automatique de la règle et de l'heuristique de projection faisait sortir l'ordonnancement d'une ou de plusieurs limites (au moment 35).

- <u>règle de base n° 3</u> :

  toutes les possibilités sont rejetées au moment 19 : la tâche 3 accusait un retard (estimation minimum) de deux.

- <u>règle de base n° 4</u> :

  l'application automatique de cette règle non accompagnée d'heuristiques et accompagnée de l'heuristique d'alternance donne un ordonnancement <u>satisfaisant</u> (ordonnancement n° 7). Les deux autres possibilités (heuristique de projection et toutes les heuristiques) sont rejetées au moment 43. Les durées d'attentes maxima des tâches étaient supérieures à 18.

Dans le tableau qui suit, nous reprenons l'ensemble des solutions testées. Si nous comparons les résultats obtenus avec ceux qui proviennent de l'application de règles heuristiques, nous pouvons tirer une conclusion très importante : l'application automatique des règles (section précédente) aboutissait à 8 ordonnancements qui vérifient les limites de contrôle (sur 72 ordonnancements envisagés). Ici sur les 45 solutions testées, 10 apparaissent "satisfaisantes" ce qui signifie qu'indépendamment des solutions qui sont satisfaisantes grâce à l'application automatique de la règle ou de l'heuristique de base, il y en a 2 qui utilisent au cours de l'ordonnancement une autre règle ou une autre heuristique que la règle et l'heuristique de base. Il s'agit des ordonnancements utilisant :

- la règle de base n° 2, aucune heuristique
- la règle de base n° 6, aucune heuristique (voir tableau).

SOLUTIONS ENVISAGEES

Règle de base	Heuristique de base [1]	Solution	date d' [2] impossibilité	Autre règle ou heuristique utilisée [3]	Ordonnancement n° [4]
1	R	impossible	18	non	-
1	A	impossible	18	non	-
1	P	impossible	18	non	-
1	AP	impossible	18	non	-
2	R	SATISFAISANTE	-	règle 1,heuristique d'alternance(455)	4
2	A	SATISFAISANTE	-	non	4
2	P	impossible	43	règle 1,heuristique d'alternance(455)	-
2	AP	impossible	43	non	-
3	R	impossible	19	non	-
3	A	impossible	19	non	-
3	P	impossible	19	non	-
3	AP	impossible	19	non	-
4	R	SATISFAISANTE	-	non	7
4	A	SATISFAISANTE	-	non	7
4	P	impossible	-	non	-
4	AP	impossible	-	non	-
5	R	impossible	15	non	-
5	A	impossible	15	non	-
5	P	impossible	41	règle 1,sans heuristique(663 et 455)	-
5	AP	impossible	41	non	-
6	R	SATISFAISANTE	-	règle 1,sans heuristique(165)	7
6	A	SATISFAISANTE	-	non	7
6	P	impossible	43	non	-
6	AP	impossible	43	non	-
7	R	impossible	19	règle 1,heuristique de projection(223)	-
7	A	impossible	19	règle 1,heuristique de projection(223)	-
7	P	impossible	36	non	-
7	AP	impossible	36	non	-
8	R	impossible	19	non	-
8	A	impossible	19	non	-
8	P	impossible	44	règle 1,aucune heuristique(365)	-
8	AP	impossible	44	non	-
9	R	impossible	18	non	-
9	A	impossible	18	non	-
9	P	impossible	18	non	-
9	AP	impossible	18	non	-
10	R	impossible	19	règle 1,aucune heuristique(223)	-
10	A	impossible	19	non	-
11	R	impossible	37	non	-
12	R	SATISFAISANTE	-	non	-
12	A	SATISFAISANTE	-	non	-
13	R	SATISFAISANTE	-	non	-
13	A	SATISFAISANTE	-	non	-
14	R	impossible	19	non	-
15	R	impossible	19	règle 1,heuristique d'alternance(212) règle 1,heuristique de projection(223)	-

[1] : R = aucune heuristique ; A = heuristique d'alternance ; P= heuristique de projection; AP= toutes les heuristiques.

[2] : Lorsqu'une solution est impossible,date à laquelle elle apparait comme telle,grâce aux projections faites sur l'ensemble de l'ordonnancement.

[3] : Lorsque une (ou plusieurs) opération est ordonnée selon une règle et une heuristique différente à la règle et à l'heuristique de base, nous l'avons indiqué : (entre parenthèses, l'opération en question) dans le cas inverse, l'application de la règle et de l'heuristique de base est systématique.

[4] : n° de l'ordonnancement ; voir annexe VI.

Conclusion de cette partie

Dans cette troisième partie, nous avons essayé de répondre aux questions posées dans l'introduction. Le sujet n'a été qu'effleuré et présente de nombreuses limites. Indépendamment de ces lacunes, nous voudrions revenir rapidement sur un point déjà soulevé (chapitre 4, section 4). Nous avons dit que nous avions affaire à un problème heuristique lorsque, entre autres, il était impossible de définir un critère unique, qui serait le reflet de l'objectif de l'entreprise. Nous avons résolu ce problème en introduisant des paramètres de contrôle et en considérant ceux-ci comme une traduction de ce critère. Mais, définir un critère, ne suppose-t-il pas l'obtention d'un seul résultat ? Dans le cas d'un programme linéaire, le critère retient, parmi toutes les solutions possibles une solution, qui est alors qualifiée d'optimale. Dans le cas de l'ordonnancement, nous n'aboutissons qu'à des solutions "satisfaisantes" Il nous semble qu'en réalité les paramètres de contrôle ne traduisent pas (ou mal) la fonction-critère mais introduisent des contraintes supplémentaires ; ces contraintes ne font que réduire le nombre de solutions. Tout le problème du critère réapparaît, d'autant plus que le domaine des solutions n'est pas délimité, excepté dans le cas d'une exploration systématique de tous les ordonnancements possibles. Même dans une perspective dynamique, la solution optimale ne pourra être qu'approchée.

Nous pensons qu'il appartient à la nature même des problèmes heuristiques de ne pouvoir aboutir qu'à des solutions "satisfaisantes".

CONCLUSIONS GENERALES

Parmi les problèmes d'organisation de la production se pose le problème de l'ordonnancement de plusieurs commandes, effectuées sur des équipements différents. Les solutions algorithmiques ne sont pas satisfaisantes parce qu'elles se situent toutes au niveau de l'ordonnancement lui-même, indépendamment de son contexte.

Dans une perspective d'entreprise, système intégré, l'approche algorithmique devait être critiquée, non seulement quant aux résultats auxquels elle aboutit, mais également quant au bien-fondé de l'utilisation de cette approche. Le problème d'ordonnancement apparait alors heuristique et non pas algorithmique.

Si on définit l'heuristique, comme la formalisation de caractéristiques de l'esprit humain, l'approche heuristique consiste à se définir des règles de décision formalisables. Nous avons testé quelques règles applicables à l'ordonnancement : l'application de ces règles est efficace. Les limites relatives à notre travail et à l'utilisation de ces règles sont les suivantes :

- nous n'avons traité qu'un seul problème,
- ce problème était statique,
- les règles utilisées sont peu nombreuses et sont très difficiles à programmer, car il faut tenir compte d'un grand nombre de circonstances (cfr. heuristique d'alternance élargie),
- elles ne tiennent compte que d'un seul critère.

Certains limites relatives à l'utilisation des règles disparaissent en abordant le problème heuristique en tant qu'analyse de situation. Après s'être fixé des limites, on fait évoluer l'ordonnancement à l'intérieur de ces limites. Les conclusions relatives à cette approche sont les suivantes :

- le nombre d'ordonnancements satisfaisant les paramètres fixés est plus élevé que celui obtenu par l'application systématique d'une règle et/ou d'une heuristique ;
- la méthode est plus facile à programmer,
- notre programme n'étant pas "opératoire", nous ne tirerons aucune conclusion quant au temps machine.
- bien qu'elle ne résoud pas entièrement le problème du critère, la méthode tient compte de multiples aspects de l'entreprise.

Cette dernière conclusion aboutit une conclusion plus générale : l'approche heuristique tient compte de l'entreprise dans son ensemble : un problème d'entreprise, quel qu'il soit, DOIT être remis dans son contexte, tenir compte ET rejaillir sur tout son environnement. Si on néglige ce point, les solutions d'un problème particulier peuvent s'avérer incompatibles. Dans une optique cybernétique, la solution sera économique, quel que soit la manière dont la nature "économique" d'un problème est définie : problème décisionnel (aspect organisation), ou problème d'affectation (aspect fonction de production).

La plupart des problèmes d'entreprise (voire tous) sont de nature heuristique ; la méthode heuristique s'appuie sur un côté technique et décisionnel, d'où la nécessité d'un décideur et d'un dialogue Homme-Machine : si nos résultats peuvent manquer de souplesse, il faut imputer ce défaut à l'absence de dialogue. La solution heuristique ne sera jamais optimale, elle ne sera que "satisfaisante".

Nous sommes partis d'un problème particulier, celui de l'ordonnancement non-unitaire et nous avons abouti sur l'entreprise dans son ensemble. Nous aurions pu partir de n'importe quel autre problème (gestion des stocks, investissements, établissement des prix de revient,...) et nous aurions abouti au même résultat. Nous n'avons pris qu'un exemple de processus heuristiques de décision.

B I B L I O G R A P H I E

ACKOFF, R.L. - "Progress in Operations Research" ; John Wiley & Sons, Inc., New York, 1961

AKERS, S.B. - "A Graphical Approach to Production Scheduling Problems" Journal of the O.R.S.A., vol. 4 n°4 a.avril 1956

AKERS, S.B. et FRIEDMAN, J. - "A Non-Numerical Approach to Production Scheduling Problems" ; Journal of the O.R.S.A., vol. 3, n° 4, novembre 1955

BAKER, C.T., DZIELINSKI, B.P. et MANNE, A.S. - "Simulation Tests of Lot Size Programming" ; Management Science, vol. 9, n° 2 janvier 1963

BAKHRU, A.N. et RAO, M.R. - "An Experimental Investigation of Job-Shop Scheduling" ; Research Report Department of Industrial Engineering Cornell University, 1964

BELLMAN, R. - "Dynamic Programming Treatment of the Travelling Salesman Problem" ; J. Assn. for Computing Machinery, vol. 9 n° 1, janvier 1962

BOULDING, K.E. et SPIVEY, W.A. - "La programmation linéaire et la théorie de l'entreprise" ; Dunod 1964

BOURGEOIS, M.- "Les modèles de décision" ; Conférence du 23-24 octobre 1968 ; Compagnie internationale pour l'informatique, Ecole Supérieure de Guerre Aérienne, Ecole des Hautes Etudes Commerciales.

BOWMAN, E.H. - "The Schedule Sequencing Problem" ; Journal of the O.R.S.A., vol. 7 n° 5, Septembre 1959

BROOKS, G.H. et WHITE, C.R. -- "An Algorithm for Finding Optimal or Near-Optimal Solutions to the Production Scheduling Problem" ; J. Ind. Eng., vol 16, n° 1, janvier 1965

BUFFA, E.S. - "Modern Production Management (2nd Edition)" ; John Wiley & Sons, Inc; New York, 1965

BURSTALL, R.M. "A Heuristic Method for a Job-Shop Scheduling Problem" ; Operations Research Quarterly, novembre 1966

CARROLL, D.C. - "Man-Machine Cooperation on Planning and Control Problems", presented at the International Symposium on Long Range Planning for Management, sponsored by the Internationaal Computation Center, Rome, held at UNESCO, Paris, Sept. 20-24, 1965

CARROLL, D.C. - "Heuristic Sequencing of Single and Multiple Component Jobs" ; unpublished Ph. D. Dissertation, Sloan School of Management, M.I.T. juin 1965

CHURCHMAN, C.W. - ACKOFF, R.L. et ARNOFF, E.L. - "Introduction to Operations Research" ; John Wiley & Sons, Inc. New York 1957

CLARK, W. - "The Gantt Chart (3rd Edition)" ; Pitman and Sons, Londres 1952

CONWAY, R.W., MAXWELL, W.L. et MILLER, L.W. - "Theory of Scheduling" ; Addison Wesley Publishing Company 1967

DUDEK, R.A. et TEUTON, O.F. - "Development of M-Stage Decision Rule for Scheduling n Jobs through m Machines" ; Journal of th O.R.S.A. vol. 12 n° 3, mai 1964

EASTMAN, W.L.; EVEN, S. et ISAACS, I.M. - "Bounds for the Optimal Scheduling of n Jobs on m Processors ; Management Science, vol. 11 n° 2, novembre 1964

EILON, S. - "Elements of Production Planning and Control" ; Mac Millan Company, New York, 1962

FISHER, H. et THOMPSON, G.L. - "Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules" ; Muth J.F. et Thompson, G.L., chap. 15

FORD, L.R. et FULKERSON, D.R. - "Flows in Network" ; Princeton University Press 1962

GAVETT, J.W. - "Three Heuristic Rules for Sequencing Jobs to a Single Production Facility" ; Management Science vol. 11 n° 8 juin 1965

GERE, W.S. - Heuristic Approach to Job-shop Scheduling" ; unpublished Ph. D. Thesis, Carnegie Institute of Technology, 1962

GERE, W.S. - Heuristics in Job-Shop Scheduling" ; Management Science vol. 13 n° 3 novembre 1966

GIFFLER, B. et THOMSPON, G.L - "Algorithms for Solving Production Scheduling problems" ; Journal of the O.R.S.A., vol 8 n° 4, juillet 1960

GIFFLER, B., THOMPSON, G.L. et VAN NESS V. "Numerical Experience with the Linear and Monte Carlo Algorithms for Solving Production Scheduling Problems" ; Muth J.F. et Thompson G.L. Chap. 3

GIGLIO, R.J. et WAGNER, H.M. - "Approximate Solutions to the three Machine Scheduling Problem" ; Journal of the O.R.S.A. vol 12 n° 2, mars 1964.

GOMORY, R.E. - "An Algorithm for Integer Solutions to Linear Programs" ; Princeton, IBM Mathematics Research Project Technical Report, n° 1, november 1958

GOMORY, R.E. - "An All Integer Programming Algorithm" ; Muth J.F. et Thompson G.L. Chap 13

HELD, H. et KARP, R.M. - "A Dynamic Programming Approach to Sequencing Problems" ; J. Soc. Ind. and Appl. Math. vol. 10 n° 2, mars 1962

HELLER, J. - "Combinatorial, Probabilistic and Statiscal Aspects of an M x J Scheduling Problem" ; NYO-2540, AEC Research and Development Report, fevrier 1959

HELLER, J. - "Some Numerical Experiments for an M x J Flow-Shop and its Decision-Theoretical Aspects" ; Journal of the O.R.S.A. vol. 8, n° 2, mars 1960

IGNALL, E. et SCHRAGE - "Applications of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems" ; Journal of the O.R.S.A. vol 13 n° 3 mai 1965

JACKSON, J.R. - "Scheduling a Production Line to Minimize Maximum Tardiness" Research Report 43, Management Science Research Project UCLA, janvier 1955

JACKSON, J.R. - "An Extension of Johnson's Result on Job-Lot Scheduling" ; Naval Research Logistics Quarterly vol. 3 n° 3 Septembre 1956

JEREMIAH, B. LALCHANDANI A. et SCHRAGE L. - "Heuristic Rules toward Optimal Scheduling" ; Research Report, Department of Industrial Engineering, Cornell University, 1964

JOHNSON, S.M. - "Optimal Two-and Three-Stage Production Schedules with Setup Times Included" ; Muth J.F. et Thompson G.L. Chap. 2

KARUSH, W. - "A Counterexample to a Proposed Algorithm for Optimal Sequencing of Jobs" ; Journal of the O.R.S.A., vol 13 n° 5, mars 1965

LITTLE, J.D.C., MURTY, K.G. SWEENY, D.W. et KAREL, C. - "An Algorithm for the Traveling Salesman Problem" ; Journal of the O.R.S.A., vol. 11 n° 6 novembre 1966

LOMNICKI, Z.A.- "A Branch and Bound Algorithm for the Exact Solution of the Three Machine Scheduling Problem" ; Operations Research Quarterly, vol. 16, n° 1, 1965

MANNE, A.S. - "On the Job-Shop Scheduling Problem" ; Journal of the O.R.S.A. vol. 8 n° 2 mars 1960; Muth J.F. et Thompson G.L. Chap. 12

MELLO, P. - "A Review of Job-Shop Scheduling" ; Operations Research Quarterly, juin 1966

MITTEN, L.G. - "Sequencing n Jobs on 2 Machines with Arbitrary Time lags" ; Management Science vol. 5 n° 3, avril 1959

MUTH, J.F. et THOMPSON G.L. - "Industrial Scheduling" ; Prentice Hall, Incl. Englewood Cliffs, N.J. 1963

NEWELL, A. SHAW, J.C. et SIMON, H. - "A General Problem Solving Program for a Computer" ; Computers and Automation, vol. 8 juillet 1959

NUGENT, C.E. - On Sampling Approaches to the Solution of the n by n Static Sequencing Problem" ; Ph. D. Thesis, Cornell University, september 1964

POUNDS, W.F. - "The Scheduling Environment" ; Muth, J.F. et Thompson, G. L. Chap. 1.

REINITZ, R.C. - "On the Job-Shop Scheduling Problem" ; Muth J.F. et Thompson, G.L. Chap. 5

ROWE, A.J. - "Toward a Theory of Scheduling" ; Journal of Industrial Engineering, vol. 11 n° 2, mars 1960

ROY, B. - "Les problèmes d'ordonnancement : applications et méthodes" Dunod, 1965

ROY, B. - "Cheminement et connexité dans les graphes : Applications aux problèmes de l'ordonnancement" ; METRA série spéciale n° 1, 1962

RUSSO, F.J. - "A Heuristic Approach to Alternate Routing in a Job Shop" unpublished Master's Thesis, Sloan School of Management, MIT, 1965

SASIENI, M., YASPAN, A. et FRIEDMAN, L. - "Operations Research : methods and problems" ; John Wiley & Sons, Inc; New York, 1959

SCHILD, A. et FREDMAN, I.J. - "Scheduling Tumbs with Linear Loss Functions" Management science, vol. 7 n° 3, avril 1961

SIMON, H.A. - "The New Science of Management Decision" ; Harper and Bros. New York 1960

SIMON, H.A. et NEWELL, A. - "Heuristic Problem Solving : the Next Advance in Operations Research" ; Journal of the O.R.S.A. vol 6 n° 1, janvier 1958

SISSON, R.L. - "Sequencing in Job Shops" ; Journal of the O.R.S.A., vol 7, 1959

SISSON, R.L. - "Sequencing Theory" ; Ackoff, R.L. chap. 7

SMITH, W.E. - "Various Optimizers for Single-State Production" ; Naval Research Logistics Quarterly, vol. 3 n° 1 mars 1956

STARR, M.K. - "Production Management : Systems and Synthesis" ; Prentice Hall, Inc. Englewood Cliffs N.J. 1964

STENGEL, J. - "Recherche dans le domaine des processus heuristiques de décision" ; dans les Actes du 2ème Congrès international de Recherche Opérationnelle, organise par "the international Federation of Operational Research Societies", à Aix en Provence en 1960 ; Dunod 1961

STENGEL, J. - "Les systèmes informatiques de programmation économique" ; "Monographies d'Informatique : AFIRO n° 3 ; Dunod, 1968

STORY, A.E. et WAGNER, H.M. - "Computational Experience with Integer Programming in Job Shop Scheduling" ; Muth, J.F. et Thompson G.L. chap. 14

TONGE, F.M. - "The Use of Heuristic Programming in Management Science" Management Science vol. 7, n° 3, avril 1961

WAGNER, H.M. - "An Integer Linear Programming Model for Machine Scheduling" ; Naval Research Logistics Quarterly vol. 6 n° 2 Juin 1959

N.B. Journal of the O.R.S.A. : The journal of the Operations Research Society of America.

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX-NAMUR

FACULTE DES SCIENCES ÉCONOMIQUES ET SOCIALES

Année Académique 1968-1969

L'ORDONNANCEMENT NON UNITAIRE, PROBLEME HEURISTIQUE

(ANNEXES)

Daniel KERVYN de MEERENDRÉ

Mémoire présenté en vue de l'obtention
du Grade de Licencié en Sciences Economiques
et Sociales (Gestion d'entreprise)

Jury du Mémoire :

MM. F BODART
L. DERWA

TABLE DES MATIERES

ANNEXE 1 : Liaison entre l'ordonnancement et d'autres problèmes d'entreprise

Section 1 : Résolution d'un problème d'investissement par une méthode d'ordonnancement.

En rapport avec le dernier chapitre de la première partie traitant des liaisons entre le problème d'ordonnancement avec d'autres problèmes d'entreprise, nous voudrions donner un exemple de cette relation ; cet exemple a trait à l'investissement et au remplacement des équipements en particulier [1].

Supposons quatre commandes pouvant être traitées par quatre machines différentes ; chaque commande ne comporte qu'une seule opération ; les coûts de préparation-machine sont tellement élevés qu'il n'est pas question de traiter une opération sur plusieurs machines. Le problème consiste à assigner une machine à chaque commande dans le but de minimiser les coûts.

Les coûts de traitement sont repris dans une matrice dont l'élément caractéristique représente le coût de la commande i traitée par la machine j. Que le coût de traitement d'une commande varie selon la machine provient soit d'une différence d'âge entre des machines par ailleurs identiques, soit de différences techniques entre ces machines. La matrice originale des coûts est la suivante :

	M_A	M_B	M_C	M_D
C_1	3	5	7	4
C_2	6	4	7	2
C_3	2	5	3	5
C_4	8	2	6	1

(Les coûts sont exprimés en dollars)

Si l'objectif consiste à minimiser le coût de traitement, on remarquera que :

- la commande 1 devrait être traitée par la machine A ainsi que la commande 3 : d'où un conflit de passage de plusieurs commandes sur une même machine.
- la meilleure assignation aux machines B et D concerne la commande 4 : d'où un conflit de passage d'une même commande sur plusieurs machines.

La résolution de ce problème consiste à dériver les coûts d'opportunité : pour ce faire, on soustrait de chaque colonne la valeur minimum de cette colonne (ex : pour la colonne A, on soustrait 2). D'un point de vue économique, cela signifie que l'assignation A-3 entraîne un coût d'opportunité nul ce qui n'est pas le cas pour le couple A-1 (3-2=1).

---

[1] Nous nous référons pour les données des deux exemples qui suivent (section 1 et 2) à STARR, M.K. : op. cit. p. 413 ss.

La matrice des coûts d'opportunité devient :

$$\begin{array}{c|cccc|}
 & M_A & M_B & M_C & M_D \\
\hline
C_1 & 1 & 3 & 4 & 3 \\
C_2 & 4 & 2 & 4 & 1 \\
C_3 & 0 & 3 & 0 & 4 \\
C_4 & 6 & 0 & 3 & 0 \\
\hline
\end{array}$$

Le problème serait résolu s'il n'y avait qu'un élément nul par ligne ET par colonne. L'absence de cette condition témoigne de la subsistance de conflits (exemple : la commande 3 pourrait se voir associée à la machine A ou C). Dans ce cas, nous passons à l'étape suivante : elle est identique à la première, sauf que nous allons rechercher les coûts d'opportunité d'une commande en fonction des machines au lieu des coûts d'opportunité d'une machine en fonction des commandes. La matrice à double coûts d'opportunité se présente alors de la façon suivante (notons qu'une ligne contenant déjà un élément nul reste identique) :

$$\begin{array}{c|cccc|}
 & M_A & M_B & M_C & M_D \\
\hline
C_1 & 0 & 2 & 3 & 2 \\
C_2 & 3 & 1 & 3 & 0 \\
C_3 & 0 & 3 & 0 & 4 \\
C_4 & 6 & 0 & 3 & 0 \\
\hline
\end{array}$$

Par construction, cette matrice contient au moins un élément nul par ligne ET par colonne : l'assignation d'une commande à une machine est donc possible. [1]

En reprenant les données originales, l'assignation est la suivante :

la commande ...	est associée à la machine ...	: coût
1	A	3
2	D	2
3	C	3
4	B	2

coût total minimum = 10

---

[1] Dans notre exemple, l'assignation était unique et non ambigüe, ce qui n'est pas toujours le cas : voir STARR, M.K., op. cit. p.415 ss. et SASIENI, M., YASPAN, A. et FRIEDMAN, L; : Operation Research : Methods and Problemns; New York, John Wiley & Sons, Inc., 1959 p. 189-190

Jusqu'à présent, nous n'avons résolu qu'un problème d'ordonnancement. En quoi peut-elle nous aider à résoudre le problème du remplacement d'un équipement? En plus d'une concurrence quant aux assignations, il peut y avoir concurrence quant à l'affectation de fonds à investir :
Si des fonds sont disponibles, vaut-il la peine de les affecter à l'achat d'une nouvelle machine? Si oui, laquelle faut-il remplacer? A notre matrice originale nous ajoutons une nouvelle ligne et une nouvelle colonne : les coûts de traitement d'une des commandes sur cette machine seront inscrits dans la nouvelle colonne. La ligne supplémentaire représente une commande imaginaire, dont les coûts de traitement sur les différentes machines sont nuls, ce qui témoigne de notre indifférence quant à l'assignation de cette commande à une machine particulière. Si nous résolvons ce problème comme il est indiqué, deux situations sont possibles :

- la commande imaginaire se voit associée à la nouvelle machine : dans ce cas, aucune machine n'est à remplacer et aucun investissement n'est à effectuer ;
- une commande originale se voit associée à la nouvelle machine : dans ce cas, le coût de traitement sera minimé en remplaçant la machine qui devait originellement traiter cette commande par la nouvelle.

Section 2 : Liaison entre l'ordonnancement, les exigences du marché et les quantités à produire par chaque machine.

Contrairement à l'exemple précédent, où la contrainte d'un temps de préparation extrêmement coûteux empêchait le traitement d'une même opération par plusieurs machines, les questions qui se posent maintenant sont les suivantes :

- comment respecter les exigences du marché étant donné un nombre limité de machines, un taux de production spécifique à chacune d'elle, des limites en disponibilité de celles-ci et un profit maximum à réaliser?
- comment répartir les commandes sur les machines et quelles sont les quantités à produire par chacune d'elles?

Bien qu'il ne s'agisse pas d'un problème d'ordonnancement au sens strict, il intervient du fait de l'existence de machines en nombre limité, de contraintes temporelles et de contraintes disjonctives (empêchant plusieurs commandes d'être traitées simultanément par une même machine). Comme il n'y a qu'une opération par commande, il n'y a pas de contrainte de routing.

Les machines n'étant pas identiques (performances, âge), leur taux de productivité et le nombre d'heures productives disponibles peut varier d'une machine à l'autre. La matrice ci-dessous reprend les taux d'activité (nombre de pièces d'une commande particulière produite par une machine par unité de temps), les exigences de la demande (en nombre de pièces) et le nombre d'heures-machine disponibles par semaine :

	M_1	M_2	M_3	M_4	DEMANDE
C_A	3	6	4,8	3,6	300
C_B	3,5	7	5,6	4,2	210
C_C	3	6	4,8	3,6	240
C_D	15	30	24	18	1800
C_E	12,5	25	20	15	400
Dispo ible (heures) :	40	60	80	$41\frac{2}{3}$	

MATRICE DES TAUX D'ACTIVITE

Nous disposons également d'une matrice des profits. Le profit unitaire se compose de trois éléments :

- le prix de vente d'une unité de la commande i : p_i
- le coût des matières nécessaires à la production d'une unité de i : c_i
- le coût de traitement d'une unité de i par la machine j : c_{ij}

Le profit unitaire de la commande i traitée par la machine j est donc égal à $P_{ij} = p_i - c_i - c_{ij}$. La matrice des profits se présente donc de la manière suivante (en dollars) :

	M_1	M_2	M_3	M_4
C_A	1,50	2,80	2,50	2,50
C_B	1,69	2,80	2,50	2,54
C_C	2,20	3,50	3,20	3,20
C_D	1,60	1,86	1,80	1,80
C_E	1,32	1,63	1,56	1,56

MATRICE DES PROFITS UNITAIRES

Comme les différentes données ne sont pas dimensionnellement homogènes, il faut les ramener à des "standards" pour pouvoir les manipuler. La première chose à faire consiste à désigner une machine "standard" ce qui permettra de comparer entre elles, les productivités des différentes machines. L'exemple est choisi de telle sorte que cette comparaison soit simple, ce qui n'est pas toujours le cas. Ce concept de la "machine-standard" est utilisé pour indiquer que les productivités relatives des machines sont proportionnelles les unes aux autres, quelles que soient les commandes à traiter. On voit que la machine 2 est deux fois plus productive que la machine 1 et que la machine 4 à 3/4 de la productivité de la machine 3. En prenant, arbitrairement, la machine 2 comme "machine-standard", les proportions pour chaque machine sont respectivement de 1/2, 1 (machine-standard), 4/5 et 3/5.

Si nous combinons les deux matrices ci-dessus et corrigeons les données pour les mettre sous forme standardisée, on obtient la matrice des profits par heure de machine-standard. Si nous regardons du côté de l'analyse dimensionnelle, nous obtenons :

$$\frac{\text{profit}}{\text{pièce}} \times \frac{\text{pièces}}{\text{heure standard}} = \frac{\text{profit}}{\text{heure standard}}$$

Cette matrice s'obtient en multipliant chaque ligne de la matrice des profits unitaires par la productivité correspondante de la machine-standard : la première ligne est donc multipliée par 6, la seconde par 7 etc ... En agissant de la sorte, nous avons négligé les productivités respectives de chaque machine : c'est ici qu'interviennent les limites ayant trait au nombre d'heures disponibles par machine.

Machine	heures disponibles par semaine	coefficient d'utilisation	heures disponibles corrigées	proportion standard	heures disponibles de machine-standard
1	40	90 %	36	0,5	18
2	60	90 %	54	1,0	54
3	80	100 %	80	0,8	64
4	$41\frac{2}{3}$	80 %	$33\frac{1}{3}$	0,6	20
TOTAL :	$221\frac{2}{3}$		$213\frac{1}{3}$		156

Le coëfficient d'utilisation tient compte des pannes éventuelles et des ajustements nécessaires, autant de facteurs restreignant le nombre d'heures productives d'une machine. Par l'intermédiaire de la colonne "proportion standard", nous tenons compte de la productivité réelle de chaque machine mais calculée en fonction de la machine-standard.

Les exigences du marché doivent encore être incorporées dans ce modèle. Si nous divisons le nombre de pièces demandées par commande par le nombre de pièces réalisées en une heure par la machine-standard, nous obtenons le nombre d'heures standard demandées par chacune de ces commandes : soit respectivement 50, 30, 40, 60, 16 heures standards.

Si en théorie, le choix de la machine-standard est arbitraire, il est préférable, en pratique de choisir la machine dont la productivité est la plus élevée car le nombre d'heures standard nécessaires pour réaliser toutes les commandes constitue alors un minimum. On constate que la demande s'élève à 196 heures, tandis que l'offre s'élève à 156; cela signifie que les commandes ne pourront pas toutes être satisfaites au cours de la période envisagée. Pour absorber ces 40 heures, nous ajoutons une colonne supplémentaire dans la matrice des "profits par heure de traitement de la machine-standard". Comme dans la section précédente, cette colonne représente une machine imaginaire, à productivité nulle et les commandes assignées à cette machine ne seront pas satisfaites.

Une fois les données standardisées, deux méthodes de résolution peuvent être utilisées : la programmation linéaire en nombres entiers [1] ou un algorithme de la théorie des graphes : "North-West Corner Allocation". Utilisé par Starr, cet algorithme repose sur le principe suivant : on répartit les heures dans une matrice de telle sorte que le total des heures demandées et offertes soient respectées; pour ce faire, on part du coin nord-ouest de la matrice en y inscrivant le plus petit des deux nombres représentant ces heures, le reste étant inscrit dans une colonne ou dans une ligne suivante.
Il est évident que cette méthode défavorise les dernières commandes de la liste.

Soit la matrice des profits par heure-standard machine (en dollars) :

	M_1	M_2	M_3	M_4	Imaginaire	heures-standards demandées
C_A	9	16,8	15	15	0	50
C_B	11,8	19,6	17,8	17,8	0	30
C_C	13,2	21	19,2	19,2	0	40
C_D	48	55,8	54	54	0	60
C_E	33	40,8	39	39	0	16
Heures-standards offertes :	18	54	64	20	40	Somme = 196

Après la première étape de l'algorithme, l'assignation (en heures-standards) des commandes aux machines est la suivante :

	M_1	M_2	M_3	M_4	Imaginaire	Demande (=Somme par ligne)
C_A	18	32	0	0	0	50
C_B	0	22	8	0	0	30
C_C	0	0	40	0	0	40
C_D	0	0	16	20	24	60
C_E	0	0	0	0	16	16
Offre : (=Somme par colonne)	18	54	64	20	40	= 196

---

[1] Le même type de problème est résolu par programmation linéaire chez EILON, S., op.cit. p. 345 ss. Il complique le modèle en y incorporant la possibilité d'heures supplémentaires et de sous-traitance.

La seconde partie de cet algorithme consiste à réarranger ces assignations en calculant la différence de profit qui en résulte et cela jusqu'à ce qu'il ne soit plus possible d'augmenter le profit. Le résultat obtenu, il ne reste qu'à reconvertir les données standardisées en données originales, en effectuant les opérations inverses à celles faites pour standardiser les données originales.

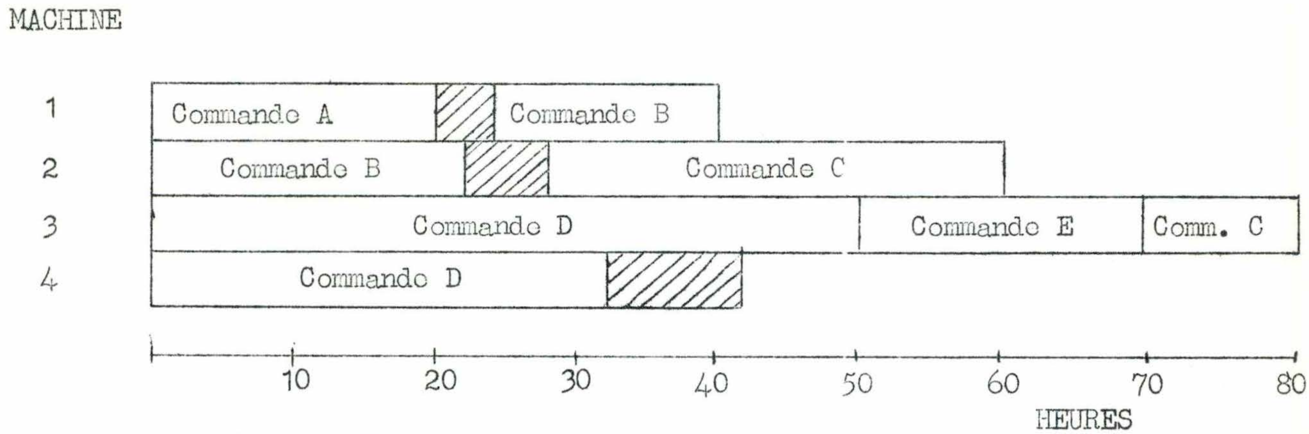
En reprenant notre exemple, la matrice des assignations (en heures-standard) optimales est la suivante :

	M_1	M_2	M_3	M_4	Imaginaire	Demande
C_A	10	0	0	0	40	50
C_B	8	22	0	0	0	30
C_C	0	32	8	0	0	40
C_D	0	0	40	20	0	60
C_E	0	0	16	0	0	16
Offre :	18	54	64	20	40	= 196

La solution finale, en données originales est donnée par la matrice ci-dessous : (en unités produites)

	M_1	M_2	M_3	M_4	Unités produites	Unités demandées
C_A	60	0	0	0	60	300
C_B	56	154	0	0	210	210
C_C	0	192	48	0	240	240
C_D	0	0	1200	600	1800	1800
C	0	0	400	0	400	400

Il s'agit donc de la solution optimale du problème posé, ce qui ne signifie pas que l'ordonnancement qui en résulte soit optimal.



Remarques quant à cet ordonnancement :

- les contraintes disjonctives sont respectées ;
- le nombre d'heures pendant lequel une commande i est traitée par la machine j s'obtient en divisant le nombre de pièces à réaliser par cette machine (solution optimale) par la productivité de cette machine (cfr. premier tableau)
- les heures d'indisponibilité-machine (selon le pourcentage d'utilisation donné ci-dessus) sont hachurées.
- cet ordonnancement vaut pour une semaine et respecte les données relatives au nombre d'heures productives disponibles d'une machine particulière.

Section 3 : Détermination de la charge optimale d'un équipement [1]

Parmi les hypothèses restreignant l'application des modèles d'ordonnancement (chap. 1 de la seconde partie), il en existe une où l'on précise qu'on ne se préoccupe ni du découpage du mois en jours ou du jour en heures. En d'autres termes, l'ordonnancement ne répond pas aux questions suivantes :

- faut-il accepter toutes les commandes qui nous parviennent ?
- quels sont les besoins en personnel et/ou en équipement nécessaires pour satisfaire les commandes acceptées ?

Il faut donc, avant d'établir l'ordonnancement proprement dit avoir une idée du volume de travail à réaliser, en tenant compte des capacités de l'équipement [2]. Ce rapport entre volume de travail et capacités des machines s'appelle le coefficient d'utilisation : si le volume de travail présent dans l'atelier est en moyenne de 160 heures, et la capacité des machines de 120 heures, il faudra soit encourir des coûts de retards, soit recourir aux heures supplémentaires et/ou à de nou-

---

[1] EILON, S. : op. cit. p. 350 ss.

[2] Nous nous occuperons ici de l'équipement uniquement ; une argumentation similaire est utilisée dans l'annexe suivante à propos de la main-d'oeuvre.

veaux investissements, soit refuser des commandes. Comme une entreprise, en général, répugne, pour des raisons de goodwill, à refuser des commandes, il faut adapter l'équipement à leurs fluctuations. Encore faut-il le faire à bon escient en évitant le surinvestissement. Il s'agit donc de trouver une capacité ou un coefficient d'utilisation optimal. Le lien entre ce problème et l'ordonnancement devient alors évident :

- Dans un problème statique, où nous connaissons les commandes à traiter au cours d'une période, le coefficient d'utilisation peut être déterminé à postériori c-à-d lorsque l'ordonnancement optimal a été trouvé.
  Supposons que nous voulions ordonnancer, pendant la période t, x tâches et que l'ordonnancement optimal donne une durée d'occupation maximale de 50 heures ; cela signifie qu'il existe au moins une machine qui ne peut terminer son travail en moins de 50 heures. Si la capacité théorique d'une machine de ce type est de 10 heures (pour la même période t), trois situations peuvent se présenter :
  - l'entreprise possède moins de 5 machines de ce type, auquel cas il s'avère impossible de terminer les travaux au cours de la période envisagée ; il faudra reporter les travaux inachevés à la période t + 1. Si le volume de travail est une moyenne, il faut envisager un investissement.
  - l'entreprise possède 5 machines de ce type : le coefficient d'utilisation s'élève à un. Aucun retard n'est subi ; reste à savoir si l'utilisation maximale des machines n'entraîne pas une obsolescence rapide.
  - l'entreprise possède plus de 5 machines de ce type : nous sommes dans une situation théorique de surinvestissement ; elle est théorique car une étude technique et économique peut révéler que la capacité souhaitable représente 80 % de la capacité maximale d'une machine : l'ordonnancement optimal serait réalisé par l'utilisation de 7 machines. Un nombre plus important de machines constituerait un surinvestissement réel.

  L'ordonnancement permet donc dans le cas statique de dériver un coefficient d'utilisation optimal.

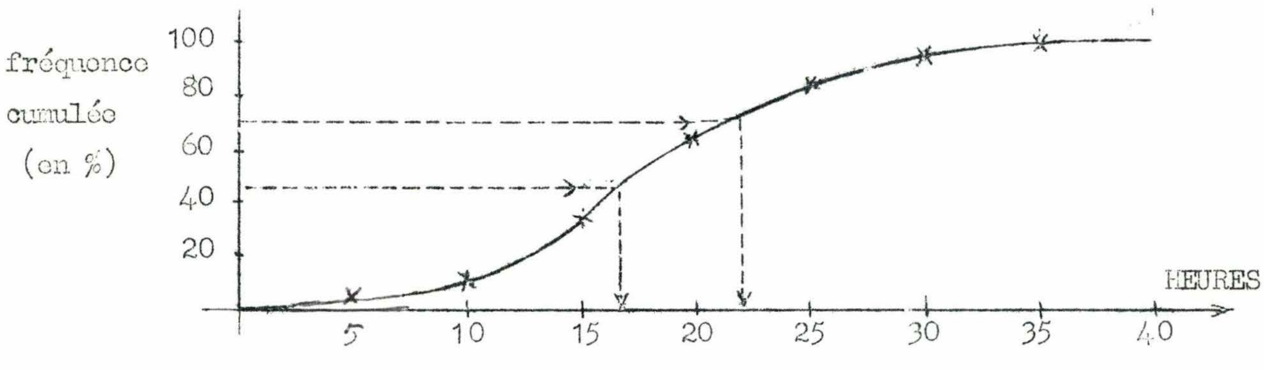
- Dans un problème dynamique, il est impossible de dériver un coefficient d'utilisation ne fût-ce que théorique ; cela vient de ce que les données concernant les commandes (durées, date d'arrivée ...) ne sont pas connues avec certitude : le processus est continu. De constant, le coefficient d'utilisation devient une variable. Dans le problème statique, il apparaissait comme une conséquence de l'ordonnancement ; dans le cas dynamique, il est inséré dans le modèle d'ordonnancement car il agit comme contrainte. Puisqu'il est calculé à priori, il s'agit de le déterminer indépendamment de l'ordonnancement.

Supposons que les commandes arrivent selon certaines lois de probabilité, que leur durée est aléatoire et qu'il n'y a qu'un type d'équipement. Lorsque l'objectif réside dans la minimisation du coût total, il faut déterminer la capacité optimale de cet équipement. Par la technique de simulation, nous allons simuler, sur base d'expériences antérieures, les réactions du système aux variations de certains paramètres.

Supposons que le nombre de commandes reçues journalièrement varie entre 1 et 10 selon une distribution uniforme ; la durée de traitement de ces commandes varie de la manière suivante :

Heures Machine	fréquence en %	Heures-machine cumulées	fréquence cumulée en %
0 - 5	4	0 - 5	4
5,01 - 10	6	0 - 10	10
10,01 - 15	24	0 - 15	34
15,01 - 20	27	0 - 20	61
20,01 - 25	22	0 - 25	83
25,01 - 30	11	0 - 30	94
30,01 - 35	5	0 - 35	99
35,01 - 40	1	0 - 40	100

La courbe des fréquences cumulées se présente alors ainsi :



Par l'intermédiaire d'une table de nombres aléatoires, on simule le nombre de commandes arrivant journalièrement (un 0 dans la table correspond à 10 arrivées) soit par exemple : 6,10,2,7,1,5,1,8,3,... Reste à simuler la durée de traitement de ces commandes : les nombres aléatoires sont pris par groupe de deux et sont supposés représenter les fréquences cumulées ; il suffit alors de lire la valeur correspondante en abcisse.

Par exemple :

jour	nombre de commandes	heures machine par commande	heures machine totales / jour
1	6	10, 32,5, 18, 14, 27,	106
2	10	7, 13, 21	219
3	2	....	41
4	7	....	239
5	1	....	36
6	5	....	168
7	1	....	34
8	8	....	292
9	3	....	116
..	..	....	...

La dernière colonne représente la charge journalière de l'équipement. En supposant que le traitement d'une commande reçue le jour j peut commencer au jour j + 1, le problème est le suivant :

- ou bien on ne veut aucun retard dans le traitement des commandes ; dans ce cas, il faut établir la capacité de l'équipement à 400 heures par jour ce qui correspond à la charge maximum : 10 commandes et une durée de traitement de 40 heures par commande. Cette politique entraine des coûts importants puisque l'équipement risque de rester très longtemps inoccupé.
- ou bien on ne veut aucun coût d'inutilisation-machine, ce qui n'arrivera qu'en l'absence d'équipement, et ce qui entrainera des coûts de retards.

Il s'agit donc de trouver un équilibre entre ces deux types de coûts. On peut partir d'une charge égale au volume moyen de travail (139 ici) mais cette estimation ne serait exacte que si le coût de retard horaire était égale au coût horaire d'inoccupation. Selon que le coût de retard est plus (moins) élevé que le coût d'inoccupation, la charge optimale sera plus (moins) élevé que le volume moyen de travail.

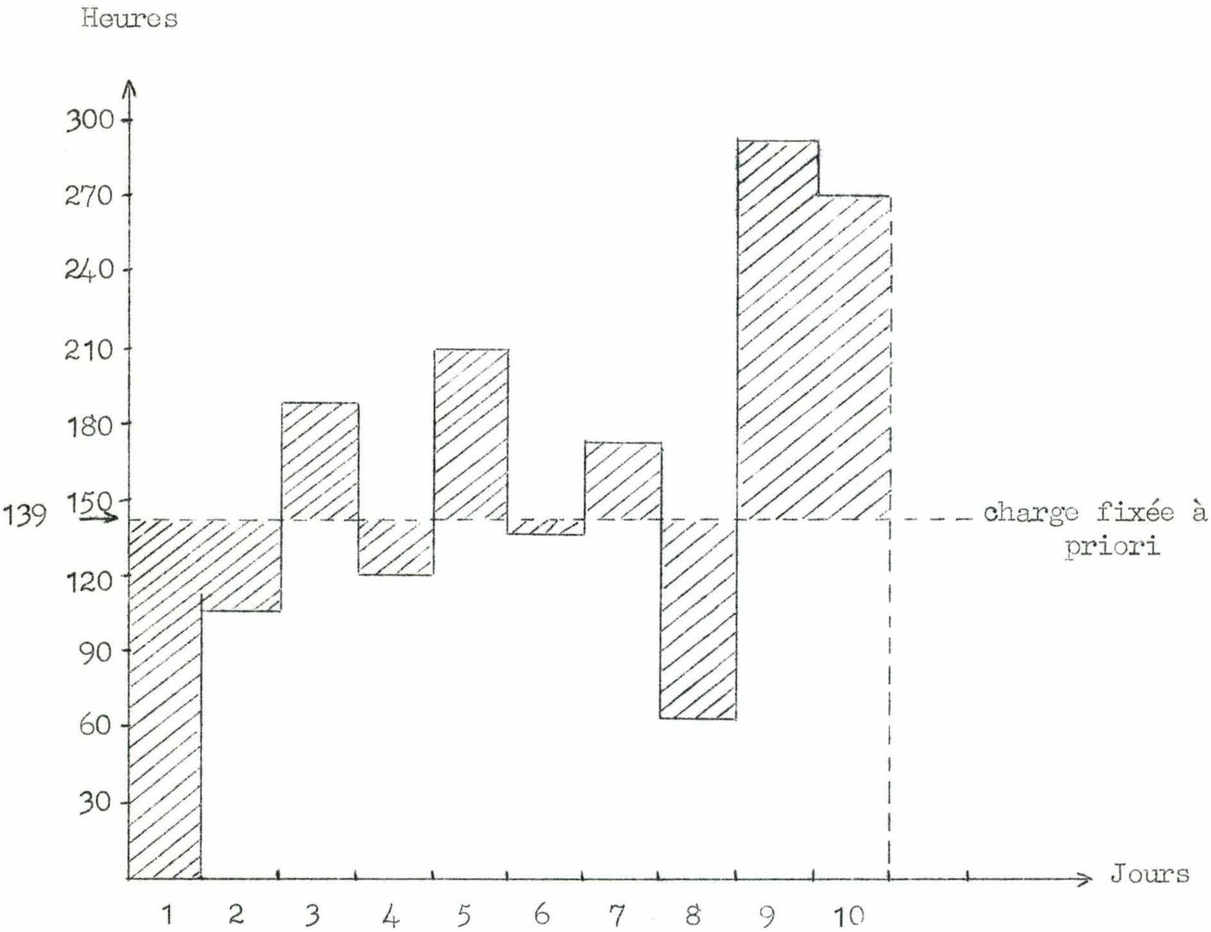
Comment évaluer cette charge optimale par simulation ? On dresse un tableau exprimant le nombre d'heures de traitement en attente, en fonction d' une capacité fixée, par exemple à 139 heures. En reprenant nos données, nous aurons :

- jour 1 : 139 - 0 = 139 heures d'inoccupation-machine [1]
- jour 2 : 139 - 106 = 33 heures d'inoccupation
- jour 3 : 139 - 219 = 80 Cela correspond à 80 heures de retards, à rattraper le lendemain.
- jour 4 : 139 - 80 - 41 = 18 heures d'inoccupation
- ..... : ....

Si le coût de l'heure-machine inoccupée s'élève à 5 et le coût de retard horaire à 1, on peut simuler la courbe de coût total en fonction d'une capacité fixée (ici 139). On dresse un diagramme reprenant le nombre d'heures de traitement journalier. Au-dessus de la capacité envisagée se trouvent le nombre d' heures de retard, en dessous le nombre d'heures d'inoccupation. Pour les 10 premiers jours, nous aurons :

---

[1] Nous avons supposé que l'équipement était libre au départ et qu'une commande reçue le jour j ne pouvait débuter qu'au jour j + 1.



En faisant varier le niveau de charge, on obtient les résultats suivants :

niveau de charge	total des heures inoccupées machine	total d'heures d'attente (retard)	coût machine	coût retard	coût total
0	0	1613	0	1613	1613
60	60	1073	300	1073	1373
62	62	1055	310	1055	1373
63	63	1046	315	1046	1361
64	65	1038	325	1038	1363
80	97	910	485	910	1395
100	137	750	685	750	1435
139	268	492	1340	492	1832

Remarques et conclusions :

- Au fur et à mesure que le niveau de charge s'accroît, le nombre d'heures d'inoccupation-machine s'accroît de façon non linéaire, tandis que le nombre total d'heures d'attente décroît et tend vers 0.

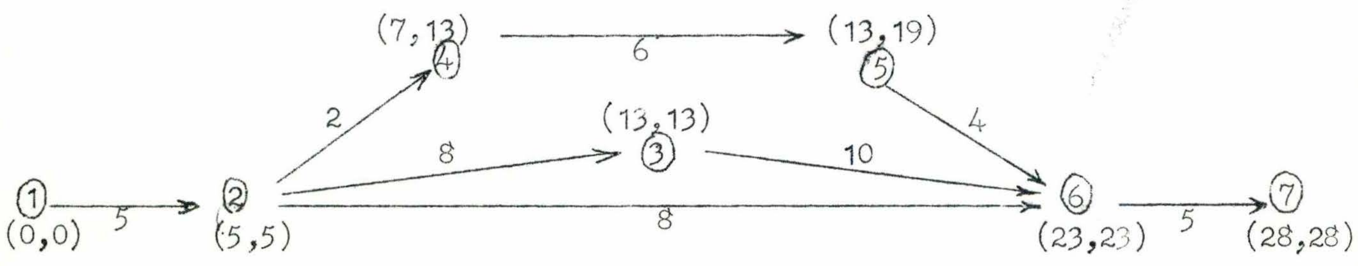
- Le coût d'inoccupation étant nettement plus élevé que le coût de retard, il est logique de trouver un niveau de charge optimal nettement inférieur au volume moyen des travaux à réaliser.
- Selon le niveau de charge réel de l'atelier, il y aura sur (sous) investissement suivant qu'il se situe au dessus (en dessous) de 63.
- Dans une étude ultérieure, on pourrait se poser la question de l'effet d' une modification des règles de priorité. Dans notre exemple, nous avons supposé que les travaux étaient effectués au fur et à mesure de leur arrivée, que les coûts de retards variaient linéairement avec le temps, que les retards ne commençaient à compter qu'à partir du surlendemain de la réception des commandes ... Ces questions sont résolues par l'ordonnancement qui risque fort de modifier ce niveau de charge optimal.

ANNEXE 2 : Le diagramme de Gantt

Henry L. Gantt (1861-1909) se souciait fort peu des problèmes d'ordonnancement, vus sous leur aspect théorique ; il cherchait un moyen rapide de comparaison entre un ordonnancement théorique et la réalité.

Le diagramme auquel il a donné son nom se compose essentiellement d'une échelle de temps, sur laquelle figurent les opérations à réaliser, qui sont chacune représentées par une ligne. La longueur d'une ligne est proportionnelle à la durée de traitement de l'opération correspondante ; elle est localisée sur l'axe des temps selon les résultats de l'ordonnancement optimal théorique. Un index mobile, rattaché à cette échelle de temps, indique la date du jour et permet de vérifier rapidement l'état <u>théorique</u> d'avancement des travaux et par la même de comparer cet état avec l'état d'avancement <u>réel</u>.

Prenons l'exemple d'un graphe PERT, où le "projet" consiste dans la construction d'un grand ensemble : celle-ci se compose d'opérations aussi diverses que la construction des fondations et des murs, la pose du toit, l'installation des conduites d'eau, du système d'éclairage etc ... Si certaines opérations doivent se suivre, pour des raisons technologiques, (la pose du toit se fait normalement après la construction des fondations), d'autres peuvent être simultanées (l'installation de l'éclairage peut être faite en même temps que les travaux de menuiserie). Par la <u>méthode du "chemin critique"</u>, on détermine les dates auxquelles ces opérations doivent (pour les opérations critiques) ou peuvent (pour les autres, ce qui entraine des marges) commencer. On aboutit ainsi à la construction du graphe potentiel-étape :



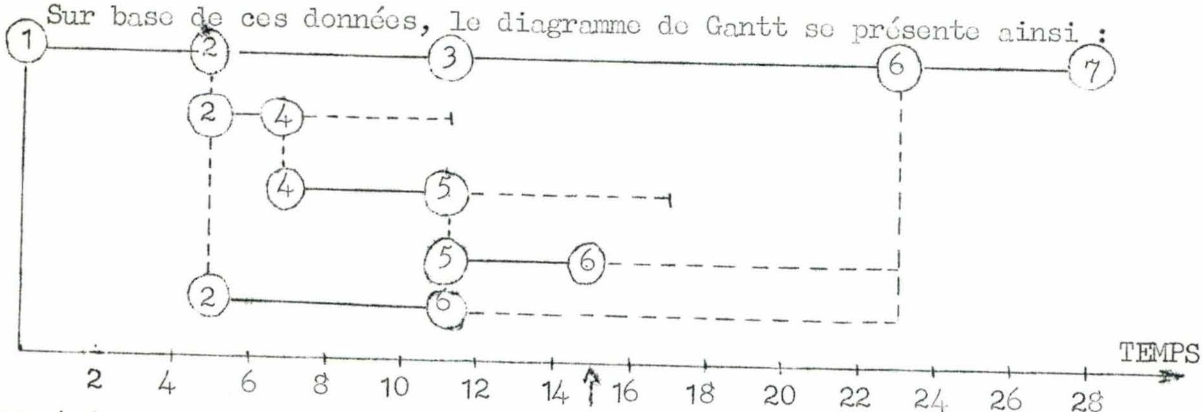
Une opération est représentée par un arc, sur lequel se trouve sa durée de traitement. Chaque arc relie 2 sommets, le sommet initial (i) et le sommet terminal (j). Les valeurs associées à ces sommets-entre parenthèses sur le graphique - sont les dates au plus tôt et au plus tard auxquelles les opérations issues de ce sommet peuvent ou doivent commencer.
Dans notre exemple, le chemin critique relie les sommets 1 - 2 - 3 - 6 - 7 .
Si $\bar{t}i$ est la date au plus tôt des opérations issues du sommet i, $\bar{\bar{t}}i$ la date au plus tard de ces mêmes opérations et a_{ij} la durée d'une opération particulière, on déduit :

- la marge d'étape : $\bar{\bar{t}}i - \bar{t}i$
- la marge libre d'une opération : $\bar{t}j - (\bar{t}i + a_{ij})$
- la marge totale d'une opération : $\bar{\bar{t}}j - (\bar{t}i + a_{ij})$

Par construction, toutes ces marges sont nulles pour les opérations critiques ; la marge totale des opérations non critiques est :

opération	marge totale
2 - 4	6
2 - 6	10
4 - 5	6
5 - 6	6

Sur base de ces données, le diagramme de Gantt se présente ainsi :



Commentaires :

- nous avons calé toutes les opérations à gauche ce qui signifie qu'elles commencent toutes à leur date au plus tôt.
- les pointillés verticaux expriment des contraintes de succession ; ainsi, l'opération 4-5 ne pourra commencer que lorsque l'opération 2-4 sera terminée.
- du côté horizontal, en continu, la durée de chaque opération, et en pointillé, la longueur des marges totales de ces opérations ; on s'aperçoit qu'il n'y en a pas pour les opérations critiques, que nous avons groupées sur une même ligne. La longueur totale des lignes continues et pointillées indique la distance totale sur laquelle une opération peut se mou-

- voir sans retarder le projet.
- l'index, en face du nombre 15 permet la comparaison entre l'ordonnancement théorique et la réalité, et revèle l'intérêt de disposer d'un diagramme de Gantt. Si nous sommes au 15ème jour, il faut que les opérations 1-2 et 2-3 soient terminées, car étant critiques, elles n'ont aucune marge. Bien que non critique, la tâche 2-4 doit être terminée car sa marge totale est dépassée ; si ces opérations ne sont pas terminées, l'ensemble du projet sera nécessairement en retard. Bien qu'en théorie, les opérations 2-6 et 4-5 devraient être achevées, le fait qu'elle ne le soient pas n'est pas catastrophique mais elles sont à surveiller car :
  . au jour 15, l'opération 4-5 n'a plus qu'une marge totale de 4 et sa durée est de 6 jours ; elle aura dû commencer depuis au moins deux jours.
  . au jour 15, l'opération 2-6 n'a plus qu'une marge de 8, ce qui correspond exactement à sa durée ; elle doit donc débuter au plus tard "aujourd'hui"

  Les opérations 3-6 et 5-6 devraient être commencées mais il s'agit d'une obligation pour la première d'entre elles car elle fait partie du chemin critique.

Cet exemple avait uniquement pour objectif de montrer le caractère pratique du diagramme de Gantt. Des chercheurs ont dépassé ce stade pour utiliser le diagramme dans une optique théorique. Dans la seconde partie nous l'examinons en ce qui concerne l'ordonnancement non unitaire, mais sans quitter les problèmes d'ordonnancement, ce diagramme devient indispensable lorsque les opérations à réaliser sont soumises, non seulement à des contraintes de potentiel mais aussi à des contraintes cumulatives (voir 1ère partie, chap. 4), par lesquelles on établit des limites à la disponibilité en main-d'oeuvre ou à la capacité des machines. Reprenons notre exemple en supposant que le nombre total d'ouvriers disponibles est de 16 ; chaque opération en occupe quelques uns (par hypothèse, chaque ouvrier est capable de réaliser chaque opération).
Besoins en main-d'oeuvre :

opération	main-d'oeuvre nécessaire
1-2	5
2-3	9
2-4	3
2-6	6
3-6	8
4-5	5
5-6	2
6-7	5

Sur le diagramme de Gantt, on indique pour chaque unité de temps, le total de main-d'oeuvre employée ; ce total dépasse souvent la capacité totale et il faut alors modifier l'ordonnancement optimal

- on commence par manipuler les opérations qui ont des marges et cela dans les limites de ces marges. A chaque manipulation, on calcule le total de main-d'oeuvre nécessaire ; le problème est résolu si ce total ne dépasse à aucun moment la disponibilité maximale ;

- s'il est impossible de respecter et les marges libres et les contraintes cumulatives, on doit allonger le chemin critique (dont la longueur est l'ordonnancement optimal ce qui accroît toutes les marges) ; on l'allonge d'une unité à la fois car ce retard entraine des coûts et on vérifie à chaque fois si cette modification permet de respecter les disponibilités en main-d'oeuvre.

Dans une perspective plus dynamique, on peut se demander quel est le nombre minimum d'ouvriers nécessaire, tout en réalisant l'ordonnancement optimal (c-a-d sans toucher au chemin critique). On peut trouver un équilibre entre le coût marginal provoqué par le retard du projet et le gain marginal provoqué par la libération de la main-d'oeuvre. Cela indique à nouveau qu'un ordonnancement optimal n'est pas nécessairement à rechercher à tout prix, et que la prise en charge de l'environnement (ici la main-d'oeuvre) peut rendre l'ordonnancement sous-optimal.

Les graphiques qui suivent résument les étapes nécessaires pour trouver cet ordonnancement sous-optimal, en prenant comme critère la minimisation des coûts totaux :

1. Le premier diagramme reprend la situation originale : les valeurs sur les arcs représentent le nombre d'ouvriers nécessaires par opération ; les opérations 2-3, 4-5 et 5-6 sont groupées sur une même ligne car elles sont soumises à des contraintes de succession ; les valeurs sur l'axe des temps indiquent les totaux de main-d'oeuvre utilisée par unité de temps. On aboutit à un maximum de 20 ouvriers, ce qui est supérieur à la limite de 16.

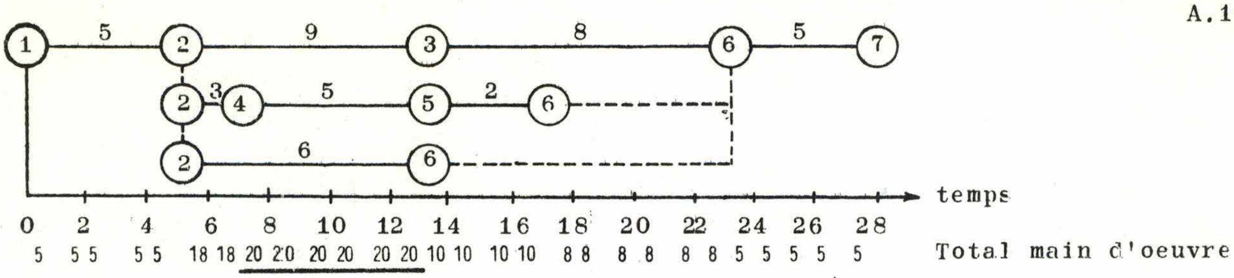
2. Pour cette raison, on modifie l'ordonnancement optimal pour respecter cette limite ; dans notre cas, l'ordonnancement reste optimal, car nous avons respecté les marges.

3. Le troisième diagramme indique, dans une optique dynamique, ce qui se passerait si le nombre d'ouvriers tombait à 14 ; cette situation entraîne un retard de 2 jours pour le projet. Supposons que le salaire journalier soit de 15 (en dollars) par ouvrier et que le coût dû au retard, de forme exponentielle, répond à la formule $C = 50\,x^2$ où x est le nombre de jours retard.

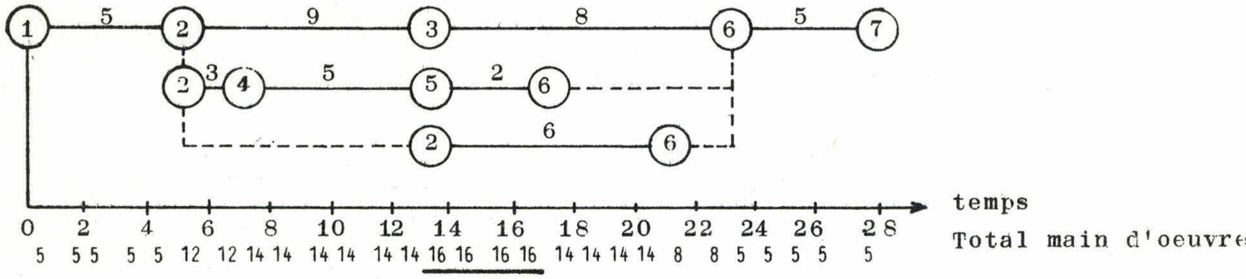
4. La situation antérieure apparait optimale car le gain provoqué par la suppression d'un ouvrier supplémentaire, réduisant leur nombre à 13 est largement compensé par le coût de retard.

En dessous de ces diagrammes, un tableau reprend les coûts totaux; un graphique montre la courbe de charge, témoignant de la nécessité d'avoir une courbe la plus lisse possible sans pour autant trop allonger l'échelle des temps.

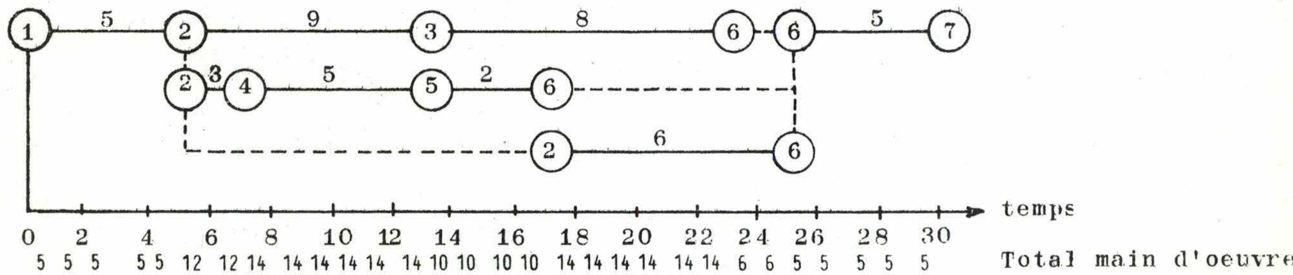
1)



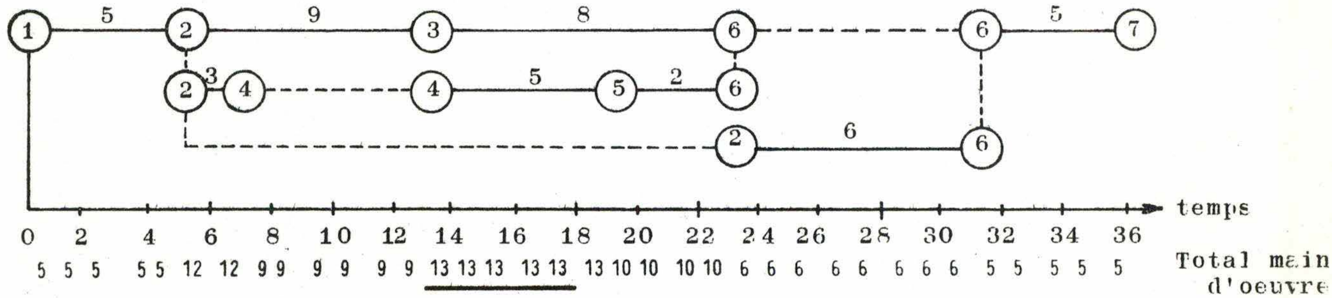
2)



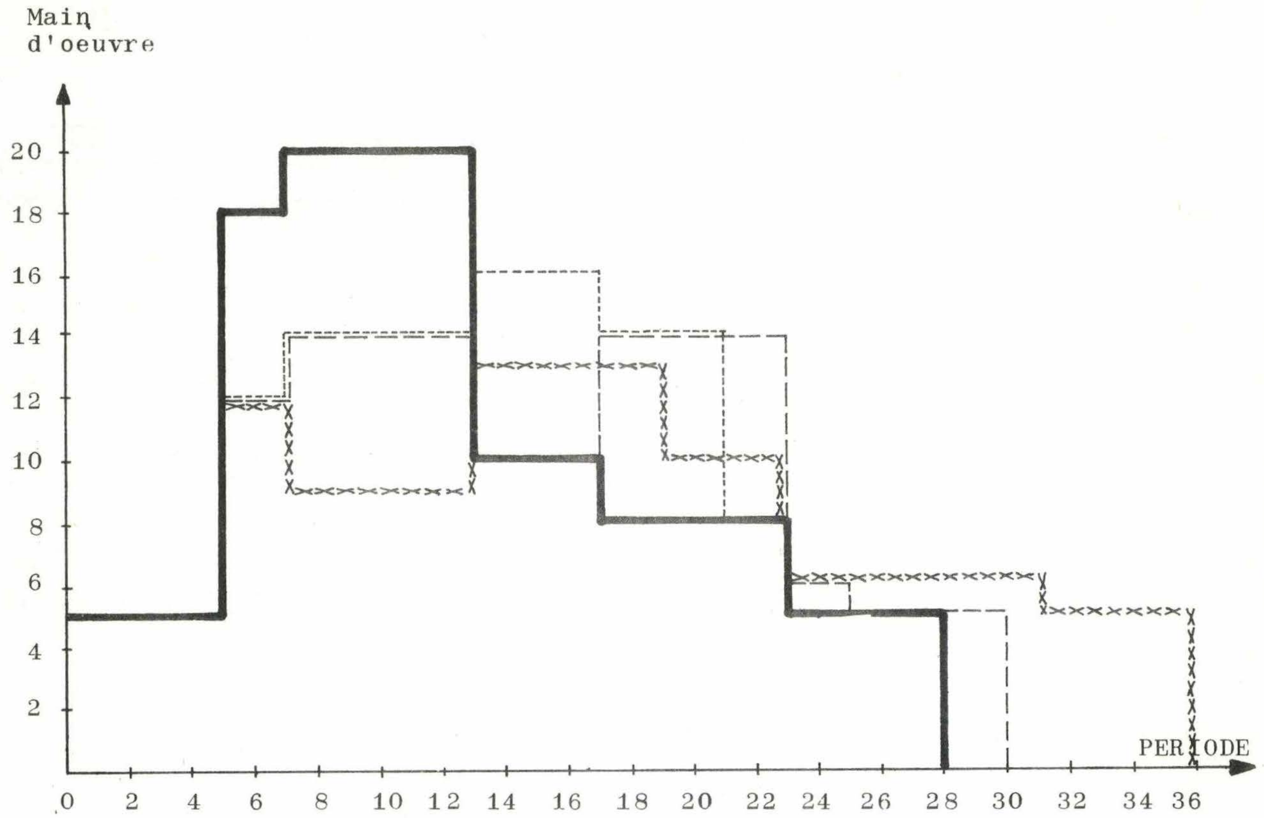
3)



4)



	Situation 1)	Situation 2)	Situation 3)	Situation 4)
Nombre de jours	28	28	30	36
Nombre d'ouvriers nécessaires	20	16	14	13
Coût salaire (15xouv.xjrs)	8400	6780	6300	7020
Coût de retard ($50x^2$)	0	0	400	3200
Coût total	8400	6780	6700	10220



Courbe de charge

Situation 1 : en trait plein
2 : en petit pointillé
3 : en grand pointillé
4 : croix

} Les parties communes à la situation 1 ne sont pas reprises.

Nombre total de jours/ouvriers nécessaires pour terminer toutes les opérations : 294 jours.

Situation	Nombre de jours de travail	Nombre d'ouvriers	Nombre de jours/ ouvriers disponibles	Nombre de jours/ ouvriers inoccupés
1	28	20	560	266
2	28	16	448	154
3	30	14	420	126
4	36	13	468	147

ANNEXE III : Développements mathématiques relatives à la seconde partie.

Section 1 : Résolution du problème $n/1/\bar{F}$

Lorsqu'on ordonnance un problème $n/1/\bar{F}$, la durée moyenne d'occupation de l'atelier est minimisée en plaçant les tâches dans une séquence telle que leurs durées de traitement forment une suite de valeurs non décroissantes

$$(p_{[1]} \leqslant p_{[2]} \cdots \leqslant p_{[n]})$$

Si on voulait maximiser $\bar{F}$, il faudrait opérer de façon inverse

Démonstration

La date à laquelle une tâche, placée en kième position sera terminée est égale à la somme des durées de traitement des tâches placées en première, deuxième, ... kième position ; d'où :

$$F_{[k]} = \sum_{i=1}^{k} p_{[i]}$$

La durée moyenne d'occupation est égale à :

$$\bar{F} = \sum_{k=1}^{n} \frac{F_{[k]}}{n} = \sum_{k=1}^{n} \frac{\sum_{i=1}^{k} p_{[i]}}{n}$$

$$\bar{F} = \frac{1}{n}\left[p_{[1]} + p_{[1]} + p_{[2]} + p_{[1]} + p_{[2]} + p_{[3]} + \cdots p_{[1]} + p_{[2]} \cdots + p_{[n]} \right]$$

$$= \sum_{i=1}^{n} \frac{(n - i + 1)\, p_{[i]}}{n}$$

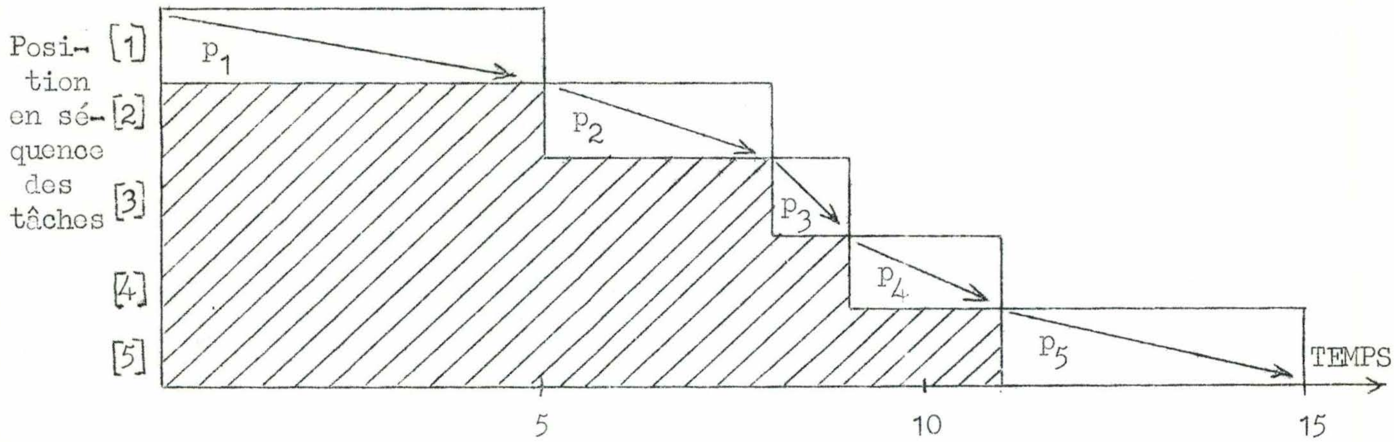
Smith [1] a démontré qu'une somme de produits de deux suites de valeurs est minimisée si une de ces suites est arrangée en ordre décroissant et l'autre en ordre croissant. Or, la suite des (n-i-1) est en ordre croissant puisque i varie de 1 à n ; $\bar{F}$ sera donc minimisé en rangeant les $p_{[i]}$ en ordre non croissant.

Représentation graphique

Soit les tâches 1 à 5 ; leurs durées de traitement sont respectivement de 5,3,1,2,4. Si on basait l'ordonnancement sur le numéro d'identification des tâches, on aboutirait au graphique suivant :

---

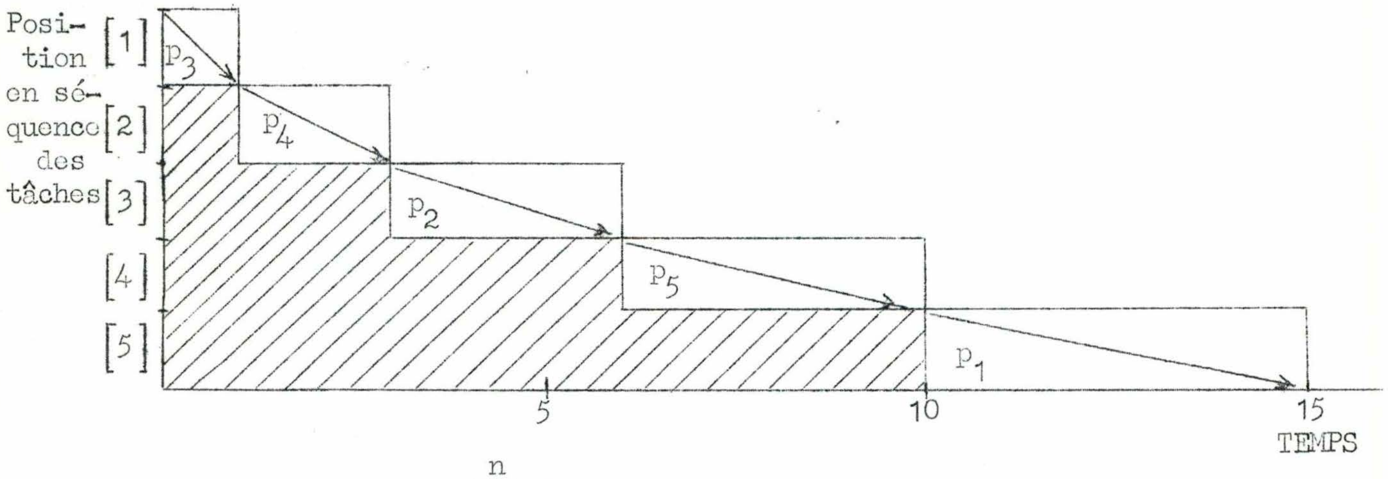
[1] SMITH W.E. : "Various Optimizers for Single-State Production" ; Naval Research Logistics Quarterly vol. 3 n° 1 ; mars 1956



La somme des durées d'occupation des 5 tâches est la somme des durées de traitement (représentées par les blocs) et des durées d'attente (partie hachurée)

Chaque bloc se caratérise par un vecteur de pente $-1/p_i$. Chaque graphique est une chaîne de tels vecteurs. On constate qu'on minimise la somme des durées d'attente lorsque cette chaîne forme une courbe convexe ce qui n'arrivera que si les opérations sont ordonnancées dans un ordre tel que : $p\ [1] \leqslant p\ [2] \leqslant p\ [3]\ \ldots \leqslant p\ [n]$

Ce n'est manifestement pas le cas dans le graphique ci-dessus. Si nous ordonnançons les opérations selon la règle de la durée de traitement la plus courte (cette règle appelée SPT "shortest processing time" est importante et nous aurons l'occasion d'y revenir), nous aboutissons à l'ordonnancement optimal suivant. (la chaîne des vecteurs-opérations forme bien une courbe convexe).



Dans ce graphique : $\sum_{k=1}^{n} F_{[k]} = 35$

$\overline{F} = 35/5 = 7.$ Dans l'ordonnancement précédent

$\overline{F} = 48/5 = 9{,}6.$

Remarques :

1. Des démonstrations similaires prouvent que la règle de la plus courte opération (SPT) minimise aussi :

   a. $\overline{C}, \overline{W}, \overline{L}, F_{min}, C_{min}, W_{max}, W_{min}$ et non nul : ces mesures d'évaluation sont minimisées parce qu'elles sont régulières vis-à vis de F ;

   b. $\overline{T}$, dans la mesure où toutes les tâches sont en retard ; dans ce cas $\overline{T}$ est égal à $\overline{L}$.

   c. le nombre moyen d'opérations présentes dans l'atelier : $\dfrac{\overline{N}(0,F_{Max})}{n} = \dfrac{\overline{F}}{F_{Max}}$; or F_{max} est constant et $\overline{F}$ est minimisé par SPT.

2. Il ne faudrait pas en conclure que la règle SPT minimise <u>toutes</u> les mesures d'évaluation qui sont fonction du temps d'occupation de l'atelier par les n opérations (F). La variance de F n'est pas nécessairement minimisée par SPT car elle n'est pas une mesure régulière à cause de l'introduction d'une différence :

$$\sigma^2(F) = \frac{1}{n}\sum_{i=1}^{n} (Fi - F)^2 = \frac{1}{n}\sum_{i=1}^{n} Fi^2 - \overline{F}^2$$

   Si SPT minimise les deux termes de l'équation, elle n'en minimise pas pour autant la différence entre les deux termes.

3. La règle antithétique à SPT qui consiste à établir un ordonnancement selon la durée d'opération la plus longue (appelée LPT, "longest processing time) maximise tout ce que SPT minimisait : cette règle se note :

$$p[1] \geqslant p[2] \geqslant \ldots \geqslant p[n]$$

Section 2 : L'ordonnancement de n tâches sur 1 machine lorsque le critère est <u>fonction des retards et de la durée moyenne d'occupation de l'atelier</u>

Lorsqu'on a une séquence basée sur la règle des dates de livraison (qui donne l'ordonnancement optimal dans le cas où le critère d'optimisation est min T_{max}) et que la valeur de ce retard maximal est nulle (ce qui signifie qu'aucune opération n'est en retard) on s'est aperçu que d'autres séquences étaient meilleures lorsqu'on voulait cumuler l'objectif de minimisation du retard maximal <u>et</u> la minimisation de la durée moyenne d'occupation de l'atelier. Le théorème dû à Smith [1] s'énonce alors de la façon suivante :

<u>THEOREME</u> : Lorsqu'on ordonnance un problème n/1 et qu'il existe une séquence telle que le retard maximum est nul, il existe une autre séquence d'opération qui minimise $\overline{F}$ et qui continue à respecter $T_{max} = 0$. Il faut et il suffit pour qu'une opération k soit mise en dernière position que :

---

[1] SMITH W.E. : op. cit.

a. $d_k \geqslant \sum_{i=1}^{n} p_i$

b. $p_k \geqslant p_i$ pour tout i dont $d_i \geqslant \sum_{j=1}^{n} p_j$

Cela veut dire qu'une opération sera mise en dernière position si elle ne cause aucun retard (condition a) et si sa durée de traitement est supérieure aux durées de traitement des autres opérations qui pourraient être mises en dernière position sans être en retard. On procède alors par récurrence pour placer les opérations qui seront en (n-1)ième position, etc. Il s'agit en fait de l'application de la règle SPT, sujette à des contraintes de dates de livraison.

Exemple :

opération	: 1	2	3	4	5	6
date de livraison	: 24	21	8	5	10	23
durée de traitement	: 4	7	1	3	2	5

- L'objectif min T_{max} serait réalisé en mettant les opérations dans l'ordre suivant : 4-3-5-2-6-1 (ordre des dates de livraison)
- L'objectif min $\bar{F}$ serait réalisé en mettant les opérations dans l'ordre suivant : 3-5-4-1-6-2 (application de la règle SPT)
- L'objectif cumulé serait réalisé par application de la règle de Smith et donnerait la séquence suivante : 3-4-5-1-2-6.

En effet, l'opération à mettre en dernière position est 6 parce que

a. $d_6 = 23 \geqslant \sum_{i=1}^{n} p_i = 22$; l'opération 1 est dans le même cas puisque $d_1 = 24$

b. la seconde condition supprime le choix qui existait entre l'opération 1 et 6 parce que la durée de traitement de l'opération 6 est supérieure à celle de l'opération 1

On continue de la même manière en éliminant l'opération 6

Règle		Opérations 1	2	3	4	5	6	T_{max}	$\bar{F}$
Date de livraison :	F_i	22	13	4	3	6	18		11
4-3-5-2-6-1	T_i	0	0	0	0	0	0	0	
SPT :	F_i	10	22	1	6	3	15		9,5
3-5-4-1-6-2	T_i	0	1	0	1	0	0	1	
Smith :	F_i	10	17	1	4	6	22		10
3-4-5-1-2-6	T_i	0	0	0	0	0	0	0	

On s'aperçoit qu'il s'agit bien d'un compromis. Par l'application de la première règle, il n'y a aucun retard mais la durée moyenne d'occupation est la plus élevée ; par application de la seconde, celle-ci est la moins élevée, mais certaines opérations sont en retard ; la troisième minimise la durée moyenne d'occupation de l'atelier, lorsqu'aucun retard n'est toléré.

Section 3 : Inégalité d'importance des tâches dans un problème n/1

§ 1 : Durée moyenne pondérée d'occupation de l'atelier : $\bar{F}u$

Dans les graphiques de la première section de cette annexe, la coordonnée verticale du vecteur représentant chaque tâche était égale à - 1 : aucune hiérarchie d'importance n'avait été établie entre les tâches. Si nous introduisons cette hiérarchie, le vecteur représentant la tâche i aura une pente de $- u_i / Pi$

Comme pour le théorème cité dans la première section :

- F_{max} est constant puisqu'il n'y a qu'une seule machine ;
- la durée totale d'occupation de l'atelier par les n tâches se décompose en deux éléments :
  - la somme des durées de traitement (constante)
  - la somme des durées d'attente.

C'est en minimisant ce dernier terme qu'un ordonnancement optimal par rapport à $\bar{F}u$ pourra être trouvé :

THEOREME : Dans un problème n/1/$\bar{F}$u, la durée d'occupation totale e l'atelier ($\sum_{i=1}^{n} u_i F_i$) et la durée moyenne d'occupation (puisque n est constant) sont minimisées en ordonnançant les opérations de telle sorte que :

$$\frac{p_{[1]}}{u_{[1]}} \leqslant \frac{p_{[2]}}{u_{[2]}} \cdots \leqslant \frac{p_{[n]}}{u_{[n]}} \qquad [1]$$

Cela signifie qu'il faut mettre en premier lieu l'opération dont le rapport entre la durée de traitement et le coefficient d'importance est le plus petit. (il s'agit donc d'une variante de la règle SPT).

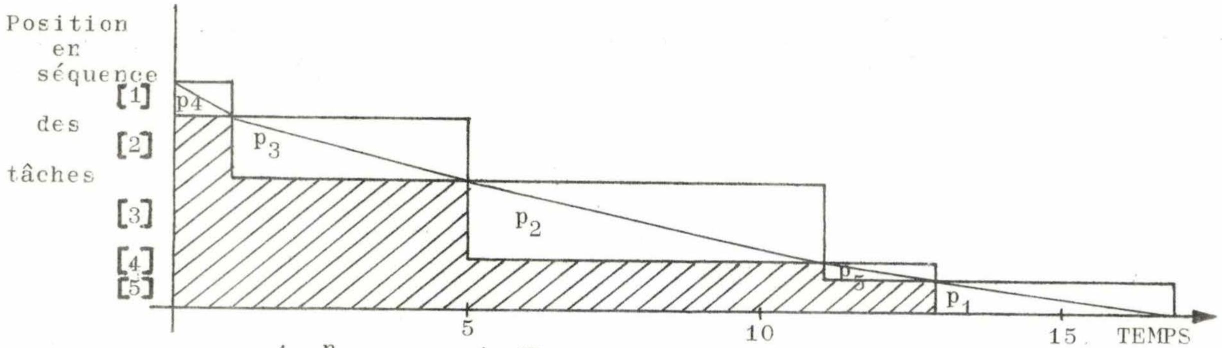
EXEMPLE

tâches	:	1	2	3	4	5
durées de traitement	:	4	6	4	1	2
coefficient d'importance	:	1	2,5	2	1	0,8
rapport entre les durées et les coefficients	:	4	2,4	2	1	2,5

L'application du théorème donne la séquence 4-3-2-5-1.
Le graphique correspondant est le suivant :

---

[1] SMITH W.E. : op. cit.



$$\bar{F}u = \frac{1}{n}\sum_{i=1}^{n} U_i F_i = \frac{1}{5}\left[(1x1) + (2x5) + (2,5x11) + (0,8x13) + (1x17)\right]$$
$$= 13,18$$

Si nous avions interverti les tâches 2 et 5, dont les rapports sont très proches, la durée moyenne pondérée serait de 13,22. Notons également que les vecteurs-tâches forment une courbe convexe.

Remarques :

1. $\bar{L}u$ et $\bar{W}u$ sont minimisés dans les mêmes conditions que $\bar{F}u$ puisqu'il s'agit de mesures régulières.
2. Il existe une méthode fort utilisée en pratique et qui consiste à ordonnancer les opérations dans un ordre de coefficients d'importance non décroissant ; en d'autres mots on traite les opérations en fonction de leur importance :

$$[1] \geqslant u_{[2]} \geqslant \dots \geqslant u_{[n]}$$

Cette politique sera optimale dans deux cas :

- si toutes les durées de traitement sont égales ; supposons que le coefficients d'importance de 4 opérations soient respectivement de 2,5,1,10 et que la durée de traitement de chacune d'entre elles soit de 8. L'application du théorème donnerait la séquence : 4-2-1-3, puisque $8/10 \leqslant 8/5 \leqslant 8/2 \leqslant 8/1$ tandis que l'ordre des coefficients d'importance aboutirait à la même séquence puisque $10 > 5 > 2 > 1$;
- si les coefficients sont directement proportionnels aux durées de traitement c-a-d si $u_i = kp_i$ où k est un coefficient multiplicateur quelconque. Dans ce cas, tous les ordonnancements sont équivalents car le rapport p_i/u_i serait

identique pour toutes les opérations.

Remarquons cependant que cette politique n'est pas à appliquer dans tous les cas. On peut même aboutir au plus mauvais ordonnancement possible en fonction de la minimisation de $\overline{F}u$. Ce serait le cas si $u_i = k\sqrt{p_i}$

EXEMPLE :

opération	:	1	2	3	4	
durée de traitement	:	4	16	9	25	(k=2)
coefficient	:	4	8	6	10	

L'ordonnancement optimal serait (par application du théorème) 1,3,2,4. L'ordre croissant des coefficients aboutirait à la séquence : 4,2,3,1 : cet ordonnancement maximise $\overline{F}u$.

§ 2 : Retard moyen pondéré : $\overline{T}u$

$Tu = \frac{1}{n}\sum_{i=1}^{n} u_i p_i = \frac{1}{n}\sum_{i=1}^{n} u_i \max(0,L_i)$. Si quelques résultats intéressants ont été trouvés, le problème général est jusqu'à présent irrésolu (il s'agit de la première manifestation de la difficulté du problème, lorsqu'on s'intéresse aux retards des opérations) :

- On peut d'abord essayer d'ordonnancer les travaux selon les dates de livraison. Cette règle minimise T_{max} (Titre II, chapitre 2, section 2). Si T_{max} est nul, $\overline{T}u$ l'est aussi et l'ordonnancement est optimal.
- Si cet ordonnancement donne un min T_{max} différent de 0, on ordonnance les opérations en ordre de p/u non décroissants, ce qui minimise $\overline{L}u$. S'il s'en suit que chaque opération est en retard, l'écart et le retard sont équivalents et la séquence est optimale par rapport à $\overline{T}u$.
- Il suffit qu'une tâche soit en retard ou qu'une ne le soit pas pour que les résultats précédents soient inapplicables. Une procédure intéressante mais ne garantissant pas l'obtention d'ordonnancements optimaux peut être trouvée chez Schild et Fredman [1]

Section 4 : L'incorporation dans un problème n/1 des durées de préparation : le Branch and Bound et la programmation dynamique

Ce problème s'apparente à celui du représentant de commerce qui doit visiter n clients habitant dans des villes différentes ; ces villes sont reliées par des routes, auxquelles sont associées soit une distance, soit un coût, qu'il s'agit de minimiser.

L'introduction dans le problème d'ordonnancement d'un état "d'inoccupation" et des durées de préparation en fait un problème particulier : sur base d'une matrice carrée S reprenant les durées de préparation, il faut trouver une matrice X :

[1] SCHILD A. et FREDMAN E.J. : "Scheduling Tumbs with Linear Loss Functions" Management Science vol. 7 n° 3 ; avril 1961.

- où x_{ik} prend la valeur 1 ou 0 selon que la tâche i précède immédiatement la tâche k ou non.
- qui soit soumise aux contraintes suivantes : $\sum_{i=1}^{n} x_{ik}$ et $\sum_{k=1}^{n} x_{ik} = 1$
- telle que $\sum_{i=1}^{n} \sum_{k=1}^{n} x_{ik}s_{ik}$ soit minimisé.

Il s'agit en fait d'un problème de programmation linéaire en nombres entiers où les valeurs des inconnues (x_{ik}) ne peuvent être que 0 ou 1. La programmation n'est pourtant pas applicable, à cause de l'impossibilité d'y introduire la contrainte suivant laquelle il ne peut pas y avoir de boucles avant que les n tâches ne soient toutes traitées. Les matrices suivantes sont donc exclues :

$$A = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \qquad B = \begin{matrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{matrix} \qquad C = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{matrix}$$

$$D = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{matrix} \qquad E = \begin{matrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{matrix}$$

Toutes ces matrices sont à exclure parce qu'elles forment des boucles dont les cas les plus flagrants se présentent dans les matrices A et E. Ces nouvelles restrictions sont tellement contraignantes qu'elles rendent la programmation linéaire pratiquement inapplicable même dans les problèmes de petites dimensions auquel cas on aurait encore plus vite fait d'explorer toutes les possibilités. Notons qu'on pourrait résoudre le problème de cette façon en réitérant la méthode du simplexe chaque fois qu'une solution est à rejeter et en insérant dans la nouvelle itération des contraintes qui exprimeraient que la solution antérieure est impossible. De toute façon la solution est <u>cyclique</u> : grâce à l'incorporation d'un état "vide", la solution se présentera de la façon suivante : état vide - opération i - opération k - ... opération , - retour à l'état vide etc ... Même si l'ordonnancement optimal se présente de la façon suivante : 2-5-4-0-3-1 , elle est acceptable, car vu son caractère cyclique elle correspond en fait à une suite sans fin du type 2-5-4-0-3-1-2-5-4-0-3-1-2-5-4.. L'opération de départ est arbitraire et la séquence 2-5-4-0-3-1 est équivalente à la séquence 4-0-3-1-2-5 ou à 0-3-1-2-5-4. Comme on part de l'état vide, c'est cette dernière représentation qui sera choisie.

Pour minimiser la matrice des durées de préparation, deux méthodes sont possibles : l'algorithme du <u>Branch and Bound</u> et la <u>programmation dynamique</u> ; nous les étudierons de façon approfondie car elles sont très souvent utilisées dans la résolution de problèmes d'ordonnancement. Remarquons auparavant :

- que dans le problème du représentant de commerce, la matrice S est symétrique puisque la distance séparant A de B est identique à celle séparant B de A, ce qui n'est pas nécessairement le cas lorsque la matrice S représente les durées de préparation ;

- que les arêtes sont plus nombreuses dans un problème d'ordonnancement, chaque tâche étant liée à toutes les autres ;

- ces liaisons ne sont pas, contrairement au problème du représentant, soumises au principe du "circuit court", par lequel la distance entre deux villes différentes est inférieure ou égale à la somme des distances reliant chacune de ces villes à une troisième.

§1 : L'algorithme du Branch and Bound [1]

Principe : si, sous le nom de Branch and Bound se cachent différentes méthodes opératoires de calcul, elles reposent toutes sur le même principe : le problème original est décomposé en un ensemble de problèmes plus simples fortement liés entre eux ; chacun de ces problèmes représente une solution plus ou moins complète du problème original ; certaines routines ont pour but d'en résoudre certains, d'en éliminer d'autres et de partitionner ceux qui sont trop complexes.

Développement : chaque problème irrésolu de la liste (comprenant l'ensemble de ces problèmes) est une matrice carrée nxn (n est le nombre d'opération + une) ; chacun d'eux varie selon le nombre d'étapes (k) déjà spécifiées (on appelle "étape" l'établissement d'un lien de précédence immédiate entre deux opérations). Il faut donc choisir pour chaque problème une permutation qui optimiserait les n-k étapes restantes. (ce principe de décomposition en sous problèmes est l'aspect Branching de l'algorithme).

Chacun de ces problèmes se caractérise par une valeur Y, donnant la limite inférieure de la solution finale (aspect Bounding) ; en d'autres termes, il est impossible de trouver une meilleure solution en continuant la spécification des étapes à partir de ce problème partiel. Des limites inférieures triviales consisteraient, par exemple à prendre la somme des valeurs minimales de la matrice S (une par ligne et par colonne) ou la somme des valeurs de la rangée minimale de S.

Comment attaquer les problèmes irrésolus ?

- s'il n'y a plus que deux étapes à spécifier, on passe à la "routine de solution". Si la valeur de cette solution est inférieure à Z, on garde cette solution ; sinon, on la rejette. Z est la valeur de la meilleure solution trouvée antérieurement ; au départ, Z reçoit la valeur - critère d'un ordonnancement quelconque.

- si la limite inférieure d'un problème partiel (Y) est supérieure à Z, on élimine ce problème car il est incapable de contribuer à l'amélioration d'une solution, antérieurement connue : routine d'éli-

[1] LITTLE J.D.C., MURTY K.G., SWEENY D.W. et KAREL C. : "An Algorithm for the Traveling-Salesman Problem" Journal of the OperationsResearch Society of America vol. 11 n° 6, novembre 1966 p. 972-989

mination.

- si aucune de ces conditions n'est satisfaite, on passe à la "routine de partitionnement. Elle subdivise le problème en deux problèmes plus simples :
  - dans le premier, on envisage le cas où une nouvelle étape est spécifiée (détermination d'une liaison ij) ; la limite inférieure de ce sous-problème peut-être supérieure à celle du problème dont il est issu ;
  - dans le second, on exclut la liaison ij dont on a supposé l'existence dans le premier sous-problème ; on ne spécifie donc aucune étape. La limite inférieure de ce sous-problème sera certainement supérieure à l'ancienne.

Expliquons-nous :

Little fait intervenir dans cette routine les notions de "réduction" et de "sélection".

- réduction : c'est le procédé par lequel on obtient au moins UN élément nul dans chaque ligne et dans chaque colonne de la matrice S. Toutes les solutions sont affectées de la même façon en retranchant une constante : en particulier, la solution optimale reste optimale. On aboutit à une matrice S', pour laquelle Y'=A (A est la constante retranchée). Comme les éléments de S' ne peuvent devenir négatifs, la constante à retrancher est au plus égale à la plus petite valeur d'une ligne ou d'une colonne. La somme de ces réductions procure une limite inférieure au problème original (Y).
- sélection : une fois S réduit, l'existence ou l'inexistence d'une liaison entre i et j forme la base de deux nouveaux problèmes :

  1. S_{ij}: par rapport à la matrice dont elle est issue, cette nouvelle matrice subit les tranformations suivantes :
     - on décide d'aller de i à j : liaison obligatoire
     - sont par conséquent interdites, les liaisons :
       - partant de i vers k, où k diffère de j
       - aboutissent à j en partant de k, où k diffère de i ;
     - la liaison s_{ij} est exclue : son choix entrainerait une boucle.

  Ces restrictions aboutissent à l'élimination de certains éléments nuls de la matrice de base (on les élimine en leur affectant une valeur "infinie") ; la nouvelle matrice peut être réduite à son tour et elle accroît la valeur de la limite inférieure, Y.

2. Sn_{ij} : par rapport à la matrice dont elle est issue, cette nouvelle matrice subit les transformations suivantes :

- on exclut la liaison i à j (la même que dans S_{ij}
- toutes les autres liaisons sont possibles.

Une nouvelle réduction est possible ; la nouvelle matrice donne une valeur de Y supérieure à celle donnée par la matrice dont elle est issue. Pourquoi ? Dans le choix de la liaison i-j, l'objectif consiste à rendre la limite inférieure de Sn_{ij} la plus élevée possible, pour pouvoir éliminer Sn_{ij} le plus rapidement possible (au moment où Y dépasse Z). Si nous agissons ainsi, c'est pour ne pas devoir continuer l'exploration de toutes les autres liaisons (dans la matrice Sn_{ij}, aucune étape n'est spécifiée). Il suffit donc de choisir l'arc ij pour lequel la somme des deux constantes de réduction (ligne et colonne) est la plus élevée. Seuls les éléments nuls sont candidats : puisqu'il y a au moins un élément nul par ligne et par colonne de la matrice réduite, choisir un élément non nul reviendrait à rendre certains éléments nuls, négatifs.

Soit le problème d'ordonnancement suivant : supposons un atelier de peinture devant traiter cinq opérations, chacune d'elles nécessitant une couleur différente. Notons tout de suite que le nombre d'opérations ne doit pas nécessairement être égal au nombre de couleurs différentes ; dans le cas où il y a plus d'opérations que de couleurs : on a avantage à grouper en une opération toutes celles qui nécessitent la même couleur, car la durée de préparation entre deux de ces opérations est nulle. Nous supposons en outre que ces 5 opérations (ou groupe d'opérations) nécessitent des durées de traitement respectivement de 8,14,4,15 et 9 ; ces opérations sont numérotées de 1 à 5. La matrice S comporte donc 6 lignes et 6 colonnes, la dernière de celles-ci représentant le temps nécessaire pour que la machine inoccupée au départ soit prête à traiter la première opération de la séquence et le temps nécessaire pour que la machine ayant terminé la dernière opération de la séquence revienne à l'état vide. On aura alors terminé le cycle.

S = (lignes : MACHINES ; colonnes : MACHINES)

	1	2	3	4	5	6
1	-	1	7	3	14	2
2	3	-	6	9	1	24
3	6	14	-	3	7	3
4	2	3	5	-	9	11
5	15	7	11	2	-	4
6	20	5	13	4	18	-

Un tiret équivaut à une impossibilité et est équivalent à :"infini"

Comme <u>limite supérieure</u> Z, prenons la valeur d'un ordonnancement quelconque, par exemple 1-2-3-4-5-6 ; puisque les durées de traitement n'interviennent pas dans la détermination de la valeur minimum de F_{max}, il suffit de s'occuper des éléments de la matrice S. D'où : 1+6+3+9+4+20= 43 ; tout problème donnant une solution (limite inférieure) supérieure à 43 sera rejettée.

<u>Première étape</u>

Elle consiste dans la réduction de la matrice S de telle sorte qu' un élément nul au moins apparaisse dans chaque ligne et dans chaque colonne :

S- ; 16

-	0^2	3	2	13	1
2	-	2	8	0^6	23
3	11	-	0^0	4	0^1
0^2	1	0^2	-	7	9
13	5	6	0^2	-	2
16	1	6	0^1	14	-

Pour obtenir cette matrice, on retranche 1 des lignes 1 et 2, 2 des lignes 4 et 5, 3 de la ligne 3 et 4 de la ligne 6.

Chaque ligne contient, par construction au moins un élément nul mais la colonne 3 n'en contient pas encore ; on peut y retrancher 3 unités. Le total des réductions s'élève à 16, qui représente la limite inférieure Y en dessous de laquelle il sera impossible de trouver une meilleure solution. Notons qu'il ne suffit pas de s'arrêter ici en suivant n'importe quelle chaîne d'éléments nuls pour obtenir une matrice admissible car des boucles se forment.

Exemple :

1-2-5-4-1 : 3 et 6 n'ont pas été traitées
3-4 : 6 n'a pas été traité
6-4 : les 6 tâches sont traitées mais la cycle devrait se terminer par 1 et non pas par 4.

Les indices suscrits aux éléments nuls de la matrice [S- ; 16] (étape spécifiée et la valeur d'Y) servent à passer à l'étape suivante qui consiste à partitionner cette matrice, (puisque les conditions nécessaires pour passer aux routines de solution et d'élimination ne sont pas remplies); ces indices indiquent les réductions qui seraient possibles si cet élément nul n'existait pas. (l'indice est égal à la somme du plus petit élément de la ligne et de la colonne correspondante).

On va donc partitionner [S-; 16] en deux matrices contenant (ou non) la spécification d'une étape. Comment la choisir ? Nous avons dit que le principe de construction des matrices $[S_{ij};Y']$ et $[Sn_{ij};Y'']$ n'était pas le même. La première de celles-ci est avantagée du fait que le nombre d'étapes spécifiées y est plus élevé (la seconde, par définition, a au moins une étape de spécifiée en moins). Pour réduire le nombre d'itérations on essaie d'éliminer le plus rapidement possible la solution $[Sn_{ij};Y'']$ en choisissant dans la matrice à partitionner l'élément nul dont l'indice suscrit est le plus élevé : Y'' sera donc certainement plus élevé qu'Y. Dans l'exemple il s'agit de l'élément 2-5, dont l'indice est de 6 ; il va servir de base au partitionnement. La matrice $[Sn_{25};22]$ exprime la situation où la liaison 2-5 est exclue (l'élément correspondant reçoit la valeur "infini"); l'élément nul ayant disparu, on peut réduire de 6 unités, donnant à Y'' la valeur 16+6=22. La matrice $[S_{25};16]$ exprime la situation où la liaison 2-5 est obligatiore ; l'élément nul subsiste mais on exclut toutes les liaisons partant de 2 vers un autre élément que 5 et tous les éléments qui aboutissent à 5 en partant d'un élément différent de 2; pour éviter les boucles, on élimine également la liaison 5-2. Quant à la valeur de Y', il suffit de voir si parmi les liaisons exclues, il y en a qui étaient nulles, ce qui permettrait une réduction supplémentaire. Dans notre cas, il n'y en a pas et Y' reste égal à Y.

Deuxième étape (Z=43)

Problèmes irrésolus :

$S_{25};16$

$Sn_{25};22$

$[S_{25};16]$

$$\begin{matrix} - & 0^2 & 3 & 2 & \ominus & 1 \\ \ominus & \ominus & \ominus & \ominus & x & \ominus \\ 3 & 11 & - & 0^0 & - & 0^1 \\ 0^3 & 1 & 0^3 & - & - & 9 \\ 13 & \ominus & 6 & 0^2 & - & 2 \\ 16 & 1 & 6 & 0^1 & - & - \end{matrix}$$

x: étape obligatoire

$\ominus$: liaison exclue

$[Sn_{25};22]$

$$\begin{matrix} - & 0 & 3 & 2 & 9 & 1 \\ 0 & - & 0 & 6 & - & 21 \\ 3 & 11 & - & 0 & 0 & 0 \\ 0 & 1 & 0 & - & 3 & 9 \\ 13 & 5 & 6 & 0 & - & 2 \\ 16 & 1 & 6 & 0 & 10 & - \end{matrix}$$

Comme tous les problèmes de la liste ont des limites inférieures à Z et plus de deux étapes à spécifier, ils ne peuvent être solutionnés ni éliminés. Comme $[S_{25};16]$ a plus d'étapes spécifiées que $[S_{n25};22]$ et une limite inférieure moins élevée, on partitionnera le premier problème sur base de l'élément 4-1 (on aurait pû prendre 4-3, puisque son indice suscrit est aussi égal à 3).

La nouvelle valeur d'Y pour $[S_{25}n_{41}]$ est égale à l'ancienne plus l'indice suscrit de l'élément 4-1 c-a-d. 16+3=19; la valeur d'Y pour $[S_{25,41}]$ est l'ancienne plus la valeur de l'indice suscrit de l'élément nul qui a disparu par l'interdiction de certaines liaisons (ici il d'agit de l'élément 4-3 dont l'indice est 3) : 16+3=19.

Troisième étape (Z=43)

Problèmes irrésolus :

$Sn_{25};22$

$S_{25,41};19$

$S_{25}n_{41};19$

$[S_{25,41};19]$

⊖	0^1	0^3	⊖	-	1
⊖	-	-	-	x	-
⊖	11	-	0^0	-	0^1
x	⊖	⊖	⊖	⊖	⊖
⊖	-	3	0^2	-	2
⊖	1	3	0^1	-	-

$[S_{25}n_{41};19]$

-	0^2	3	2	-	1
⊖	-	-	-	x	-
0^{10}	11	-	0^0	-	0^1
-	1	0^4	-	-	9
10	-	6	0^2	-	2
13	1	6	0^1	-	-

Inférieures à 43, toutes les solutions restent possibles ; nous continuons l'esploration à partir de $[S_{25,41};19]$ dont le nombre d'étapes spécifiées est plus élevé que celui de la matrice $[S_{25}n_{41};19]$.

On passe à la quatrième étape en partitionnant $[S_{25,41};19]$ sur base de l'élément 1-3 ; la valeur Y de $[S_{25,41}n_{13}]$ s'élève à 22 tandis que celle de $[S_{25,41,13}]$ est portée à 20 par l'interdiction de l'élément 1-2.

Quatrième étape (Z=43)

Problèmes irrésolus :

$S\bar{n}_{25};22$

$S_{25}\bar{n}_{41};19$

$S_{25,41,13};20$

$S_{25,41}\bar{n}_{13};22$

$$\left[S_{25,41,13};20\right]$$

$$\begin{array}{cccccc}
- & - & x & - & - & - \\
- & - & - & - & x & - \\
- & 10 & - & - & - & 0^{12} \\
x & - & - & - & - & - \\
- & - & - & 0^{2} & - & 2 \\
- & 0^{10} & - & 0^{0} & - & -
\end{array}$$

$$\left[S_{25,41}\bar{n}_{13};22\right]$$

$$\begin{array}{cccccc}
- & 0 & - & - & - & 1 \\
- & - & - & - & x & - \\
- & 11 & - & 0 & - & 0 \\
x & - & - & - & - & - \\
- & - & 0 & 0 & - & 2 \\
- & 1 & 0 & 0 & - & -
\end{array}$$

Remarquons que dans la première matrice, l'élément 3-4 doit être exclu, car le garder aboutit à la création d'une boucle, la séquence 4-1-6-3 étant déjà spécifiée; admettre 3-4 donnerait 4-1-6-3-4 ce qui est inadmissible puisque toutes les opérations ne sont pas encore traitées.

Pour les raisons données dans les étapes précédentes, il faut continuer le partitionnement de $\left[S_{25,41,13};20\right]$ et ce sur base de l'élément 3-6. Les deux nouvelles valeurs d'Y sont de 20(20+0) et 32(20+12).

Cinquème étape (Z=43)

Problèmes irrésolus :

$S\bar{n}_{25};22$

$S_{25}\bar{n}_{41};19$

$S_{25,41}\bar{n}_{13};22$

$S_{25,41,13,36};20$

$S_{25,41,13}\bar{n}_{36};32$

$$\left[S_{25,41,13,36};20\right]$$

$$\begin{array}{cccccc}
- & - & x & - & - & - \\
- & - & - & - & x & - \\
- & - & - & - & - & x \\
x & - & - & - & - & - \\
- & - & - & 0^{0} & - & - \\
- & 0^{0} & - & 0^{0} & - & -
\end{array}$$

$$\left[S_{25,41,13}{}^{n}{}_{36};32\right]$$

$$\begin{array}{cccccc}
- & - & x & - & - & - \\
- & - & - & - & x & - \\
- & 0 & - & - & - & \ominus \\
x & - & - & - & - & - \\
- & - & - & 0 & - & 0 \\
- & 0 & - & 0 & - & -
\end{array}$$

Puisqu'il ne reste que deux étapes à spécifier, on passe à la routine de solution pour $\left[S_{25,41,13,36}\right]$. Nous avons deux séquences : 2-5 et 4-1-3-6 ; il suffit de voir si la liaison 5-4 est préférable à la liaison 6-2. Dans la matrice originale, ces éléments ont les valeurs 2 et 5 ; on serait donc tenté de choisir la liaison 5-4 pour aboutir à la séquence 2-5-4-1-3-6.

En fait les deux solutions sont identiques, parce que l'ordonnancement est cyclique : les liaisons 5-4 et 6-2 apparaitront dans les deux ordonnancements : 2-5-4-1-3-6 est équivalent à 4-1-3-6-2-5 et à 6-2-5-4-1-3 ; on choisira cette dernière présentation car elle commence par l'état "vide". La valeur de ces ordonnancements est de 20 (il ne faut pas oublier d'insérer la liaison finale 3-6 qui termine la cycle). On peut donc, dans l'ensemble des problèmes restés inachevés éliminer ceux dont la limite inférieure Y est supérieure à la nouvelle valeur de Z (donnant la valeur d'un ordonnancement); il ne reste qu'à explorer la matrice $\left[S_{25}{}^{n}{}_{41}\right]$ puisque Y y est égal à 19. Remarquons qu'on peut éliminer tous les problèmes dont la limite inférieure est égale à 20; dans le cas où on voudrait trouver tous les ordonnancements optimaux, il faudrait explorer les problèmes dont les limites égalent 20, ce qui peut considérablement accroître le nombre d'itérations.

La matrice $\left[S_{25}{}^{n}{}_{41}\right]$ est donc partitionnée sur base de l'élément 3-1. Les deux nouveaux problèmes sont : $\left[S_{25}{}^{n}{}_{41}{}^{n}{}_{31};29\right]$ et $\left[S_{25,31}{}^{n}{}_{41};20\right]$; le premier sous-problème peut être éliminé d'office, Y étant supérieur à Z; en théorie le second pourrait l'être auusi si on ne cherche qu'un ordonnancement optimal ; si on veut trouver tous les ordonnancements optimaux il faudrait continuer.

Etape 7 (Z=20)

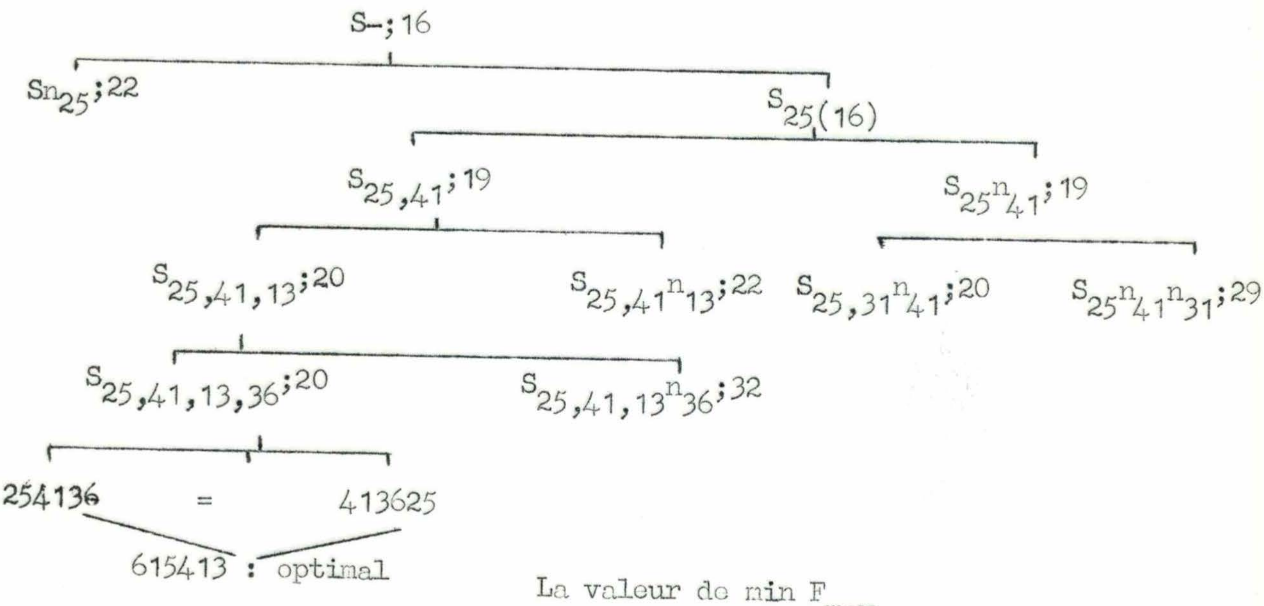
Problème irrésolu

$S_{25,31}{}^{n}{}_{41};20$

$$\begin{array}{cccccc}
\ominus & 0^{1} & \ominus & 2 & - & 0^{1} \\
\ominus & - & - & - & x & - \\
x & \ominus & \ominus & \ominus & \ominus & \ominus \\
\ominus & 1 & 0^{7} & - & - & 8 \\
\ominus & - & 6 & 0^{1} & - & 1 \\
\ominus & 1 & 6 & 0^{1} & - & -
\end{array}$$

Le partitionnement de cette nouvelle matrice n'aboutit pas encore à l'élimination des deux problèmes qui en sont issus et pour trouver tous les ordonnancements optimaux il faudrait continuer ; en fait la solution que nous avons antérieurement trouvée est unique (voir infra : programmation dynamique).

Le problème initial aurait nécessité l'exploration de 720 solutions (6!); nous en avons examiné 12 qu'il s'agissait de bien choisir et dont certaines sont restées inachevées. L'arbre des solutions envisagées se présente de la façon suivante :

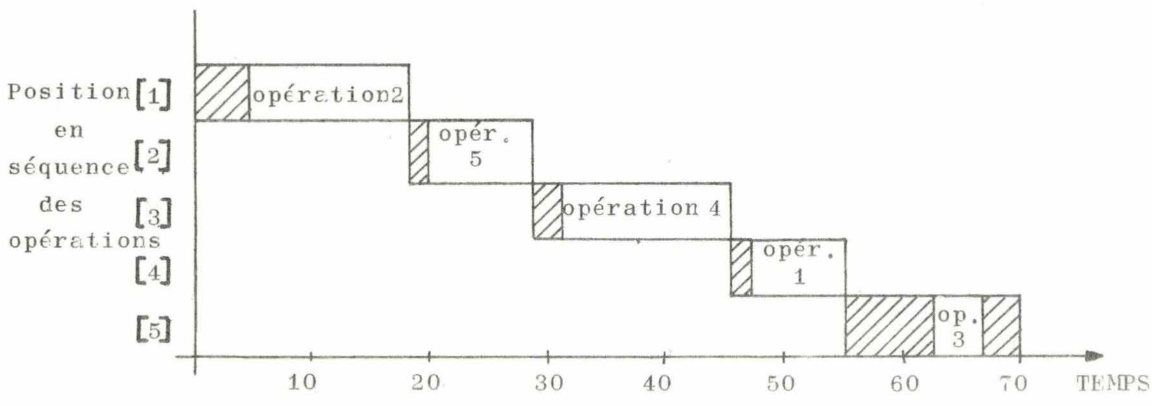


La valeur de min F_{max}

$$= \sum_{i=1}^{n} s_{[i-1],[i]} + \sum_{i=1}^{n} p_{[i]}$$
$$= 20 + 50 = 70$$

Graphique :

La partie hachurée est la durée de préparation nécessaire entre le traitement des opérations placées en (i - 1)ième en i ième position.



§2 : La programmation dynamique [1]

Il existe une procédure plus générale que celle du Branch and Bound pour résoudre ce genre de problème ; plus générale car elle permet d'aborder une plus grande diversité de problèmes d'ordonnancement ; elle a pourtant des limites que nous avons évoquées. (Titre II, chapitre 2)

Principe :

Un problème d'optimisation où le temps n'apparait ni explicitement ni implicitement se présente comme un tout. On le découpe en plusieurs parties. On détermine la ou les solutions optimales (dites sous-politiques optimales) pour une partie, pour deux parties et ainsi de suite, de proche en proche, pour arriver à la solution optimale ou politique optimale. Ces déterminations successives de sous-politiques optimales pour aboutir à la politique optimale, constitue un processus de résolution dynamique en soi. La méthode de programmation qui s'inspire fondamentalement de ce processus mérite pour cette raison le nom de programmation dynamique.

Procédure opératoire : l'ensemble des villes (opérations) est divisé en quatre sous-ensembles :

Co : l'opération de départ

Ci : une opération quelconque (différente de Co)

$\{Ck\}$: k opérations qui ne soient ni Co, ni Ci

$\{C(n-k-2)\}$:les opérations restantes.

Supposons que nous connaissions l'opération de départ et d'arrivée (nous nous occupons de la minimisation de S et donc du cycle des opérations) : notons cette opération Co. On explore toutes les liaisons existant entre cette opération et les autres. Cette première étape aboutit à une sous-politique optimale.

La seconde étape consiste à reprendre les résultats de l'étape précédente en leur ajoutant une liaison avec une opération restante (pour éviter les boucles). La combinaison minimum donne la sous-politique optimale de la seconde étape. On continue de la même façon tant que toutes les opérations ne sont pas liées les unes aux autres.

Si on définit $f(Ci;\{Ck\})$ comme la longueur du plus court chemin allant de Ci à Co en passant par Ck, il est clair que si k=o, le vecteur $\{Ck\}$ est nul ; $f(Ci;\{0\}) = s_{io}$ où s_{io} exprime la longueur du chemin reliant directement i à l'état de départ. On aura alors spécifié la dernière étape. Si de plus, on connaît $f(Co;\{C_{n-1}\})$ on aura déterminé le chemin optimal.

[1] BELLMAN R. : "Dynamic Programming Treatment of the Travelling Salesman Problem", J. Assn. for Computing Machinery vol. 9 n° 1, janvier 1962
HELD H. et KARP R.M. : "A Dynamic Programming Approach to Sequencing Problems" ; J. Soc. Ind. and Appl. Math. vol. 10 n° 2, mars 1962

La procédure consiste donc à accroître la valeur de k d'une unité (en commençant par 0) chaque fois qu'une nouvelle étape est franchie. Le principe d'optimalité s'exprime alors :

$$f(Ci; \{Ck\}) = \min_{Cj \in \{Ck\}} \left[s_{ij} + f(Cj; \{Ck\} - \{Cj\}) \right]$$

Cela signifie que pour trouver la valeur du chemin optimal, partant de Ci et aboutissant en Co, en passant (dans un certain ordre) par Ck , il faut explorer k alternatives. La valeur d'une de ces alternatives se compose de deux éléments :

- la valeur du chemin qui, partant de Cj aboutit en Co en passant par les opérations contenues dans Ck excepté Cj : $f(Cj; \{Ck\} - \{Cj\})$
- la valeur du chemin reliant Ci à Cj ;

On opère de la même façon pour toutes les opérations que Ci peut représenter.

Reprenons notre exemple :

$$S = \begin{matrix} - & 1 & 7 & 3 & 14 & 2 \\ 3 & 6 & 6 & 9 & 1 & 24 \\ 6 & 14 & - & 3 & 7 & 3 \\ 2 & 3 & 5 & - & 9 & 11 \\ 15 & 7 & 11 & 2 & - & 4 \\ 20 & 5 & 13 & 4 & 18 & - \end{matrix}$$

A l'étape 0, correspond k=0 puisqu'aucune opération n'a encore été spécifiée : Ck est un vecteur vide. Il y a 5 problèmes de ce type puisque l' opération de départ et d'arrivée est connu (nous prenons 6 par facilité mais nous aurions pu prendre comme Co n'importe quelle autre opération, le problème étant cyclique).

$f(C1; \{\}) = s16 = 2$ $\qquad$ $f(C4; \{\}) = s46 = 11$
$f(C2; \{\}) = s26 = 24$ $\qquad$ $f(C5; \{\}) = s56 = 4$
$f(C3; \{\}) = s36 = 3$

La sous politique optimale : chemin 1 - 6

A l'étape 1, on reprend les résultats antérieurs et on ajoute les nouvelles liaisons existant entre Ci et Cj ; il y a 20 problèmes du type k=1, une étape étant déjà spécifiée :

$f(C1; \{C2\}) = s12 + f(C2; \{\})$
$f(C1; \{C3\}) = s13 + f(C3; \{\})$
$\vdots$
$f(C5; \{C4\}) = s54 + f(C4; \{\})$

Les diverses équations donnent la valeur des chemins reliant Ci à Co en passant par Cj ; cette valeur est égale à la somme du chemin reliant Cj à Co dont les valeurs ont été déterminées dans l'étape précédente et du chemin séparant Ci à Cj, dont les valeurs correspondent aux éléments s_{ij} de la matrice S.

f(C1;{C2}) = 1 +24 = 25
f(C1;{C3}) = 7 + 3 = 10
f(C1;{C4}) = 3 +11 = 14
f(C1;{C5}) =14 + 4 = 18
f(C2;{C1}) = 3 + 2 = 5
f(C2;{C3}) = 6 + 3 = 9
f(C2;{C4}) = 9 +11 = 20
f(C2;{C5}) = 1 + 4 = 5
f(C3;{C1}) = 6 + 2 = 8
f(C3;{C2}) =14 +24 = 38
f(C3;{C4}) = 3 + 11 = 14
f(C3;{C5}) = 7 + 4 = 11
f(C4;{C1}) = 2 + 2 = 4
f(C4;{C2}) = 3 + 24 = 27
f(C4;{C3}) = 5 + 3 = 8
f(C4;{C5}) = 9 + 4 = 13
f(C5;{C1}) =15 + 2 = 17
f(C5;{C2}) = 7 + 24 = 31
f(C5;{C3}) =11 + 3 = 14
f(C5;{C4}) = 2 + 11 = 13

La sous politique optimale est à la fin de l'étape 1 le chemin : 4-1-6

A l'étape 2, on fait la même chose en spécifiant une étape de plus : k=2. Les éléments du vecteur {Ck} donnent les opérations de "passage". Par application du principe d'optimalité, les 30 problèmes se posent de la façon suivante :

f(C1;{C2;C3}) = min {s12 + f(C2;{C3}), s13 + f(C3;{C2}) }
f(C ;{C2,C4}) = min {s12 + f(C2;{C4}), s14 + f(C4;{C2}) }
:
:
f(C5;{C3,C4}) = min {s53 + f(C3;{C4}), s54 + f(C4;{C3}) }

Dans notre exemple, les valeurs spécifiques sont :

f(C1;{C2;C3}) = min (1 + 9 , 7 + 38) = 10
f(C1;{C2;C4}) = min (1 +20 , 3 + 27) = 21
f(C1;{C2;C5}) = min (1 + 5 ,14 + 31) = 6
f(C1;{C3;C4}) = min (7 +14 , 3 + 8) = 11
f(C1;{C3;C5}) = min (7 +11 ,14 + 14) = 18
f(C1;{C4;C5}) = min (3 +13 ,14 + 13) = 16
f(C2;{C1;C3}) = min (3 +10 , 6 + 8) = 13
f(C2;{C1;C4}) = min (3 +14 , 9 + 4) = 13
f(C2;{C1;C5}) = min (3 +18 , 1 + 17) = 18
f(C2;{C3;C4}) = min (6 +14 , 9 + 8) = 17
f(C2;{C3;C5}) = min (6 +11 , 1 + 14) = 15
f(C2;{C4;C5}) = min (9 +13 , 1 + 13) = 14
f(C3;{C1;C2}) = min (6 +25 ,14 + 5) = 19
f(C3;{C1;C4}) = min (6 +14 , 3 + 4) = 7

$$
\begin{aligned}
f(C3;\{C1;C5\}) &= \min(6+18,\ 7+17) = 24\\
f(C3;\{C2;C4\}) &= \min(14+20,\ 3+27) = 30\\
f(C3;\{C2;C5\}) &= \min(14+5,\ 7+31) = 19\\
f(C3;\{C4;C5\}) &= \min(3+13,\ 7+13) = 16\\
f(C4;\{C1;C2\}) &= \min(2+25,\ 3+5) = 8\\
f(C4;\{C1;C3\}) &= \min(2+10,\ 5+8) = 12\\
f(C4;\{C1;C5\}) &= \min(2+18,\ 9+17) = 20\\
f(C4;\{C2;C3\}) &= \min(3+9,\ 5+38) = 12\\
f(C4;\{C2;C5\}) &= \min(3+5,\ 9+31) = 8\\
f(C4;\{C3;C5\}) &= \min(5+11,\ 9+14) = 16\\
f(C5;\{C1;C2\}) &= \min(15+25,\ 7+5) = 12\\
f(C5;\{C1;C3\}) &= \min(15+10,\ 11+8) = 19\\
f(C5;\{C1;C4\}) &= \min(15+14,\ 2+4) = 6\\
f(C5;\{C2;C3\}) &= \min(7+9,\ 11+38) = 16\\
f(C5;\{C2;C4\}) &= \min(7+20,\ 2+27) = 27\\
f(C5;\{C3;C4\}) &= \min(11+14,\ 2+8) = 10
\end{aligned}
$$

Les sous-politiques optimales sont : $\begin{cases} 5-4-1-6 \\ 1-2-5-6 \end{cases}$

<u>A l'étape 3</u>, k = 3 ; les 20 problèmes sont alors de la forme :

$$
\begin{aligned}
f(C1;\{C2;C3;C4\}) = \min\big(&s12 + f(C2;\{C3;C4\}),\ s13 + f(C3;\{C2;C4\}),\\
&s14 + f(C4;\{C2;C3\})\big)\\
&\vdots\\
f(C5;\{C2;C3;C4\}) = \min\big(&s52 + f(C2;\{C3;C4\}),\ s53 + f(C3;\{C2;C4\}),\\
&s54 + f(C4;\{C2,C3\})\big)
\end{aligned}
$$

Nous avons alors les résultats suivants :

$$
\begin{aligned}
f(C1;\{C2;C3;C4\}) &= \min(1+17,\ 7+30,\ 3+12) = 15\\
f(C1;\{C2;C3;C5\}) &= \min(1+15,\ 7+19,\ 14+16) = 16\\
f(C1;\{C2;C4;C5\}) &= \min(1+14,\ 3+8,\ 14+27) = 11\\
f(C1;\{C3;C4;C5\}) &= \min(7+16,\ 3+16,\ 14+10) = 19\\
f(C2;\{C1;C3;C4\}) &= \min(3+11,\ 6+7,\ 9+12) = 13\\
f(C2;\{C1;C3;C5\}) &= \min(3+18,\ 6+24,\ 1+19) = 20\\
f(C2;\{C1;C4;C5\}) &= \min(3+16,\ 9+20,\ 1+6) = 7\\
f(C2;\{C3;C4;C5\}) &= \min(6+16,\ 9+16,\ 1+10) = 11\\
f(C3;\{C1;C2;C5\}) &= \min(6+21,\ 14+13,\ 3+8) = 11\\
f(C3;\{C1;C4;C5\}) &= \min(6+6,\ 14+18,\ 7+12) = 12\\
f(C3;\{C1;C4;C5\}) &= \min(6+16,\ 3+20,\ 7+6) = 13\\
f(C3;\{C2;C4;C5\}) &= \min(14+14,\ 3+8,\ 7+27) = 11\\
f(C4;\{C1;C2;C3\}) &= \min(2+10,\ 3+13,\ 5+19) = 12\\
f(C4;\{C1;C2;C5\}) &= \min(2+6,\ 3+18,\ 9+12) = 8\\
f(C4;\{C1;C3;C5\}) &= \min(2+18,\ 5+24,\ 9+19) = 20\\
f(C4;\{C2;C3;C5\}) &= \min(3+15,\ 5+19,\ 9+16) = 18\\
f(C5;\{C1;C2;C3\}) &= \min(15+10,\ 7+13,\ 11+19) = 30
\end{aligned}
$$

f(C5;{C1;C2;C4}) = min (15 + 21 , 7 + 13 , 2 + 8) = 10
f(C5;{C1;C3;C4}) = min (15 + 11 , 11 + 7 , 2 + 12) = 14
f(C5;{C2;C3;C4}) = min (7 + 17 , 11 + 30 , 2 + 12) = 14

La sous-politique optimale est : 2-5-4-1-6

A l'étape 4, k = 4 ; les problèmes de cette étape ont la forme :

f(C1;{C2;C3;C4;C5} = min [(s12 + f(C2;{C3,C4,C5},s13 + f(C3;C2;C4;C5},
s14 + f(C4 ;{ C2,C3,C5}, s15 + f(C5;{C2,C3,C4})
f(C1;{C2;C3;C4;C5} = min (1 + 11, 7 + 11 , 3 + 18 , 14 + 14) = 12
f(C2;{C1;C3;C4;C5} = min (3 + 19, 6 + 13 , 9 + 20 , 1 + 14) = 15
f(C3;{C1;C2;C4;C5} = min (6 + 11, 14 + 7 , 3 + 8 , 7 + 10) = 11
f(C4;{C1;C2;C3;C5} = min (2 + 16, 3 + 20 , 5 + 12 , 9 + 20) = 17
f(C5;{C1;C2;C3;C4} = min (15 + 15, 7 + 13 ,11 + 11 , 2 + 12) = 14

La sous-politique optimale est : 3-4-1-2-5-6

A l'étape 5, k = 5 ; jusqu'à présent on aboutit toujours à l'étape 6 ; encore ~~faut-il~~ que l'étape de départ soit 6. Ce problème a la forme :

f(C6;{C1;C2;C3;C4;C5}) = min s61 + f(C1;{C2;C3;C4;C5}, s62 + f(C2;{C1;C3;
C4;C5}, s63 + f(C3;{C1;C2;C4;C5}, s64 + f(C4;
{C1;C2;C3;C5}, s65 + f(C5;{C1;C2;C3;C4)
= min (20 + 12, 5 + 15, 13 + 11, 4 + 17, 18 + 14)
= 20

La solution optimale est donc de 6-2-5-4-1-3. Pour retrouver cette séquence, il suffit de remonter de la dernière étape vers la première. La dernière spécifie que 6 précède les opérations 1,2,3,4,5 et termine le cycle. La valeur minimale a été trouvée pour f(C2;{C1,C3,C4,C5} ce qui signifie que 2 précède les opérations 1,3,4,5. Si on remonte à la quatrième étape, on cherche l'équation qui relie ces 4 opérations. La valeur minimale y a été trouvée en mettant 5 avant les opérations 1,3,4; on remonte à la troisième étape pour trouver l'équation reliant ces 3 opérations et ainsi de suite jusqu'à la première étape. La séquence est donc complètement déterminée : 6-2-5-4-1-3-6.

§ 3 L'algorithme du "closest unvisited city" [1]

Pour être complet, il faut citer un autre algorithme connu sous le nom anglo-saxon de "closest unvisited city". Le principe en est extrêmement simple : il suffit de visiter la ville (traiter l'opération) la plus proche (ayant la durée de préparation la plus courte) de l'endroit où on se trouve (par rapport à l'opération qui vient d'être terminée). Cette procédure qui est à rapprocher de la "durée de traitement la plus courte (SPT)" ne donne pas de résultats optimaux ; c'est pour cette raison que nous la mettons à

---

[1] CONWAY R.W., MAXWELL W.L. et MILLER L.W. : op. cit. p. 66

part ; bien qu'elle puisse aboutir à des résultats désastreux, elle peut servir de première approche lorsqu'on cherche un ordonnancement optimal. Il est par exemple intéressant de l'utiliser avant d'appliquer le Branch and Bound car elle permet de donner une valeur <u>originale</u> à Z, moins élevée que si on avait pris un ordonnancement quelconque ; une valeur moins élevée de Z permet de diminuer appréciablement le nombre de problèmes et d'itérations en appliquant plus rapidement la routine d'élimination. Si nous appliquons cet algorithme à notre exemple en permettant à chacune des n+1 opérations d'être l'opération de départ, nous aurons :

$$S = \begin{matrix} - & 1 & 7 & 3 & 14 & 2 \\ 3 & - & 6 & 9 & 1 & 24 \\ 6 & 14 & - & 3 & 7 & 3 \\ 2 & 3 & 5 & - & 9 & 11 \\ 15 & 7 & 11 & 2 & - & 4 \\ 20 & 5 & 13 & 4 & 18 & - \end{matrix}$$

Origine	Séquence	Valeur de la séquence
1	1-2-5-4-3-6-1	1+1+2+5+3+20 = 32
2	2-5-4-1-6-3-2	1+2+2+2+13+14 = 34
3	3-4-1-2-5-6-3	3+2+1+1+4+13 = 24
	3-6-4-1-2-5-3	3+4+2+1+1+11 = 22
4	4-1-2-5-6-3-4	2+1+1+4+13+3 = 24
5	5-4-1-2-3-6-5	2+2+1+6+3+18 = 32
6	6-4-1-2-5-3-6	4+2+1+1+11+3 = 22

Au lieu de commencer, dans le Branch and Bound avec un ordonnancement quelconque 1-2-3-4-5-6-1 dont la valeur de Z=43, on aurait pu partir avec les ordonnancemants 3-6-4-1-2-5 et 6-4-1-2-5-3 dont les valeurs (Z=22) sont très proches de celle de l'ordonnancement optimal (Z=20).

Gavett [1] a testé cet algorithme en supposant que les éléments de la matrice S sont des variables aléatoires identiquement distribuées (distribution normale ou rectangulaire). Il l'a comparé avec les résultats donnés par l'algorithme du Branch and Bound, qui sont optimaux ; dans un cas, il a choisi au hasard l'opération d'origine et dans le second il a passé toutes les origines possibles en revue. Il en a conclu que :

- lorsque le nombre d'opérations augmentait, la différence entre la moyenne de la longueur du chemin optimal et l'espérance mathématique de la longueur du chemin lorsque l'origine est arbitrairement fixée, augmentait mais elle diminuait lorsque la variance des éléments de la matrice S diminuait.
- la même chose se vérifiait lorsqu'on explore toutes les origines possibles.

Quelques résultats lorsque les éléments de S sont distribués normalement avec une moyenne de 1 :

---

[1] GAVETT Y.W. : "Three Heuristic Rules for Sequencing Jobs to a Single Production Facility" ; Management Science vol. 11 n° 8 ; juin 1965.

Nombre d'opérations	écart-type	nombre de problèmes	moyenne du chemin optimal	Espérance mathématique du chemin le plus court origine fixée		toute origine	
5	0,4	100	3,28	3,64	(1,11)	3,44	(1,05)
5	0,1	10	4,49	4,59	(1,02)	4,57	(1,02)
10	0,4	10	4,78	5,51	(1,15)	5,26	(1,10)
10	0,1	10	8,69	9,00	(1,04)	8,87	(1,02)
15	0,4	10	6,40	7,81	(1,22)	7,23	(1,13)
15	0,1	10	12,83	13,17	(1,03)	13,00	(1,01)
20	0,4	10	7,68	9,65	(1,26)	9,07	(1,18)
20	0,1	10	16,51	16,96	(1,03)	16,84	(1,02)

Les valeurs entre parenthèses expriment les rapports entre le solution donnée par l'algorithme du "closest unvisited city" et celle donnée par le Branch and Bound.

Section 5 : Le problème n/2/F/...

§ 1 : Le problème $n/2/F/F_{max}$: algorithme de Johnson et de Mitten

C'est à Johnson [1] que nous devons la résolution de ce problème ; pour des raisons de facilité, nous utiliserons sa notation :

Ai = durée de traitement de la première opération (indiquée par A) de la tâche i ;

Bi = durée de traitement de la seconde opération (indiquée par B) de la tâche i ;

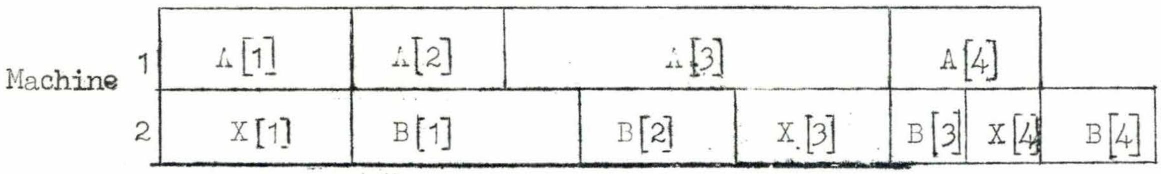
Puisque nous sommes dans un atelier uniforme, les indices A et B indiquent respectivement la première et la seconde machine.
Chaque tâche se compose donc d'un couple (Ai,Bi) ; il n'est pas nécessaire que toutes les tâches aient deux opérations, auquel cas Ai ou Bi sera nul.

Principe de l'algorithme de Johnson :

Le problème consiste en fait à réduire les durées d'attente qui peuvent se produire sur la seconde machine et qui résulte de ce que la seconde opération d'une tâche ne peut commencer que lorsque la première opération de cette tâche est terminée ; si cette durée d'attente est notée X[i], on trouve :

$$X[1] = A[1]$$
$$X[2] = \max(A[1] + A[2] - B[1] - X[1] \;;\; 0)$$
$$X[3] = \max(A[1] + A[2] + A[3] - B[1] - B[2] - X[1] - X[2] \;;\; 0)$$
$$\vdots$$
$$X[j] = \max\left(\sum_{i=1}^{j} A[i] - \sum_{i=1}^{j-1} B[i] - \sum_{i=1}^{j-1} X[i];\; 0\right)$$

[1] JOHNSON S.M. :
chap. 2 de MUTH et THOMPSON op. cit.



Si nous sommons les X[i] :

$$X[1] = A[1]$$
$$X[1] + X[2] = \max\left(A[1] + A[2] - B[1]\ ;\ A[1]\right)$$
$$X[1] + X[2] + X[3] = \max\left(\sum_{i=1}^{3} A[i] - \sum_{i=1}^{2} B[i]\ ;\ \sum_{i=1}^{2} A[i] - \sum_{i=1}^{1} B[i] ;\ A[1]\right)$$

$$\sum_{i=1}^{j} X[i] = \max\left(\sum_{i=1}^{j} A[i] - \sum_{i=1}^{j-1} B[i]\ ;\ \sum_{i=1}^{j-1} A[i] - \sum_{i=1}^{j-2} B[i]\ ;\ \ldots\ldots \sum_{i=1}^{2} A[i] - B[1]\ ;\ A[1]\right)$$

Si on définit $Yj = \sum_{i=1}^{j} A[i] - \sum_{i=1}^{j-1} B[i]$, on a : $\sum_{i=1}^{j} X[i] = \max(Y1, Y2, \ldots Yj)$

$$F\max = \sum_{i=1}^{n} B[i] + \sum_{i=1}^{n} X[i] = \sum_{i=1}^{n} B[i] + \max(Y1, Y2 \ldots Yn)$$

$\sum_{i=1}^{n} B[i]$ étant constant, il faut, pour minimiser Fmax, minimiser le maximum des n valeurs de Yj ; il faut donc trouver un ordonnancement S* tel que Fmax $(S^*) \leqslant$ Fmax S.

Le théorème s'énonce alors de la manière suivante :
Dans un problème n/2/F/Fmax où toutes les tâches sont disponibles, j précède (j+1) si min (Aj,Bj+1) est inférieur au minimum de (Aj+1, Bj).
Johnson a démontré que :

- l'ordonnancement basé sur cette inégalité minimise le maximum des Yi.
- cette inégalité est transitive de telle sorte qu'un ordonnancement complet est spécifié par cette mise en séquence "par couple".
- plusieurs ordonnancements optimaux seront obtenus s'il y a égalité entre certains couples.

Exemple : tâche :

tâche	Ai	Bi
1	9	4
2	8	8
3	7	6
4	2	3

pour le couple	1-2 : min (9,8) > min (8,4)	: 2 précède 1
	1-3 : min (9,6) > min (7,4)	: 3 précède 1
	1-4 : min (9,3) > min (2,4)	: 4 précède 1
	2-3 : min (8,6) > min (7,8)	: 2 précède 3
	2-4 : min (8,3) > min (2,8)	: 4 précède 2
	3-4 : min (7,3) > min (2,6)	: 4 précède 3

La séquence qui vérifie les relations de précédence est : 4-2-3-1

Cette procédure est extrêmement fastidieuse, surtout lorsque le nombre de tâches est élevé car il faut examiner tous les couples de tâches et trouver une séquence qui vérifie les conditions de précédence. L'inégalité ci-dessus nous permet d'affirmer les deux propositions suivantes :

- La dernière tâche de la séquence ne pourra être terminée avant que toutes les autres tâches n'aient été traitées par la première machine plus le temps nécessaire pour réaliser la seconde opération de la dernière tâche de la séquence (puisque A[n] ne peut pour des raisons de routing, chevaucher avec B[n]) ; il s'en suit :

$$Fmax \geqslant \sum_{i=1}^{n} A[i] + B[n] \qquad 1)$$

- La dernière tâche de la séquence ne pourra être terminée avant que toutes les autres tâches n'aient été traitées par la seconde machine plus le temps nécessaire pour que la seconde machine puisse commencer à traiter la première tâche de la séquence ; ce temps est égal au temps nécessaire pour que la première machine traite la première tâche de la séquence ; il s'en suit :

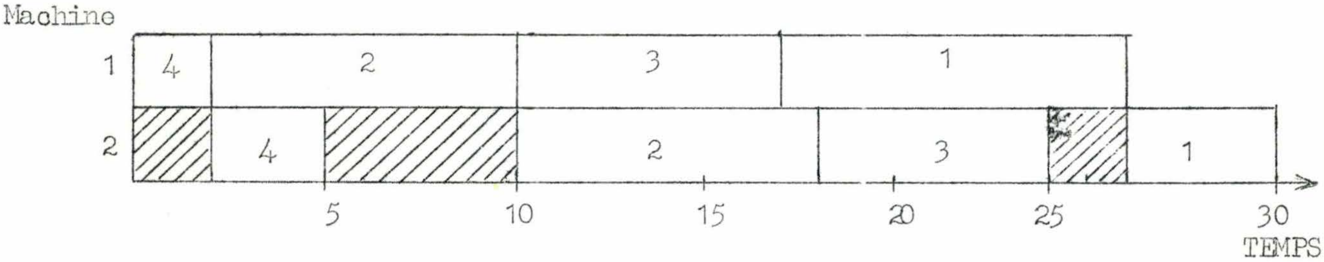
$$Fmax \geqslant A[1] + \sum_{i=1}^{n} B[i] \qquad 2)$$

Comme $\sum_{i=1}^{n} B[i]$ et $\sum_{i=1}^{n} A[i]$ sont constants et restent inaffectés par l'ordonnancement des tâches, on ne peut réduire les limites ci-dessus qu'en influant sur le choix de B[n] et A[1] ; il suffit de choisir parmi les 2n valeurs de Ai et de Bi, la valeur minimum :

- si c'est un Ai, on mettra la tâche i en première position ce qui minimisera 2)
- si c'est un Bi, on mettra la tâche i en dernière position ce qui minimisera 1)

On opère de la même façon après avoir éliminé la tâche ordonnancée.

Dans l'exemple ci-dessus, l'élément dont la valeur est minimale est A_4 ; la tâche 4 sera donc mise en première position ; la tâche 4 éliminée, la valeur minimale restante est B_1 ; appartenant à l'ensemble des Bi, cette valeur entraîne la mise de la tâche 1 en dernière position. La ligne correspondante à cette tâche éliminée, l'élément minimal est B_3 ; on mettra donc la tâche 3 en avant-dernière position ; reste la tâche 2 qu'on intercalera entre les tâches 4 et 3. La séquence optimale est donc 4-2-3-1, dont la valeur de F_{max} est 30.



L'algorithme de Johnson a une <u>interprétation géométrique</u>. Supposons un graphique dont l'abcisse représente le travail de la machine 1 et l'ordonnée celui de la machine 2. Un ordonnancement $\mathcal{J}$ est indiqué par une ligne S comportant :

- des segments horizontaux : de longueur Ai, ils expriment le traitement d'une tâche i par la machine 1.
- des segments verticaux : de longueur Bi, ils expriment le traitement d'une tâche i par la machine 2.

Puisqu'il y a n! ordonnancements possibles, n! lignes S pourraient être dessinées.
La durée d'occupation maximale d'un atelier (F_{max}) est une ligne V qui :

- part de (0,0) pour aboutir en $(\sum_{i=1}^{n} Ai \; , \sum_{i=1}^{n} Bi)$
- doit se situer en dessous de la ligne S correspondante
- est composée de segments verticaux ou horizontaux de façon à ce qu'elle respecte les deux premières conditions.

L'interprétation de ce graphique est simple :

- un segment horizontal indique que la machine 1 est occupée alors que la seconde est inoccupée ;
- un segment vertical indique que la machine 2 est occupée et que la machine 1 est inoccupée ;
- un segment à 45° indique que les deux machines sont occupées.

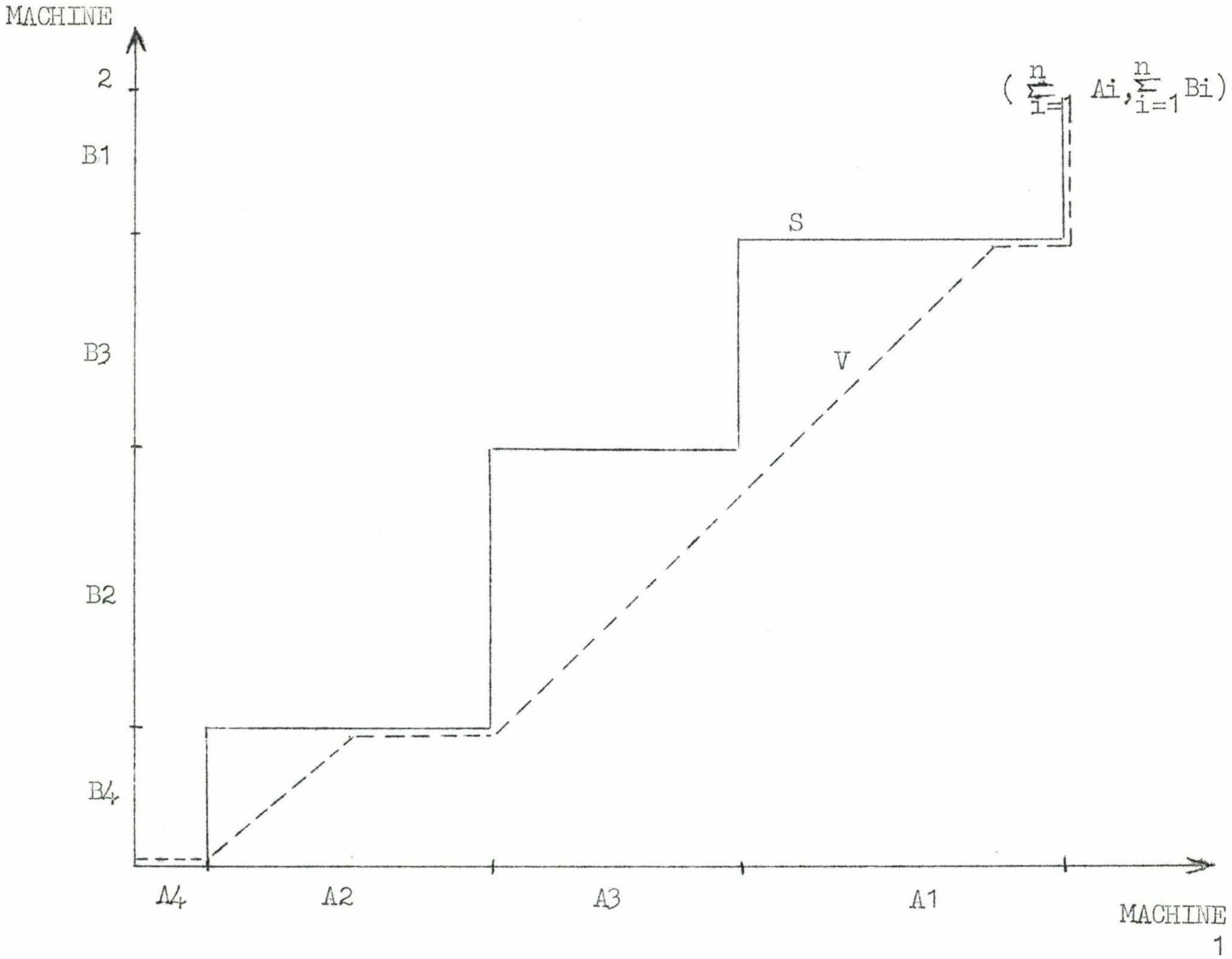
Pour trouver min F_{max}, on a avantage à faire en sorte que les deux machines soit le plus souvent occupées ensemble (la ligne V doit être la plus oblique possible). Au point de départ, la machine 2 sera certainement inoccupée : le premier segment, dont la longueur est A[1] sera donc horizontal alors que le dernier dont la longueur est de B[n] sera vertical.

Avant de dessiner le graphique correspondant à l'exemple ci-dessus, notons que les durées d'attente des tâches n'apparaissent pas sur les axes où les opérations se suivent sans interruption ; on vérifie ainsi les contraintes d'ordonnancement ; les contraintes de routing sont vérifiées en exigeant que la ligne V soit toujours en dessous de la ligne S.

La valeur de F_{max} se calcule aisément : elle est égale à

- la somme des Ai plus la somme des segments verticaux (ici 26 + 4 = 30)
- la somme des Bi plus le somme des segments horizontaux (ici 21 + 9 = 30)

Comme la somme des Ai et des Bi sont constantes, il faut minimiser la somme des segments verticaux ou horizontaux (on retrouve le principe de l'algorithme de Johnson : propositions 1 et 2)



Mitten [1] a généralisé l'algorithme de Johnson en levant l'hypothèse 9 (Titre 2, Chapitre 1) : on n'oblige plus les tâches à commencer le plus rapidement possible, en ne respectant que les contraintes de routing. Dans le problème envisagé, on impose :

- un certain délai entre le début de traitement d'une tâche sur la machine 1 et le début du traitement de cette tâche sur la machine 2 ; (ce délai est appelé "start-lag" ou délai initial)
- un certain délai entre le traitement d'une tâche sur la machine 1 et la fin du traitement de cette tâche sur la machine 2 (ce délai est appelé "stop-lag" ou délai terminal)

Le cas se présente lorsque, dans un processus de production les tâches sont traitées par plusieurs machines dont deux forment des goulots d'étranglement. Comme nous le signalions dans les modèles à une machine, il suffit de s'occuper de ces deux machines ; les raisons d'être de ces goulots peuvent être nombreuses : les deux raisons que nous envisageons sont économiques et non pas techniques :

- les coûts de main d'oeuvre et d'énergie sont tels qu'on ne peut pas se permettre de laisser les machines inoccupées ;
- les moyens et les durées de transport entre les machines sont tels qu'on ne peut pas se permettre <u>un</u> transport par tâche ;

<u>Le délai initial</u> se définit comme étant le temps nécessaire pour traiter une tâche sur la première de ces machines plus le temps nécessaire pour terminer les opérations intermédiaires (qui sont traitées par des machines qui ne forment pas de goulots) plus le temps nécessaire pour accumuler un certain "volume" de travail devant la seconde de ces machines (1er type de goulot).

<u>Le délai terminal</u> se définit comme étant le temps nécessaire pour achever les opérations intermédiaires d'une tâche, pour traiter cette tâche sur la seconde machine et pour accumuler un certain volume de travail à la sortie de cette machine (2ème type de goulot)

<u>Remarque</u> : puisque nous nous occupons de deux machines, nous n'ordonnançons que les tâches qui doivent être traitées par elles ; si quelques interruptions apparaissent, cela provient de ce que d'autres tâches qui ne subissent pas de traitement par ces machines n'interviennent qu'indirectement dans l'ordonnancement. (par ce que nous avons appelé le "volume de travail").

<u>Algorithme de Mitten :</u>

- si x_i et y_i représente le délai initial et terminal de la tâche i (par définition $x_i \geqslant Ai$ et $y_i \geqslant Bi$, on calcule pour tout i :

---

[1] MITTEN L.G. : "Sequencing n jobs on 2 Machines with Arbitrary Time-Lags" Management Science vol.5 n° 3 ; avril 1959.

$m_i = \max(x_i - Ai, y_i - Bi)$; cette valeur donne l'intervalle de temps minimum (imposé) séparant la fin du traitement de la tâche i sur la machine 1 et le début du traitement de cette même tâche sur la machine 2 ;

- on divise les tâches en deux sous-ensembles :
  - M_1 est l'ensemble des tâches pour lesquelles $Ai < Bi$; les tâches appartenant à cet ensemble sont ordonnancées de telle sorte que, si $m_i + Ai < m_j + Aj$, la tâche i précédera la tâche j ;
  - M_2 est l'ensemble des tâches pour lesquelles $Ai \geqslant Bi$; les tâches appartenant à cet ensemble sont ordonnancées de telle sorte que, si $m_i + Bi \geqslant m_j + Bj$, la tâche i précédera la tâche j.

La séquence optimale consiste à faire précéder l'ensemble M_2 par M_1.

Cet algorithme ressemble fort à celui de Johnson (qui consiste à placer une tâche, dont la durée de traitement sur la première machine est plus courte que celle de la seconde machine, au début de la séquence ; dans le cas contraire, on placera cette tâche en dernière position).

Exemple :

	Ai	Bi	xi	yi
tâche 1 :	1	2	6	4
2 :	3	4	3	5
3 :	5	4	9	7
4 :	3	2	10	9

$m_1 = \max(6-1, 4-2) = 5$

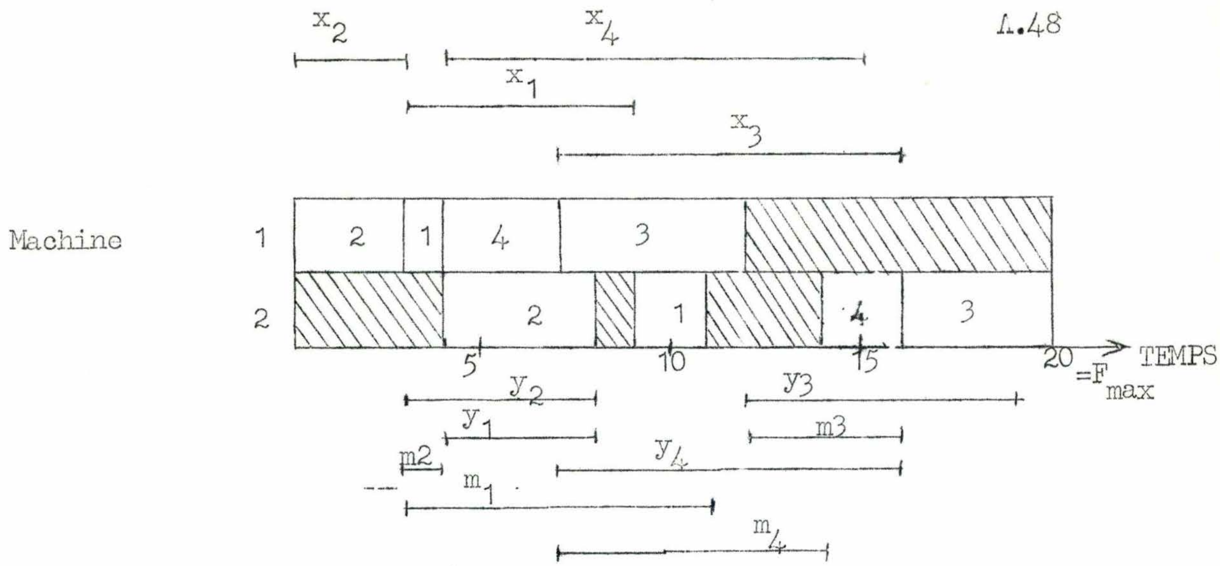
$m_2 = \max(3-3, 5-4) = 1$

$m_3 = \max(9-5, 7-4) = 4$

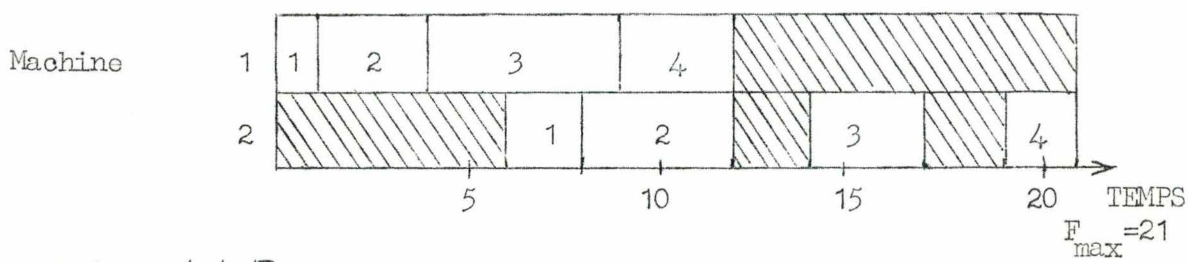
$m_4 = \max(10-3, 9-2) = 7$

Appartiennent à l'ensemble M_1, les tâches 1 et 2 : la tâche 2 précédera la tâche 1 puisque $5 + 3 < 4 + 1$
Appartiennent à l'ensemble M_2, les tâches 3 et 4 : la tâche 4 précédera la tâche 3 puisque $7 + 2 > 4 + 4$

L'ordonnancement optimal par rapport à F_{max} est donc 2-1-4-3



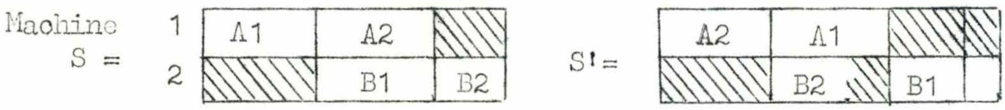
L'extrémité initiale de x_i et l'extrémité terminale de y_i donne la date au plus tôt à laquelle la seconde opération de la tâche i peut commencer et la date au plus tôt à laquelle la seconde opération de la tâche i doit être terminée,; m_i mesure l'intervalle de temps qui sépare l'achèvement d'une tâche sur la première machine et le début de cette tâche sur la seconde.
Le problème consiste en fait à trouver un ordonnancement qui, tout en respectant les contraintes que sont les m_i n'allonge pas le délai entre les deux opérations d'une même tâche ;
Dans l'ordonnancement suivant, nous avons repris l'ordonnancement de Johnson qui respecterait les délais imposés ; F_{max} y est supérieur.



§ 2 : Le problème $n/2/F/\overline{F}$

Un ordonnancement qui cherche à minimiser $\overline{F}$ est difficile à trouver même dans le cas de deux machines. L'algorithme de Johnson est inapplicable : dans l'exemple donné à propos de cet algorithme, $\overline{F}$=19,25 alors que l'ordonnancement 4-3-2-1 donne $\overline{F}$=18,75.

Si nous n'avions que deux tâches, deux ordonnancements sont possibles



Avec S : la somme des $F_i = 2\bar{F} = 2A_1 + B_1 + B_2 + \max(A_2,B_1)$

Avec S': la somme des $F_i = 2\bar{F}' = 2A_2 + B_1 + B_2 + \max(A_1,B_2)$

La tâche 1 précédera la tâche 2 si $\bar{F} < \bar{F}'$ c-a-d si :

$$2A_1 + \max(A_2,B_1) < 2A_2 + \max(A_1,B_2) \text{ ou si}$$

$$2A_2 - 2A_1 + \max(A_1,B_2) - \max(A_2,B_1) > 0 \qquad (1)$$

Dans l'exemple donné à propos de l'algorithme de Johnson :

	Ai	Bi
tâche 1 :	9	4
2 :	8	8
3 :	7	6
4 :	2	3

Pour le couple	la valeur de l'inégalité est :	il s'en suit :
1-2	$2x8 - 2x9 + \max(9,8) - \max(8,4) < 0$	2 précède 1
1-3	$2x7 - 2x9 + \max(9,6) - \max(7,4) < 0$	3 précède 1
1-4	$2x2 - 2x9 + \max(9,3) - \max(2,4) < 0$	4 précède 1
2-3	$2x7 - 2x8 + \max(8,6) - \max(7,8) < 0$	3 précède 2
2-4	$2x2 - 2x8 + \max(8,3) - \max(2,8) < 0$	4 précède 2
3-4	$2x2 - 2x7 + \max(7,3) - \max(2,6) < 0$	4 précède 3

La séquence optimale est donc 4-3-2-1, dont la valeur de $\bar{F}$ est 18,75. Nous noterons (comme nous l'avons fait à propos de l'algorithme de Johnson) le caractère fastidieux de cette méthode ; si elle garantit un ordonnancement optimal, n'existe-t-il pas de condition plus simple ? Si nous examinons le cas d'une seule machine, $\bar{F}$ est minimisé en ordonnançant les tâches selon la discipline SPT. Malheureusement comme nous avons affaire à deux machines, il ne suffit pas de s'intéresser aux durées de traitement de la première opération de chaque tâche : même si $A_2 > A_1$, l'inégalité (1) peut être confirmée ou infirmée selon les valeurs prises par B_1 et B_2. L'inégalité (1) n'est pas nécessairement confirmée non plus si on place la tâche 1 en première position lorsque $(A_1+B_1) < (A_2+B_2)$

Exemple : $B_1 = 0$; $A_2 = B_2$; $A_1 = A_2 + B_2 - \varepsilon$

où $A_1, A_2, B_2 > 0$
ε = valeur infiniment petite mais positive

(1) $2A_2 - 2A_1 + \max(A_1,B_2) - \max(A_2,B_1) > 0$

En substituant dans cette inégalité les Ai et Bi par leurs valeurs :

$= 2A_2 - 2A_1 + \max(A_1,B_2) - A_2 > 0$?
$= A_2 - 2A_1 + \max(A_1,B_2) > 0$?
$= A_2 - 2A_1 + \max(A_1,A_2) > 0$?
$= -2A_1 + \max(A_1-A_2;\ 0) > 0$? ; $-2A_1 + \max(A_2 + B_2 - \varepsilon - A_2, 0) > 0$?
$= -2A_1 + \max(B_2-\ \ , 0) > 0$? ; $-2A_1 + B_2 - \varepsilon > 0$? ; $2A_2 - 2B_2 + 2\varepsilon + B_2 - \varepsilon > 0$?
$= -2B_2 - B_2 + \varepsilon > 0$? ; $-3B_2 + \varepsilon > 0$?

On peut négliger   puisqu'il est certainement infériuer à B_2 :
$-3B_2 > 0$?
$B_2 < 0$: impossible par hypothèse.

L'inéquation 1 ne se vérifie que dans le cas ou la première opération de chaque tâche est plus courte que la seconde. Si nous supposons (dans le cas de deux tâches) que :

$A_2 \geqslant A_1 >$ et $B_2 > B_1$
ou
$A_2 > A_1$ et $B_2 \geqslant B_1$

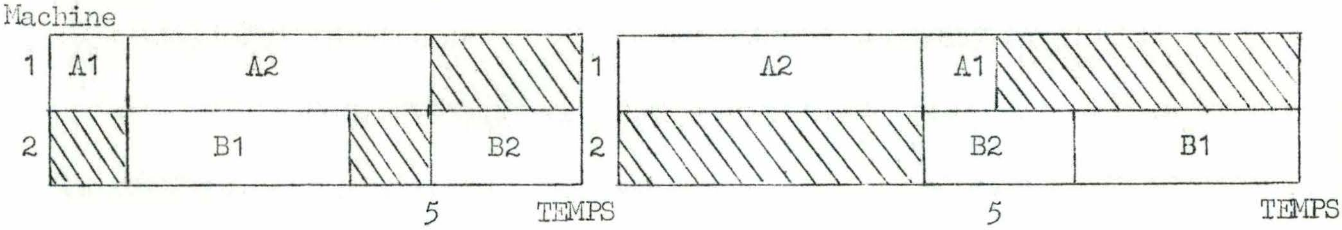
on vérifie que :

* $2A_2-2A_1$ est positif (par hypothèse)
* $\max(A_1,B_2) - \max(A_2,B_1)$ est positif : quatre possibilités s'offrent à nous :
  1. $A_1 > B_2$ et $A_2 > B_1$: l'inéquation devient : $2A_2-2A_1+A_1-A_2 = A_2-A_1$; elle est positive par hypothèse ;
  2. $A_1 > B_2$ et $B_1 > A_2$: D'après les hypothèses $A_2 > A_1$; les conditions données impliqueraient que $B_1\ \ B_2$ ce qui est contraire aux hypothèses
  3. $B_2 > A_1$ et $A_2 > B_1$: l'inéquation devient $2A_2-2A_1+B_2-A_2 = A_2-A_1+B_2-A_2$; elle est positive par hypothèse ;
  4. $B_2 > A_1$ et $B_1 > A_2$: l'inéquation devient : $2A_2-2A_1+B_2-B_1$; elle est positive par hypothèse ;

En conclusion, nous pouvons affirmer que cette condition est suffisante mais pas nécessaire ; dans l'exemple qui suit, les conditions ne sont pas respectées et pourtant l'ordonnancement optimal consiste à placer la tâche 1 avant la tâche 2 :

$A_1 = 1$; $B_1 = 3$; $A_2 = 4$; $B_2 = 2$

Les deux ordonnancements sont :



F1 = 4 ; F2 = 7 $\qquad\qquad$ F1 = 9 ; F2 = 6

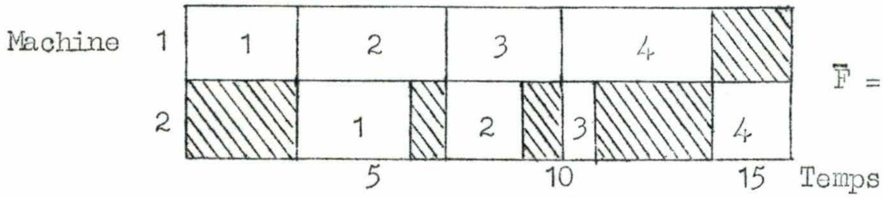
$\bar{F} = 5{,}5$ $\qquad\qquad$ $\bar{F} = 7{,}5$

Lorsque nous avons n tâches à ordonnancer, l'argumentation est semblable à celle de l'algorithme de Johnson :

$$F[1] = A[1] + B[1] = X[1] + B[1]$$
$$F[2] = X[1] + B[1] + X[2] + B[2]$$
$$\vdots$$
$$F[j] = \sum_{i=1}^{j} B[i] + \sum_{i=1}^{j} X[i] = \sum_{i=1}^{j} B[i] + \max(Y1,Y2,..Yj)$$
$$\bar{F} = \frac{1}{n}\left[\sum_{j=1}^{n}\sum_{i=1}^{j} B[i] + \sum_{j=1}^{n} \max(Y1,Y2,...Yj)\right]$$

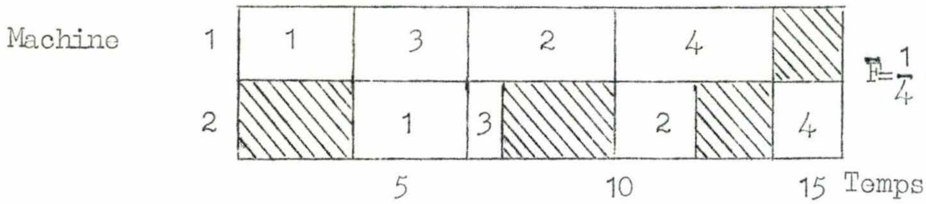
On peut par l'emploi de cette expression, montrer que pour des tâches adjacentes, I et J, I précédera J si $A_J \geqslant A_I$ et $B_J \geqslant B_I$; elle ne constitue pourtant pas une solution parce qu'elle ne produit pas nécessairement un ordonnancement complet des tâches (en ne comparant que des tâches <u>adjacentes</u>)

Soit l'ordonnancement (quelconque) :



$\bar{F} = \frac{1}{4}(6+9+11+16) = 42/4$

On peut améliorer cet ordonnancement en permutant les tâches 3 et 2 puisqu'elles vérifient les conditions citées. Aucun autre couple ne les remplit, ce qui ne signifie pas que l'ordonnancement obtenu soit optimal.



$\bar{F} = \frac{1}{4}(6+7+12+16) = 41/4.$

Ignall et Schrage [1] ont appliqué la technique du Branch and Bound à ce problème. Malheureusement (contrairement à F_{max}) un ordonnancement optimal (vis à vis de $\bar{F}$) de n-1 tâches n'est pas nécessairement un bon point de départ pour l'ordonnancement des n tâches : si, dans un ordonnancement partiel, on spécifie une nouvelle liaison, la limite inférieure de F_{max} de ce nouvel ordonnancement ne peut, en aucun cas, être inférieure à celle de l'ordonnancement partiel initial. Vis à vis de $\bar{F}$, par contre, l'ordonnancement optimal des deux premières tâches de l'exemple ci-dessus les place dans l'ordre 1-2 ; ajouter la tâche 3 inverse cet ordre, l'ordonnancement optimal étant 3-2-1.

---

[1] IGNALL E. et SCHRAGE L. : "Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems" ; Journal of the O.R.S.A. vol. 13 n° 3 ; mai 1965.

Section 6 : Génération d'ordonnancements actifs et non-retardés

Comme nous le signalions dans le Titre II, chapitre 4, le concept de base des procédures de génération est "l'ensemble des opérations ordonnançables". A tout instant, cet ensemble, noté $\{S_{so}\}$ est un sous-ensemble des G opérations où G est l'ensemble de toutes les opérations à ordonnancer. $\{S_{so}\}$ contient toutes les opérations dont les prédécesseurs ont été ordonnancés: une opération appartient à $\{S_{so}\}$ si une date de départ a été assignée à chacune des opérations précédentes de cette tâche. Cet ensemble ne contient donc que n opérations au maximum (une par tâche). Au moment 0, $\{S_{so}\}$ contient la première opération de chacune des tâches : lorsqu'une de celles-ci est ordonnancée, elle est remplacée parla seconde opération de cette tâche : l'ordonnancement est achevé lorsque $\{S_{so}\}$ est un ensemble vide. $\{S_{so}\}$ peut être partitionné de deux façons :

§ 1. Partitionnement par tâche

On divise $\{S_{so}\}$ en n sous-ensembles, chacun d'eux ne comportant qu'une opération au maximum. Le choix entre plusieurs sous-ensembles non vides se fait sur base d'une règle de décision. (cfr. Titre II). Si n tâches sont à ordonnancer et que la procédure de génération consiste à ordonnancer toutes les opérations d'une tâche de façon consécutive, n! ordonnancements peuvent être tracés.

§ 2. Partitionnement par machine

On divise $\{S_{so}\}$ en m sous-ensembles, notés $\{S^k_{so}\}$ où k varie de 1 à m ; ils contiennent les opérations ordonnançables sur la machine k au moment t, c-a-d les opérations qui attendent devant la machine k et celles dont la précédente est en cours de traitement sur une autre machine. Au moment 0, les n premières opérations sont donc distribuées en m sous-ensembles $\{S^k_{so}\}$. Ces sous-ensembles passent alternativement d'un état vide à un état occupé. L'ordonnancement sera achevé lorsque tous ces sous-ensembles sont simultanément vides.

Quelle machine faut-il choisir pour la prochaine assignation ? (quel sous-ensemble $\{S^k_{so}\}$ est-il prioritaire ?) Si une règle de décision sert à opérer un choix entre les éléments appartenant à un même $\{S^k_{so}\}$, il faut encore établir une priorité entre ces différents sous-ensembles.

Si on définit $\{S_{ip}\}$ comme l'ensemble des opérations en traitement, une opération j y sera placée dès qu'elle quitte $\{S_{so}\}$. Si $\{S_{ip}\}$ est partitionné en m sous-ensembles, une opération au maximum sera en traitement sur une des m machines : chaque $\{S^k_{ip}\}$ contient donc un élément au maximum. Au moment 0, chaque $\{S^k_{ip}\}$ contient une opération imaginaire dont la durée de traitement est nulle. Dès qu'une opération est placée dans un $\{S^k_{ip}\}$, elle chasse l'occupant antérieur, puisqu'elle ne peut y entrer que lorsque l'occupant antérieur a terminé son traitement.

Soit C_k le moment où l'opération appartenant à $\{S^k_{ip}\}$ est terminée,

s_{jk} la date de départ potentiel de l'opération j appartenant à $\{S^k_{so}\}$. Cette date est celle de l'achèvement de l'opération antérieure de cette tâche (ou 0, s'il s'agit de la première opération d'une tâche,

p_{jk} la durée de traitement de l'opération j contenue dans $\{S^k_{so}\}$.

Il s'en suit :

i) $\max(C_k, s_{jk})$ est la date de départ au plus tôt de l'opération j contenue dans $\{S^k_{so}\}$;

ii) $\max(C_k, s_{jk}) + p_{jk}$ est la date d'achèvement au plus tôt de l'opération j contenue dans $\{S^k_{so}\}$.

Un ordonnancement non retardé sera obtenu en choisissant un ensemble non vide S^k_{so} sur base du critère :

$$\min_k \min_{j \in S^k_{so}} (\max(C_k, s_{jk}))$$

Il faut en outre que l'opération sélectionnée aie une valeur de s_{jk} égale à la valeur obtenue par cette équation.

Un ordonnancement actif sera obtenu en choisissant un ensemble non vide $\{S^k_{so}\}$ sur base du critère :

$$\min_k \min_{j \in S^k_{so}} (\max(C_k, s_{jk}) + p_{jk})$$

Il faut en outre que l'opération sélectionnée aie une valeur de s_{jk} inférieure à la valeur obtenue par cette équation.

On trouvera dans l'annexe V, un exemple d'application des procédures de génération d'ordonnancement actifs en non-retardés.

Remarque : La dernière procédure de génération envisagée engendre tous les ordonnancements actifs et non-retardés. Nous n'avons fait que choisir la machine à propos de laquelle une assignation doit être faite. Le problème ne serait résolu que si à tout moment, il n'y avait qu'une opération à ordonnancer ; nous n'avons rien dit sur la manière de choisir une opération parmi toutes celles appartenant à un même $\{S^k_{so}\}$. Une fois la machine choisie, il n'est, d'aucune façon, automatique de choisir celle dont la valeur de s_{jk} est minimale car ce choix supposerait l'application d'un système de priorité de type : premier arrivé, premier servi.

ANNEXE IV : Exemples de résolution d'un problème $n/3/F/F_{max}$

Section 1 : L'algorithme de Johnson appliqué au problème $n/3/F/F_{max}$

	MACHINES A	B	C
tâches : 1	8	5	4
2	10	6	9
3	6	2	8
4	7	5	6
5	11	4	5

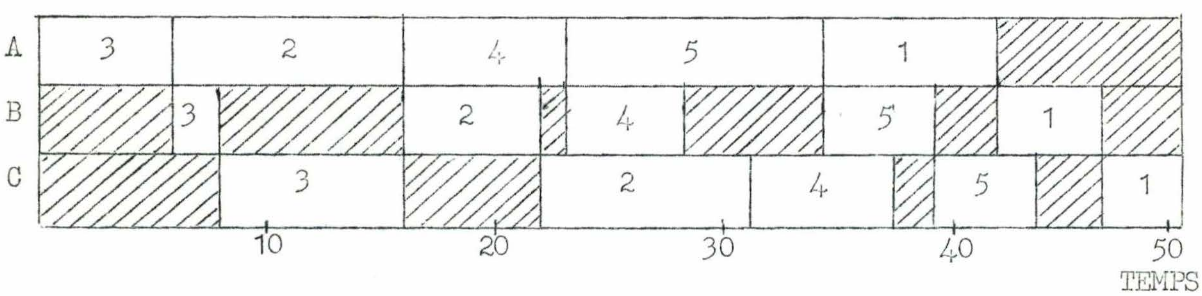
L'algorithme de Johnson est applicable puisque une des conditions est remplie : min A_i (6) max B_i (6) (cfr Titre II chap. 3)
On réduit le problème à celui de deux machines en remplaçant A_i par A_i+B_i et B_i par B_i+C_i.

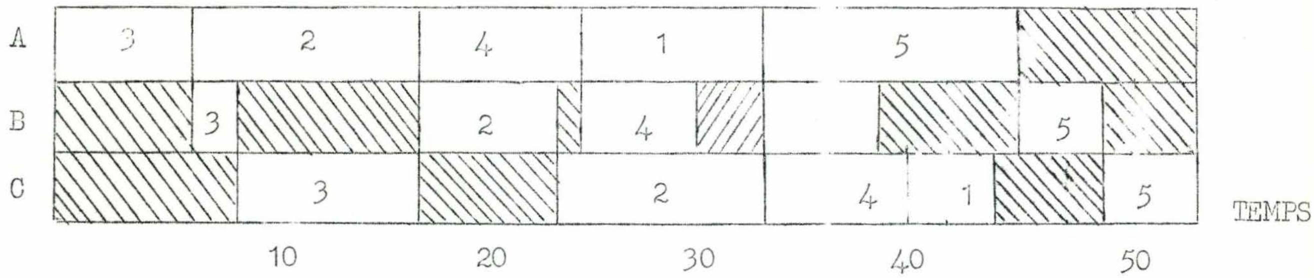
	(A_i+B_i)	(B_i+C_i)
tâche : 1	13	9
2	16	15
3	8	10
4	12	11
5	15	9

Si on applique l'algorithme de Johnson comme nous l'avons fait dans le cas de deux machines, on trouve les ordonnancements optimaux suivants :

3-2-4-5-1
3-2-4-1-5

Les deux diagrammes-machines permettent de trouver facilement la valeur de F_{max} (qui est minimale) : 51





Remarque : Le théorème de Roy (Titre II, chap 3) est d'application ; la séquence des opérations est donc identique sur les trois machines.

Section 2 : Algorithme du Branch and Bound appliqué au problème $n/3/F/F_{max}$

(cfr Annexe III) : Nous ne reviendrons pas sur le principe de cet algorithme. Les modalités de calcul, utilisés ici, sont différentes.

Soit les durées de traitement de 6 tâches sur 3 machines :

	MACHINES A	B	C
tâche : 1	6	7	3
2	12	2	3
3	4	6	8
4	3	11	7
5	6	8	10
6	2	14	12

Remarquons au passage que l'algorithme de Johnson est inaplicable puisqu' aucune des deux conditions n'est réalisée. (Titre II chap 3)

1ère partie

La première partie de cet algorithme consiste à trouver un ordonnancement qui, s'il n'est pas optimal, en soit très proche. Nous définissons les variables suivantes :

1. $g_i^! = A_i + B_i + C_i + \sum_{k=1}^{n} C_{k \neq i}$

$g_i^!$ exprime le temps nécessaire pour que la tâche i soit traitée sur toutes les machines plus le temps des tâches restant à traiter sur la dernière machine ; il s'agit en fait d'une limite inférieure de F_{max} ; F_{max} est certainement supérieur à la $\sum_{i=1}^{n} C_i$ car une égalité entre ces deux valeurs signifierait non seulement qu'aucun temps mort ne s'intercalle entre les tâches soumises à la machine C mais aussi que cette machine commence à travailler au moment 0 ce qui est impossible ; les contraintes de routing sont telles que la première tâche de la séquence ne sera traitée sur la machine C qu'au moment où les deux premières opérations de la tâche i sont terminées ; c'est pour cette raison que nous ajoutons à $\sum_{i=1}^{n} C_i$, A_i+B_i.

2. $g''_i = A_i+B_i+\sum_{k=1}^{n} B_{k\neq i} + \min C_{k\neq i}$

g''_i exprime le temps nécessaire pour que la tâche i soit traitée sur les deux premières machines plus le temps nécessaire pour que la machine B traite les autres tâches plus le temps de traitement de la dernière tâche sur la machine C. Comme nous ne connaissons pas cette dernière tâche, on prend la valeur minimale des $C_{k\neq i}$; il s'agit à nouveau d'une limite inférieure de F_{max} qui est certainement supérieure à la somme des durées de traitement sur la machine B ; celle-ci ne pourra commencer que lorsque la tâche i aura été terminée sur la machine A ; comme la dernière tâche de la séquence (que nous ne connaissons pas) ne pourra être traitée par la machine C que lorsqu'elle aura terminé sa seconde opération (sur la machine B), on ajoute une valeur de C_k.

3. $g'''_i = A_i+\sum_{k=i}^{n} A_{k\neq i} + \min(B_{k\neq i}+ C_{k\neq i})$

g'''_i exprime le temps nécessaire pour que la tâche i soit traitée sur la première machine, plus le temps nécessaire pour que la machine A traite les autres tâches plus le temps de traitement de la dernière tâche de la séquence sur les deux dernières machines. Comme nous ne connaissons pas cette tâche, on prend la valeur minimum de $(B_{k\neq i} + C_{k\neq i})$; F_{max} est certainement supérieur à la somme des A_i, car les machines B et C ne peuvent commencer le traitement de la dernière tâche que lorsque cette dernière a été traitée respectivement par les machines A et B.

4. Si $g_i = \max(g'_i, g''_i, g'''_i)$, g_i exprime la valeur minimale que pourrait prendre F_{max}. Si la tâche i était la première de la séquence , on choisira comme première tâche de la séquence celle pour laquelle g_i est minimisé ; min g_i. Appelons cette tâche I.
Il reste donc (n-1) tâches à ordonnancer ; comme chacune de ces tâches est susceptible d'occuper la seconde position dans la séquence, nous aurons (n-1) permutations possibles : I1,12,I3,...I(I-1), I(I+1), ...
Pour chacune d'elles, on applique l'algorithme du flux maximal, par lequel on calcule les dates au plus tôt auxquelles les deux tâches faisant partie de la permutation pourraient être terminées sur chacune des trois machines. Un nouveau calcul des g_i est effectué ou i représente non plus une tâche mais une permutation. Celle pour laquelle g_i est minimisée est choisie : on a donc spécifié la tâche qui occupera la deuxième position position dans la séquence. On continue de la même façon (calcul du flux maximal et des g_i) jusqu'à ce que toutes les tâches soient ordonnancées. On obtient ainsi un ordonnancement, mais comme nous ne savons pas s'il est optimal, il s'agira, dans la seconde partie de l'algorithme de voir si un meilleur ordonnancement ne peut être trouvé. Reprenons notre exem-

ple et appliquons lui la première partie de l'algorithme.

ETAPE 1 :

i	A	B	C	g'_i	g''_i	g'''_i	g_i
1	6	7	3	6+7+43=56	6+48+3=57	33+5=38	57
2	12	2	3	12+2+43=57	12+48+3=63	33+10=43	63
3	4	6	8	4+6+43=53	4+48+3=53	33+5=38	55
4	3	11	7	3+11+43=57	3+48+3=54	33+5=38	57
5	6	8	10	6+8+43=57	6+48+3=54	33+5=38	57
6	2	14	12	2+14+43=59	2+48+3=53	33+5=38	59
Σ =	33	48	43				

Le minimum des g_i est g_3 ; la tâche sera donc placée en première position dans la séquence.

ETAPE 2 :

Les permutations à partie de 3 sont : 31,32,34,35,36. Si on leur applique l'algorithme du flux maximal en respectant les contraintes de routing (arcs horizontaux) et d'ordonnancement (arcs verticaux) on a :

Cela signifie que les tâches 3 et 1 seront terminées au plus tôt sur les machines A,B,C au moment 10,17,21

permutation 31 :

$$\begin{array}{c|ccccc} i & A & & B & & C \\ \hline 3 & 4 & \rightarrow & 6 & \rightarrow & 8 \\ & \downarrow & & \downarrow & & \downarrow \\ 1 & 6 & \rightarrow & 7 & \rightarrow & 3 \\ & 10 & & 17 & & 21 \end{array}$$

permutation 32 :

$$\begin{array}{c|ccccc} 3 & 4 & \rightarrow & 6 & \rightarrow & 8 \\ & \downarrow & & \downarrow & & \downarrow \\ 2 & 12 & \rightarrow & 2 & \rightarrow & 3 \\ & 16 & & 18 & & 21 \end{array}$$

permutation 34 :

$$\begin{array}{c|ccccc} 3 & 4 & \rightarrow & 6 & \rightarrow & 8 \\ & \downarrow & & \downarrow & & \downarrow \\ 4 & 3 & \rightarrow & 11 & \rightarrow & 10 \\ & 7 & & 21 & & 28 \end{array}$$

permutation 35 :

$$\begin{array}{c|ccccc} 3 & 4 & \rightarrow & 6 & \rightarrow & 8 \\ & \downarrow & & \downarrow & & \downarrow \\ 5 & 6 & \rightarrow & 8 & \rightarrow & 10 \\ & 10 & & 18 & & 28 \end{array}$$

permutation 36 :

$$\begin{array}{c|ccccc} 3 & 4 & \rightarrow & 6 & \rightarrow & 8 \\ & \downarrow & & \downarrow & & \downarrow \\ 5 & 2 & \rightarrow & 14 & \rightarrow & 12 \\ & 6 & & 24 & & 36 \end{array}$$

Les nouvelles valeurs g_i sont :

i	A	B	C	g'	g''
31	10	17	21	21+(43-8- 3)=53	17+(48-6- 7)+3=55
32	16	18	21	21+(43-8- 3)=53	18+(48-6- 2)+3=61
34	7	21	28	28+(43-8- 7)=56	21+(48-6-11)+3=55
35	10	18	28	28+(43-8-10)=53	18+(48-6- 8)+3=55
36	6	24	36	36+(43-8-12)=59	24+(48-6-14)+3=55

i	A	B	C	g'''	g_i
31	10	17	21	10+(33-4- 6)+ 5=38	55
32	16	18	21	16+(33-4-12)+10=43	61
34	7	21	28	7+(33-4- 3)+ 5=38	56
35	10	18	28	10+(33-4- 6)+ 5=38	55
36	6	24	36	6+(33-4- 2)+ 5=38	59

Pour la permutation 31 :

1) g_i' donne la limite inférieure de F_{max} si les tâches 3 et 1 sont ordonnancées en premier lieu sur la dernière machine. Elle se compose de la date au plus tôt à laquelle ces deux tâches seront terminées sur la machine C (21) plus le temps nécessaire pour traiter les tâches 2, 4, 5, 6 sur la machine C (3+7+10+12 ou 43-8-3).

2) g_i'' donne la limite inférieure de F_{max} si les tâches 3 et 1 sont ordonnancées en premier lieu sur la deuxième machine. Elle se compose de la date au plus tôt à laquelle ces deux tâches seront terminées sur la machine B (17) plus le temps nécessaire pour traiter les tâches 2, 4, 5, 6 sur cette même machine (48-6-7) plus le temps nécessaire pour que la machine C traite la dernière tâche de la séquence. Comme nous ne connaissons pas cette tâche mais qu'il ne s'agit en aucun cas des tâches 1 et 3, on prend la valeur minimum de C_2, C_4, C_5, C_6. Il s'agit de C_2 dont la valeur est 3.

3) g_i''' donne la limite inférieure de F_{max} si les tâches 3 et 1 sont ordonnancées en premier lieu sur la première machine. Elle se compose de la date au plus tôt à laquelle ces deux tâches seront terminées sur la machine A (10) plus le temps nécessaire pour traiter les tâches 2, 4, 5, 6 sur cette machine (33-4-6) plus le temps nécessaire pour que les machines B et C traitent la dernière tâche de la séquence. Comme nous ne connaissons pas cette tâche mais qu'il ne peut en aucun cas s'agir des tâches 3 et 1, on prend la valeur minimum de B_2+C_2, B_4+C_4, B_5+C_5, B_6+C_6. Il s'agit de B_2+C_2 dont la valeur est 5.

4) Max (g_i' , g_i'' , g_i''') = g_i donne la limite inférieure de F_{max} si la séquence commence par la permutation i. On choisira min(g_i^{max}), en l'occurence 35 ou 31. Les tâches 5 ou 1 seront choisie pour être placée en seconde position dans la séquence.
Prenons 31.

ETAPE : 3 :

On prend toutes les permutations possibles qui ont 31 comme point de départ ; 312, 314, 315, 316. L'algorithme du flux maximal permet de calcules les dates au plus tôt auxquelles le traitement de ces permutations seront terminées sur chacune des trois machines. Notons qu'il n'est pas nécessaire dans ce calcul d'examiner les tâches séparément ; on peut se servir des dates au plus tôt trouvées antérieurement ; mais il n'y a plus d'arcs horizontaux symbolisant le passage de la permutation de la machine A vers la machine B et de la machine B vers la machine C car les valeurs inscrites n'expriment plus les durées de traitement mais bien des dates (ce qui a pour effet de cumuler les durées de traitement antérieures).

permutation 312

i	A	B	C
31	10	17	21
2	12	2	3
	22	24	27

permutation 314

i	A	B	C
31	10	17	21
4	3	11	7
	13	28	35

permutation 315

31	10	17	21
5	6	8	10
	16	25	35

permutation 316

31	10	17	21
6	2	14	12
	17	31	43

Les nouvelles valeurs g_i sont :

i	A	B	C	g'	g''
312	22	24	27	27+(43-8-3- 3)=56	24+(48-6- 7- 2)+7=64
314	13	28	35	35+(43-8-3- 7)=60	28+(48-6- 7-11)+3=55
315	16	25	35	35+(43-8-3-10)=57	25+(48-6- 7- 8)+3=55
316	12	31	43	31+(43-8-3-12)=63	31+(48-6- 7-14)+3=55

i	A	B	C	g'''	g
312	22	24	27	22+(33-7-3-12)+18=51	64
314	13	28	35	13+(33-7-3- 3)+5=38	60
315	16	25	35	16+(33-7-3- 6)+5=38	57
316	12	31	43	12+(33-7-3- 2)+5=38	63

La meilleure solution est 315 et c'est elle que nous continuons à explorer. Les permutations à envisager sont 3152, 3154, 3156.

ETAPE 4 :

permutation 3152

i	A	B	C
315	16	25	35
	↓	↓	↓
2	12 →	2 →	3
	28	30	38

permutation 3154

i	A	B	C
315	16	25	35
	↓	↓	↓
4	3 →	11 →	7
	19	36	43

permutation 3156

i	A	B	C
315	16	25	35
	↓	↓	↓
6	2 →	14 →	12
	18	39	51

Les nouvelles valeurs g_i sont :

i	A	B	C	g'	g''
3152	28	30	38	38+(43-8-3-10- 3)=57	30+(48-6-7-8- 2)+7=62
3154	19	36	43	43+(43-8-3_10- 7)=58	36+(48-6-7-7-11)+3=55
3156	18	39	51	51+(43-8-3-10-12)=61	39+(48-6-7-8-14)+3=55

i	A	B	C	g'''	g
3152	28	30	38	28+(33-7-3-6-12)+18=31	62
3154	19	36	43	19+(33-7-3-6- 3)+ 5=38	58
3156	18	39	51	18+(33-7-3-6- 2)+ 5=38	61

La meilleure solution est 3154, qui forme la base des permutations 31542, 31546.

ETAPE 5 :

i	A	B	C
3154	19	36	43
	↓	↓	↓
2	12 →	2 →	3
	31	38	46

i	A	B	C
3154	19	36	43
	↓	↓	↓
6	2 →	14 →	12
	21	50	62

Les nouvelles valeurs g_i sont :

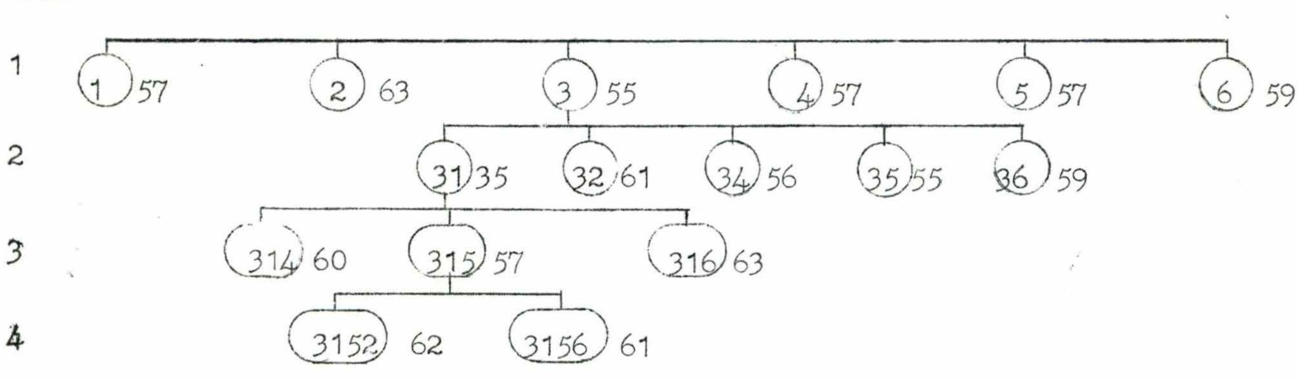
i	A	B	C	g'	g''	g'''	g
31542	31	38	46	46+12=58	34+14+12=64	31+ 2+26=59	64
31546	21	50	62	62+3 =65	50+ 2+ 3=55	31+12+ 5=38	65

La valeur minimale des g exprime ici la valeur de F_{max}, si nous prenons l'ordonnancement 315426. Appelons cette valeur g_1 : elle est égale à 64.

2ème partie

Nous avons un ordonnancement mais il faut voir si nous ne pouvons pas l'améliorer, en d'autres mots voir s'il est optimal. Nous pouvons maintenant passer à la routine d'élimination par laquelle toutes les permutations incomplètes (celles qui ne contiennent pas toutes les tâches) qui ont une valeur de F_{max} supérieure à 64 peuvent être éliminées. Puisque nous ne cherchons qu'un ordonnancement optimal, on peut éliminer les solutions égales à cette valeur. Nous éliminons également dans les investigations ultérieures la permutation 3154 dont toutes les possibilités ont été analysées. Pour avoir une vision claire des permutations à explorer, nous construisons une arborescence dont les sommets représentent les permutations qui restent à analyser. A chacune de ces permutations sont associées les valeurs de F_{max} qui leur correspond.

ETAPE :



Grâce aux suppressions, nous sommes remontés d'une étape, celle où (n-2) tâches sont ordonnancées. La deuxième partie de l'algorithme consiste à y explorer toutes les permutations. Dès que nous rencontrons une permutation incomplète dont la valeur est supérieure ou égale à 64, nous pouvons l'éliminer ; il suffit qu'une valeur des g le soit. Si nous rencontrons une permutation incomplète dont la valeur est inférieure à 64, il faut continuer l'exploration ; si la permutation complète a une valeur (g_2) inférieure à 64, nous aurons trouvé un meilleur ordonnancement. Celui-ci va nous permettre de supprimer tous les sommets dont la valeur est égale ou supérieure à g_2. On remontera ainsi d'étapes en étapes jusqu'à ce qu'il ne soit plus possible de trouver un g_k inférieur à un g_h antérieurement trouvé et g_k sera la valeur de F_{max} de l'ordonnancement optimal.

1) Amélioration de g_1 ($g_1 = 64$)

Nous remontons à l'étape 4 où nous devons explorer les permutations 3152 et 3156 ; la valeur de cette dernière étant inférieure à la première, nous commençons par explorer 3156 qui forme la base des permutations 31562 et 31564.

permutation 31562

i	A	B	C
3156	18	39	51
2	12 →	2 →	3
	30	41	53

permutation 31564

i	A	B	C
3156	18	39	51
4	3 →	11 →	7
	21	50	58

Valeurs de g_i :

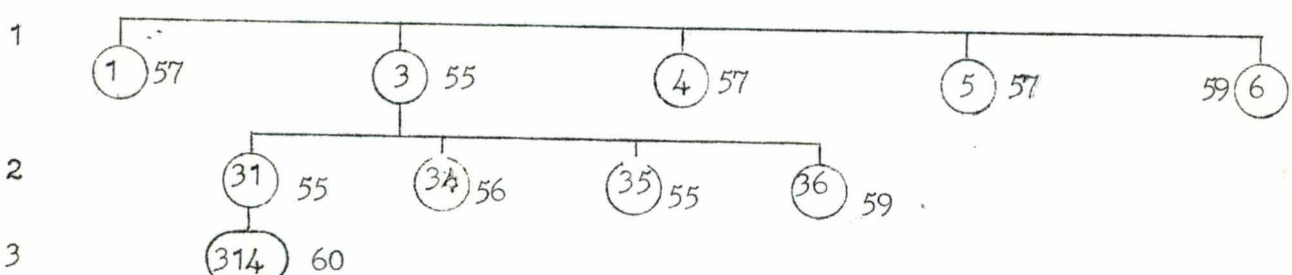
i	A	B	C	g'	g''	g'''	g
31562	30	41	54	54+7=61	41+11+7=59	30+ 3+18=51	61
31564	21	50	58	58+3=61	50+ 2+3=59	21+12+ 5=38	61

Nous venons de trouver deux ordonnancements qui étaient meilleurs que le précédent : 315624 et 315642 dont la valeur (g_2) est 61. Nous éliminons :

- tous les sommets dont la valeur est inférieure ou égale à 61, notamment 3152.
- le sommet 3156 dont nous venons d'explorer toutes les possibilités.

Il nous reste donc à envisager les permutations suivantes :

ETAPE :



2) Amélioration de g_2 ($g_2 = 61$)

Toutes les permutations de l'étape 4 étant explorées ou éliminées, nous remontons à l'étape 3 où la seule permutation restante est 314.

permutation 3142

i	A	B	C
314	13	28	35
2	12 →	2 →	3
	25	30	38

permutation 3145

i	A	B	C
314	13	28	35
5	6 →	8 →	10
	19	36	46

permutation 3146

i	A	B	C
314	13	28	35
6	2 →	14 →	12
	15	42	54

Valeurs de g_i :

i	A	B	C	g'	g''
3142	25	30	38	$38+(43-8-3-7-3)=60$	$30+(48-6-7-116-2)+10=\underline{62}$
3145	19	36	46	$46+(43-8-3-7-10)=\underline{61}$	
3146	15	42	54	$54+(43-8-3-7-12)=\overline{\underline{67}}$	

Il est inutile de calculer les autres valeurs de g'' et g''' car par celles que nous avons, nous pouvons affimer que la valeur de F_{max} est supérieure ou égale à 61.

Puisqu'il ne reste plus de permutation dans la troisième étape, nous remontons vers la seconde, où nous pouvons éliminer 31 puique toutes les permutations qui en étaient issues ont été envisagées. Il reste donc 34, 35, et 36 ; commençons par 35 dont F_{max} est inférieur aux autres.

permutation 351

$$\begin{array}{c|ccccc}
i & A & & B & & C \\
\hline
35 & 10 & & 18 & & 28 \\
 & \downarrow & & \downarrow & & \downarrow \\
1 & 6 & \longrightarrow & 7 & \longrightarrow & 3 \\
 & 16 & & 25 & & 31
\end{array}$$

permutation 352

$$\begin{array}{c|ccccc}
i & A & & B & & C \\
\hline
35 & 10 & & 18 & & 28 \\
 & \downarrow & & \downarrow & & \downarrow \\
2 & 12 & \longrightarrow & 2 & \longrightarrow & 3 \\
 & 22 & & 24 & & 31
\end{array}$$

permutation 354

$$\begin{array}{c|ccccc}
i & A & & B & & C \\
\hline
35 & 10 & & 18 & & 28 \\
 & \downarrow & & \downarrow & & \downarrow \\
4 & 3 & \longrightarrow & 11 & \longrightarrow & 7 \\
 & 13 & & 29 & & 36
\end{array}$$

permutation 356

$$\begin{array}{c|ccccc}
i & A & & B & & C \\
\hline
35 & 10 & & 18 & & 28 \\
 & & & \downarrow & & \downarrow \\
6 & 2 & \longrightarrow & 14 & \longrightarrow & 12 \\
 & 12 & & 32 & & 44
\end{array}$$

Valeurs de g_i :

i	A	B	C	g'	g''
351	16	25	31	$31+(43-8-10-3)=53$	$25+(48-6-8-7)+3=\underline{55}$
352	22	24	31	$31+(43-8-10-3)=53$	$24+(48-6-8-2)+3=\underline{59}$
354	13	29	36	$36+(43-8-10-7)=54$	$29+(48-6-8-11)+3=\overline{\underline{55}}$
356	12	32	44	$44+(43-8-10-12)=\underline{57}$	$32+(48-6-8-14)+3=\overline{55}$

i	A	B	C	g'''	g
351	16	25	31	$16+(33-4-6-6)+5=38$	55
352	22	24	31	$22+(33-4-6-12)+10=43$	59
354	13	29	36	$13+(33-4-6-3)+5=38$	55
356	12	32	44	$12+(33-4-6-2)+5=38$	57

Aucune de ces permutations n'est à rejeter ; il nous faut donc continuer à les explorer ; commençons par 351.

∴ permutation 3512

i	A	B	C
351	16	25	31
	↓	↓	↓
2	12 →	2 →	3
	28	30	34

permutation 3514

i	A	B	C
351	16	25	31
	↓	↓	↓
4	3 →	11 →	7
	19	36	43

permutation 3516

i	A	B	C
351	16	25	31
	↓	↓	↓
6	2 →	14 →	12
	18	39	51

Valeurs de g_i :

i	A	B	C
3512	28	30	34
3514	19	36	43
3516	18	39	51

g'

$34+(43-8-10-3-3)=53$
$43+(43-8-10-3-7)=\underline{58}$
$51+(43-8-10-3-12)=\underline{61}$

g''

$30+(48-6-8-7-2)+7=\underline{62}$
$36+(48-6-8-7-11)+3=55$

i	A	B	C
3512	28	30	34
3514	19	36	43
3516	18	39	51

g'''

$19+(33-4-6-6-3)+5=38$

g

58

Les autres valeurs de g'' et g''' ne sont pas à envisager puisque F_{max} sera certainement supérieur à 61 ; mais il faut continuer à explorer 3514.

permutation 35142

i	A	B	C
3514	19	36	43
	↓	↓	↓
2	12 →	2 →	3
	31	38	46

permutation 35146

i	A	B	C
3514	19	36	43
	↓	↓	↓
6	2 →	14 →	12
	21	50	62

Valeurs de g_i :

Comme Fmax est certainement supérieur à la date au plus tôt à laquelle la permutation 35146 sera terminée sur la troisième machine, il est inutile de le calculer puisque cette date est de 62.

i	A	B	C	g'	g''
35142	31	38	46	$46+(43-8-10-3-7-3)=58$	$38+(48-6-8-7-11-2)+12=\underline{64}$: Eliminé ($>$ 61)

La permutation 351 n'aboutissant à aucun résultat intéressant, explorons la permutation 354

permutation 3541

i	A	B	C
354	13	29	36
	↓	↓	↓
1	6 →	7 →	3
	19	36	39

permutation 3542

i	A	B	C
354	13	29	36
	↓	↓	↓
2	12 →	2 →	3
	25	31	39

permutation 3546

i	A	B	C
354	13	29	36
	↓	↓	↓
6	2 →	14 →	12
	15	43	55

Valeurs de g_i :

i	A	B	C	g'	g''
3541	19	36	39	39+(43−8−10−7− 3)=54	36+(48−6−8−11−7)+3=$\underline{55}$
3542	25	31	39	39+(43−8−10−7− 3)=54	31+(48−6−8−11−2)+3=$\overline{\underline{55}}$
3546	15	43	55	55+(43−8−10−7−12)=$\underline{61}$: A éliminer puisque égal à 61	

i	A	B	C	g'''	g
3541	19	36	39	19+(33−4−6−3− 6)+ 5=38	55
3542	25	31	39	25+(33−4−6−3−12)+10=43	55
3546	15	43	55		

Les deux premières permutations doivent être retenues ; examinons 3541 :

permutation 35412

i	A	B	C
3541	19 ↓	36 ↓	39 ↓
2	12 →	2 →	3
	31	38	42

permutation 35416

i	A	B	C
3541	19 ↓	36 ↓	39 ↓
6	2 →	14 →	12
	21	50	62

La permutation 35416 est à éliminer pour la même raison que celle qui nous a mené à éliminer la permutation 35146. Les valeurs de g pour 35412 sont :

A	B	C	g'	g''	
31	38	42	42+(43−8−10−7−3−3−)=54	38+(48−6−8−11−7−2)+12=$\underline{64}$	à éliminer (> 61)

Puisque la permutation 3541 n'aboutit pas, essayons la permutation 3542 :

permutation 35421

i	A	B	C
3542	25 ↓	31 ↓	39 ↓
1	6 →	7 →	3
	31	38	42

permutation 35426

i	A	B	C
3542	25 ↓	31 ↓	39 ↓
6	2 →	14 →	12
	27	45	57

Valeurs de g_i :

i	A	B	C	g'
35421	31	38	42	42+(43−8−10−7−3−3)=53
35426	27	45	57	57+(43−8−10−7−3−12)=$\underline{60}$

i	A	B	C	g''	g'''	g
35421	31	36	42	38+(48−6−8−11−2− 7)+12=$\underline{64}$: à éliminer		
35426	27	45	57	45+(48−6−8−11−2−14)+ 3=55	27+(33−4−6−3−12−2)+10=43	60

Nous venons de trouver un ordonnancement meilleur que ceux associés à la valeur de g_2. Il s'agit de l'ordonnancement 354261 dont la valeur de F_{max}, g_3 vaut 60 : Nous pouvons éliminer :

- tous les sommets dont la valeur est égale ou supérieure à 60
- le sommet 351 et 354 dont toutes les possibilités ont été envisagées.

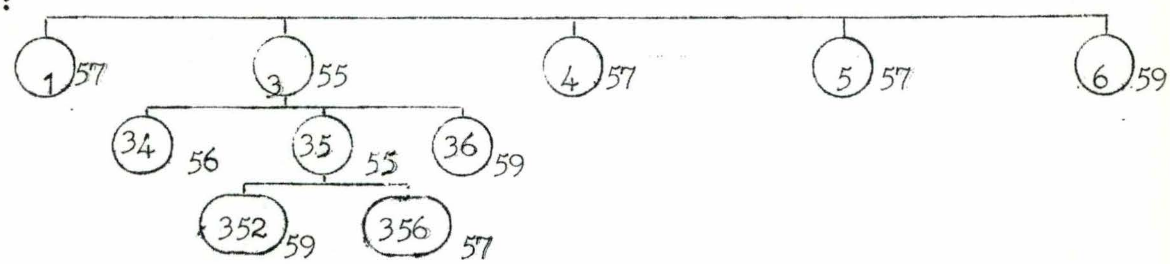
Il nous reste à envisager les sommets suivants :

ETAPE :

1

2

3



3. Amélioration de g_3 (g_3 = 60)

Il reste à explorer, dans la troisième étape les sommets 352 et 356 ; examinons les permutations issues de 356 :

permutation 3561

i	A	B	C
356	12	32	44
1	6	7	3
	18	39	47

permutation 3562

i	A	B	C
356	12	32	44
2	12	2	3
	24	34	47

permutation 3564

i	A	B	C
356	12	32	44
4	3	11	7
	15	13	51

Valeurs de g_i :

i	A	B	C	g'	g''
3561	18	39	47	47+(43− 8−10−12−3)=57	39+(48−6−8−14−7)+3=55
3562	24	34	47	47+(43− 8−10−12−3)=57	34+(48−6−8−14−2)+3=55
3564	15	43	41	41+(43− 8−10−12−7)=61 : à éliminer puisque supérieur à g_3.	

i	A	B	C	g'''	g
3561	18	39	47	18+(33−4−6−2− 6)+5=38	57
3562	24	34	47	24+(33−4−6−2−12)+10=43	57
3564	15	43	41		

Examinons les permutations issues de 3561

permutation 35612

i	A	B	C
3561	18	39	47
2	12	2	3
	30	41	50

permutation 35614

i	A	B	C
3561	18	39	47
4	3	11	7
	21	50	57

Valeurs de g_i

i	A	B	C	g'	g''	g'''	g
35612	30	41	50	50+7=57	41+11+7=59	30+3+18=51	59
35614	21	50	57	57+3=60: à éliminer puisque nous ne cherchons qu'un ordonnancement optimal			

Un meilleur ordonnancement vient d'être trouvé : 356124 dont la valeur de $F_{max}=59=g_4$.

4. <u>Amélioration de g_4</u> (g_4 = 59)

Il existe encore une permutation de la quatrième étape à explorer : 3562.

permutation 35621

i	A	B	C
3562	24	34	47
1	6 →	7 →	3
	30	41	50

permutation 35624

i	A	B	C
3562	24	34	47
4	3 →	11 →	7
	27	45	54

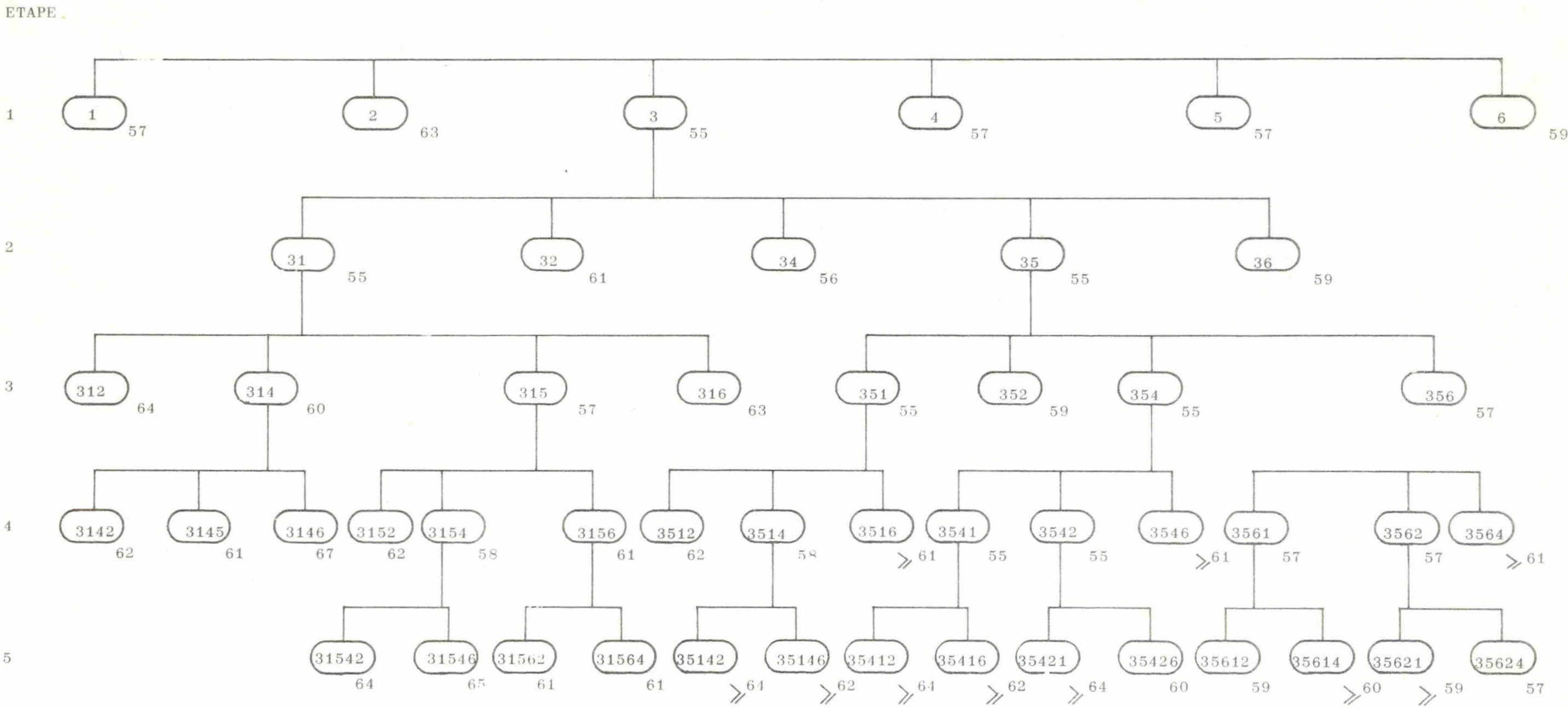
Valeurs de g_i :

i	A	B	C	g'	g''	g'''	g
35621	30	41	50	50+7=57	41+11+7=59 : à éliminer puisqu'égal à 59		
35624	27	45	54	54+3=57	45+ 7+3=55	27+6+10=43	57

Un meilleur ordonnancement vient d'être trouvé : 356241 dont F_{max}=57

Cet ordonnancement est optimal car tous les autres sommets ont une valeur supérieure ou égale à 57 sauf ceux qui se trouvent sur le chemin aboutissant à cet ordonnancement.

La technique apparaît lourde mais son caractère répétitif lui permet d'être facilement programmée. Au lieu des 720 ordonnancements possibles (6!) nous en avons examiné 48, dont seuls 14 étaient complets (contenaient l'ensemble de 6 tâches). Pour visualiser le problème, il y a un avantage certain à construire une arborescence dont chaque étape se caractérise par un nombre différent de tâches ordonnancées, et dont chaque sommet représente une permutation ; cet arborescence est particulièrement utile lorsqu'il s'agit d'éliminer certains sommets. Nous donnons ici l'arborescence complète des solutions examinées et la valeur de F_{max} qui leur est associée.



ETAPE
1
2
3
4
5
1 57
2 63
3 55
4 57
5 57
6 59
31 55
32 61
34 56
35 55
36 59
312 64
314 60
315 57
316 63
351 55
352 59
354 55
356 57
3142 62
3145 61
3146 67
3152 62
3154 58
3156 61
3512 62
3514 58
3516 ≥61
3541 55
3542 55
3546 ≥61
3561 57
3562 57
3564 ≥61
31542 64
31546 65
31562 61
31564 61
35142 ≥64
35146 ≥62
35412 ≥64
35416 ≥62
35421 ≥64
35426 60
35612 59
35614 ≥60
35621 ≥59
35624 57

ANNEXE V : Simulation du problème n/m/G/...

Section 1 : Exemple de génération d'ordonnancements actifs et non-retardés (Cfr. Titre II, chapitre 4)

Soit le problème $2/2/G/F_{max}$:

Tâche			
1	111	122	131
2	212		

§1 : Ordonnancement non-retardé

Au point de départ (moment 0), nous n'avons aucune opération en traitement. Il s'en suit :

$$\left\{S^1_{ip}\right\} = \left\{\text{vide}\right\} \quad ; \quad \left\{S^1_{so}\right\} = \left\{111\right\}$$

$$\left\{S^2_{ip}\right\} = \left\{\text{vide}\right\} \quad ; \quad \left\{S^2_{so}\right\} = \left\{212\right\}$$

Or $p_{111} = p_{\circ 11} = 2$; $p_{212} = p_{\circ 12} = 3$

D'où $C_1 = 0$, $C_2 = 0$, $s_{\circ 11} = 0$, $s_{\circ 12} = 0$

Appliquons la formule (cfr. Annexe III, section 6) :

$$\min_k \left\{ \min_{j \in \{S^k_{so}\}} \left[\max\left(C_k, s_{jk}\right) \right] \right\}$$

$$\min \left[\max(C_1, s_{\circ 11}) \; ; \; \max(C_2, s_{\circ 12}) \right] = 0$$

Puisque les deux opérations (111 et 212) ont un $s_{\circ jk}$ égal à cette valeur minimale, on peut positionner indistinctement une de ces deux opérations. Supposons que nous ordonnancions 111 ;

Au moment 1, nous aurons la situation suivante :

$\left\{S^1_{ip}\right\} = \left\{111\right\}$; $\left\{S^1_{so}\right\} = \left\{\text{vide}\right\}$ puisque l'opération susceptible d'entrer dans cet ensemble (131) n'a pas son prédecesseur en traitement (122).

$$\left\{S^2_{ip}\right\} = \left\{\text{vide}\right\} \quad ; \quad \left\{S^2_{so}\right\} = \left\{122 \; , \; 212\right\}$$

Comme $p_{122}=p_{.22}=2$ et $p_{212}=p_{.12}=3$,

$C_2=0$, $s_{.22}=2$ et $s_{.12}=0$

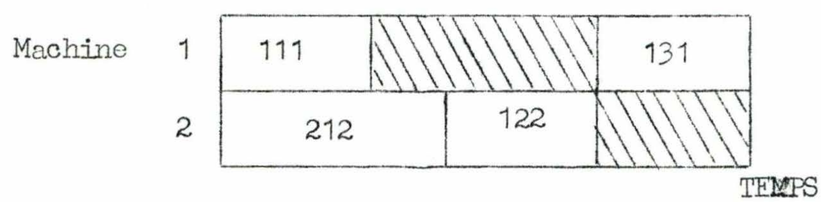
L'application de la formule donne :

$$\min\left[\max(0,2);\max(0,0)\right] = 0$$

La seule opération dont la valeur de $s_{.jk}$ est égale à ce montant est l' opération 212. Elle sera donc placée le plus à gauche possible c-a-d au moment 0. (Si nous avions ordonnancé 212 en premier lieu, 111 aurait été sélectionné à ce stade-ci).

Il est inutile de poursuivre la méthode pour les moments 2,3 ... car il n'y a plus de conflits : il reste les opérations 131 et 122 et elles doivent être traitées par deux machines différentes.

On aboutit donc à un ordonnancement non-retardé (nous retrouvons celui du chapitre 4 du titre II) :



§2 : Ordonnancement actif

Au moment 0, les données sont identiques à celles du cas précédent. La formule applicable aux ordonnancements actifs (cfr. Annexe III, section 6) donne :

$$\min_{k}\left\{\min_{j\in\{S^k_{so}\}}\left[\max(C_k\ ,\ s_{.jk}) + p_{jk}\right]\right\}$$

$$= \min\left[\max(0,0)+2\ ;\ \max(0,0)+3\right] = 2$$

Puisque les deux opérations (111 et 212) ont un $s_{.jk}$ inférieur à cette valeur, on peut positionner une de ces deux opérations indistinctement. Supposons l'ordonnancement de 111.

Au moment 1, la situation est la suivante :

$\{S^1_{ip}\} = \{111\}$; $\{S^1_{so} = \text{vide}\}$

$\{S^2_{ip}\} = \{\text{vide}\}$; $\{S^2_{so} = 122,212\}$

Comme $p_{122}=p_{.22}=2$, $p_{212}=p_{.12}=3$

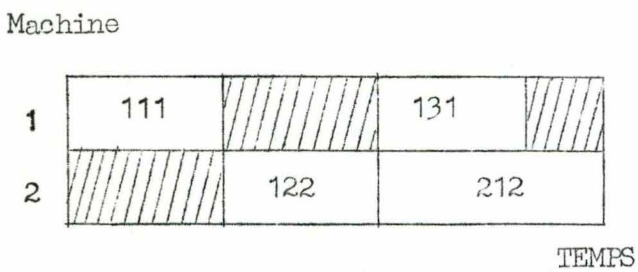
$C_2=0$, $s_{.22}=2$; $s_{.12}=0$

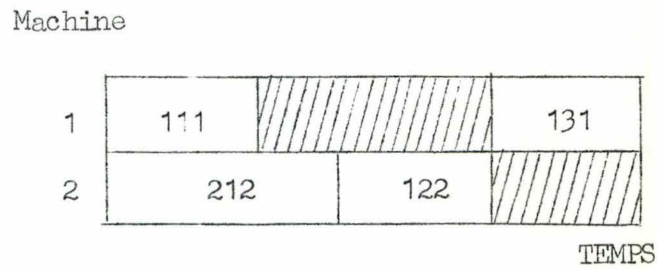
Le résultat de l'application de la formule est :

$$\min\left[\max(0,2)+2\ ;\ \max(0,0)+3\right] = 3$$

Les deux opérations 212 et 122 ont une valeur de $s_{.jk}$ inférieure à 3 ; elles sont toutes les deux susceptibles d'être ordonnancées.
(Si nous avions ordonnancé 212 en premier lieu, la seule opération contenue dans un $\{S^k_{so}\}$ non vide aurait été 111, puisque 122 ne pouvait être séléctionné, son prédecesseur n'étant pas en traitement).

Nous avons donc deux ordonnancements actifs (nous retrouvons ceux du Titre II, chapitre 4) :





Section 2 : Application du Branch and Bound au problème $n/m/G/F_{max}$ [1]

Dans l'arborescence des problèmes irrésolus générés par cette approche, chaque sommet représente :

- un moment du temps où plusieurs opérations sont ordonnançables sur la machine k.
- un ordonnancement actif des opérations appartenant à ce sommet.

Chaque branche issue de ce sommet représente la sélection de l'une des opérations en conflit. Le choix d'une branche particulière est fait sur base de la valeur minimale de la limite inférieure de F_{max} de chaque branche. L'efficacité de la procédure dépend essentiellement de la manière dont le branchement est effectué ainsi que de la qualité des limites inférieures. Brooks et White ont examiné trois possibilités de branchement ; nous n'en expliciterons que deux, la troisième étant trop longue et se rapprochant de l'application des règles de décision.

---

[1] BROOKS G.H. et WHITE C.R. : "An Algorithm for finding Optimal or Near-Optimal Solutions to the Production Scheduling Problem" J.Ind.Eng.,vol.16 n°1;janvier 1965

1. On cherche pour chaque tâche la date au plus tôt à laquelle l'opération suivante non ordonnancée pourrait commencer ; on y ajoute la durée de traitement de toutes les opérations non ordonnancées de cette tâche. La limite inférieure de F_{max} sera la valeur maximum trouvée.

Exemple : $3/4/G/F_{max}$

OPERATION :

	1		2		3		4	
	p_i	mach.	p_i	mach.	p_i	mach.	p_i	mach.
TACHE 1 :	6	1	8	4	9	3	4	2
2 :	1	1	3	2	9	3	6	4
3 :	5	1	5	3	3	2	6	3

i. Au moment t=0, les opérations 111,211 et 311 sont ordonnançables ; un conflit de passage existe car elles doivent toutes être traitées par la machine 1. Calculons la limite inférieure de F_{max} qui serait obtenue si une de ces trois opérations était choisie (on parcourt les trois branches). La branche retenue sera celle pour laquelle cette limite est minimisée.

• opération 111 : (Que se passe-t-il si elle est ordonnancée ?)

a. L'opération suivante de la tâche 1, 124 commencera au plus tôt au moment 6 ; si on y ajoute les durées de traitement des opérations restant à traiter, on obtient 6+(27-6)=27.

b. L'opération suivante de la tâche 2, 211 commencera au plus tôt (si 111 est ordonnancé) au moment 6 ; les opérations restant à traiter ont une durée totale de traitement de 19. D'où : 6+19=25.

c. Si on utilise la même argumentation pour la tâche 3, on obtient 6+19=25.

Si l'opération ordonnancée au moment 0 était 111, la limite inférieure de F_{max} s'élève à : max(27,25,25)=27.

Si nous opérons de la même manière pour les opérations 211 et 311 (Les opérations indiquées sont les branches possibles) :

• opération 211 :

a. 111 : 1+27 =28
b. 222 : 1+18 =19
c. 311 : 1+19 =20

La limite inférieure de F_{max} : max(28,19,20)=28

• opération 311

a. 111 : 5+27=32
b. 211 : 5+19=24
c. 323 : 5+14=19

La limite inférieure de F_{max} : max(32,24,19)=32

Au moment 0, l'opération 111 sera ordonnancée puisque des trois limites inférieures, elle réalise le minimum.(min(27,28,32))

ii. Au moment t=6 (aucune opération n'est ordonnançable au moment 1...5 à cause des contraintes de routing ou d'ordonnancement).

Comme il n'y a qu'une opération ordonnançable sur la machine 4 on peut la positionner. Subsiste un conflit sur la machine 1 entre les opérations 211 et 311. Les limites inférieures (calculées comme ci-dessus) donnent respectivement 26 et 35. On ordonnancera donc 211.

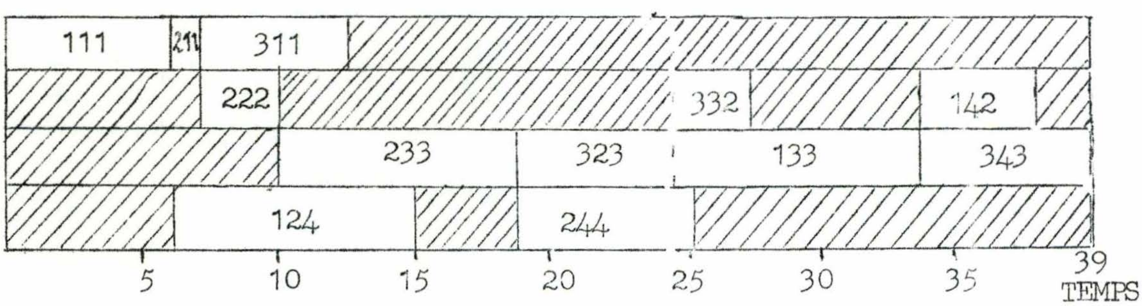
iii. Au moment t=7, les opérations 311 et 222 sont ordonnançables. Traitées par deux machines différentes, on peut les positionner.

iv. Au moment t=10, la seule opération ordonnançable est 233.

v. Au moment t=19, 244 peut être ordonnancé mais il existe un conflit sur la machine 3 entre les opérations 133 et 323 ; leurs limites inférieures respectives étant de 42 et 33, on ordonnance l'opération 323.

vi. Il n'y a plus de conflits pour les opérations restantes ; il suffit de les ordonnancer en respectant les contraintes de routing et d'ordonnancement en les "calant" à gauche.

L'ordonnancement complet se présente de la manière suivante ; la valeur de F_{max} s'élève à 39.



2. La seconde possibilité envisagée par Brooks et White consiste à s'occuper des machines et non plus des tâches ; on cherche la date au plus tôt à laquelle une opération non ordonnancée pourrait commencer et cela pour chaque machine ; on y ajoute la somme des durées de traitement des opérations non ordonnancées exigeant cette machine. La limite inférieure est le maximum de ces quantités.

   i. Au moment 0, les opérations ordonnançables sont 111,211,311.

      a. Opération 111 : calcul de la date au plus tôt

         - machine 1 : en traitant l'opération 111, qu'on suppose ordonnancée, cette machine ne pourra traiter une autre opération que lorsque 111 est achevé, c-a-d au moment 6.

         - machine 2 : elle doit traiter les opérations 222,332,142 ; or :

            . 222 ne pourra commencer que lorsque 211 est terminé mais 211 devra attendre la fin du traitement de 111 qu'on suppose ordonnancé ; 222 commencera au plus tôt au moment 6+1=7.

            . 332 ne pourra commencer que lorsque les opérations 111,311 et 323 sont achevées, c-a-d au plus tôt au moment 6+5+5=16.

            . 142 ne pourra commencer que lorsque 111,124 et 133 sont achevés, c-a-d au moment 6+8+9=23.

           La machine 2 commencera au plus tôt en min(7,16,23)=7 (si 111 est ordonnancé).

         - machine 3 : elle traite les opérations 323,233,133 et 343 :

            . 323 peut commencer lorsque 111 et 311 sont achevés, c-a-d au moment 6+5=11

            . 233 peut commencer lorsque 111,211,222 sont achevés, c-a-d au moment 6+1+3=10

            . 133 peut commencer lorsque 111 et 124 sont achevés, c-a-d au moment 6+8=14

            . 343 n'est pas à envisager puisque de toute façon, cette opération commencera plus tard que 323.

           La machine 3 commencera au plus tôt en min (11,10,14)= 10 (si 111 est ordonnancé).

         - machine 4 : les opérations que cette machine doit traiter (124 et 244) ne pourront pas commencer avant la moment 6 et 19. La machine 4 commencera donc au plus tôt au moment 6 (si 111 est ordonnancé).

Si l'opération 111 est ordonnancée, les opérations restant à traiter sur les machines 1 à 4 ont une durée totale de traitement de 6,10,29 et 10. La limite inférieure de F_{max} est le maximum de la date au plus tôt à laquelle une machine peut commencer plus la somme des durées de traitement des opérations que cette machine doit encore traiter c-a-d :

$max(6+6,\ 7+10,\ 10+29,\ 6+14) = 39$

Si on opère de la même façon dans le cas de l'ordonnancement au moment 0 des opérations 211, 311 :

b. opération 211

Les dates au plus tôt auxquelles les machines 1 à 4 pourront commencer (si 211 est ordonnancé) sont respectivement de 1,1,4,7. De plus ces machines seront encore occupées pendant 11,10,29 et 10 unités de temps. La limite inférieure de F_{max} s'élève à : $max(1+11, 1+10, 4+29, 7+10)=33$.

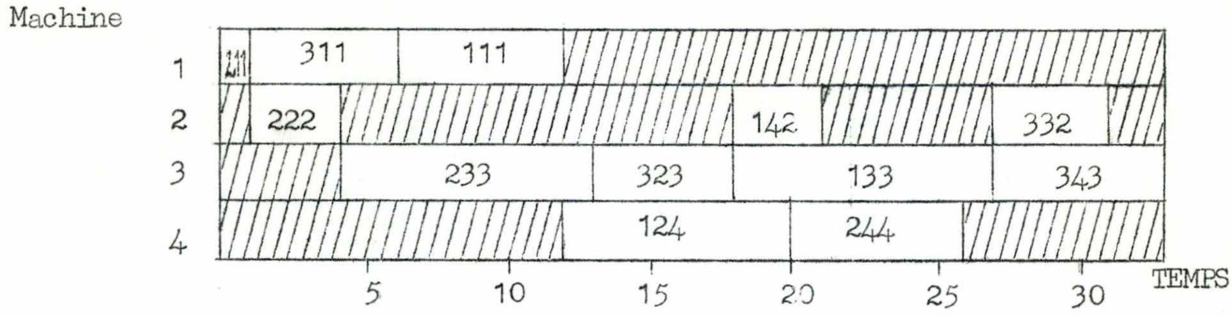
c. opération 311

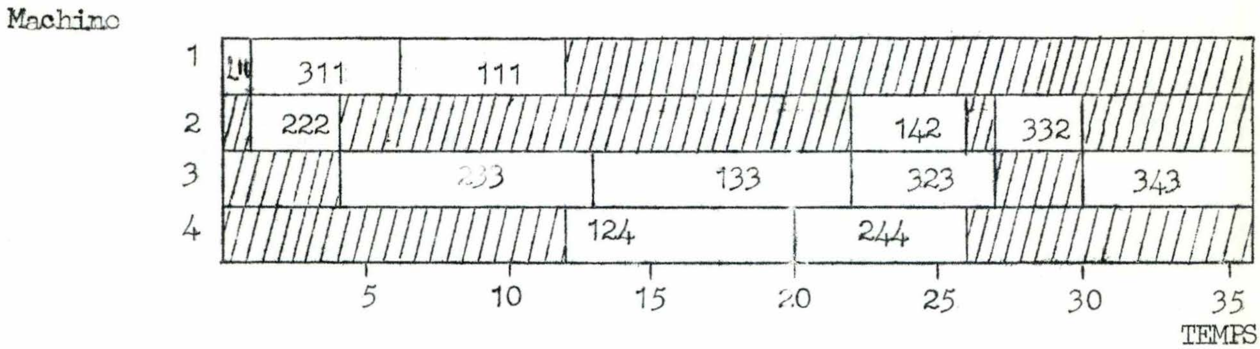
Les dates au plus tôt sont : 5,6,5,11
Les durées de traitement sont : 7,10,29,10
La limite inférieure de F_{max} : 34

Au moment 0, on ordonnancera l'opération dont la limite inférieure de F_{max} est minimum. Dans notre exemple, ce minimum est de 33 : on ordonnancera donc 211.

ii. Au moment 1, l'application de l'algorithme tranche entre les opérations 211 et 311 et choisit 311. On ordonnancera également l'opération 222 ; il en de même, au moment 4 pour l'opération 233 et au moment 6 pour l'opération 124 ; pour ces trois dernières operations, il n'y avait pas de conflit.

iii. Au moment 13, il existe un conflit sur la machine 3 entre les opérations 133 et 323 ; il apparait que les limites inférieures sont identiques ; on peut donc ordonnancer ces deux opérations indistinctement ; nous aurons donc deux ordonnancements (les autres opérations se placent facilement, car il n'y a plus de conflits). Remarquons cependant que les valeurs de F_{max} de ces ordonnancements ne sont pas identiques.





Nous pouvons tirer les conclusion suivantes de cet exemple :

- le premier algorithme est baucoup plus simple que le second ;
- aucun des deux ne garantit un ordonnancement optimal :
  . l'ordonnancement trouvé par le premier sera d'autant meilleur que le total des durées de traitement des opérations diffèrent d'une tâche à l'autre.
  . l'ordonnancement trouvé par le second sera d'autant meilleur que le total des opérations à traiter par une machine particulière diffère.

  On pouvait donc s'attendre à ce que le second algorithme donne un meilleur résultat car les charges des machines sont respectivement de 12,10,26,10, ce qui donne un rapport entre la charge la plus lourde et la plus faible de 2,17. Par contre les durées de traitement des tâches sont de 27,19,19 dont le rapport correspondant est 1,42.
- L'inconvénient d'un algorithme est l'avantage d'un autre. Le premier ne s'occupe pas de la charge des machines, ce que fait le second, mais celui-ci ignore les opérations de chaque tâche restant à traiter, ce dont le premier tient compte. Ceci est particulièrement évident dans notre exemple. Dans notre second algorithme les opérations 323 et 133 pouvaient être ordonnancées indistinctement : les deux ordonnancements complètés donnent deux valeurs de F_{max}. Cela provient de ce que l'algorithme était incapable de prendre les opérations traitées sur

la machine 2 en considération, parce que cette machine ne fournissait pas la valeur de F_{max} : il s'est occupé des containtes d'irdonnancement mais pas des contraintes de routing (ce dont le premier algorithme aurait tenu compte) Ces contraintes étaient manifestes dans le cas de l'opération 323 et 332. Il fallait d'autant plus tenir compte de cette contrainte que la dernière opération de la tâche 3 devait être traitée par la machine 3 (343) et ne pourrait commencer que lorsque 332 aurait été terminée.

Section 3: Application des règles de décision

Dans l'exemple précédent, les règles de décision forment un système de priorité relativement implicite. Dans le premier algorithme, une tâche dont la somme des durées de traitement des opérations restantes est la plus élevée est prioritaire, tandis que dans le second, la machine, dont la somme des durées de traitement des opérations restant à traiter (par cette machine) est prioritaire. Il n'empêche que la règle reste assez complexe. Il résulte d'un ensemble de travaux que les règles les plus simples peuvent donner d'aussi bons résultats.

1. Bakhru et Rao [1] ont comparé les ordonnancements actifs et non-retardés entre eux ; nous avons montré comment il était possible de les engendrer. Les deux auteurs ont simplifié le problème en supposant que chaque tâche contenait autant d'opérations que de machines et qu'une machine ne traitait qu'une seule opération. De chacun des 11 problèmes (de type 10/6) choisis, ils ont engendré un échantillon aléatoire de 50 ordonnancements actifs et de 50 ordonnancements non-retardés.

problème n°	ordonnancements : critère F				ordonnancements : critère F_{max}				
	actif		non-retardé		actifs		non-retardés		
	min.	et max.	min.	et max.	min	et max	min.	max.	limite inf.
1	61,9	74,9	59,1	69,4	76	106	71	103	71
2	61,5	75,4	61,3	70,3	83	115	78	99	67
3	62,8	81,2	69,2	73,8	86	119	82	106	73
4	51,7	63,6	49,0	59,8	68	99	64	91	64
5	56,0	74,7	54,1	67,9	74	109	70	93	56
6	67,7	82,8	65,2	75,6	95	122	85	116	71
7	51,8	65,4	51,0	62,1	69	112	66	83	59
8	57,4	72,3	55,0	67,5	76	108	73	95	67
9	60,0	75,5	58,9	69,6	82	119	75	113	65
10	52,6	62,4	51,3	61,0	67	100	62	88	54
11	61,4	77,3	61,0	68,6	84	116	86	105	78

---

[2] BAKHRU A.N. et RAO M.R. : "An Experimental Investigation of Job-Shop Scheduling" ; research report, Department of Industrial Engineering ; Cornell University 1964.

Ce tableau reprend pour chaque problème la plus petite et la plus grande valeur de $\bar{F}$ et de $\bar{F}_{max}$ des 50 ordonnancements actifs et non-retardés.

- Pour $\bar{F}$, la valeur minimum des ordonnancements non-retardés est inférieure, dans les 11 problèmes à celle des ordonnancements actifs. Quant au maximum, la valeur de $\bar{F}$ est inférieure dans 10 problèmes.

- Pour F_{max}, les conclusions sont identiques. Les auteurs ont ajouté la colonne "limite inférieure" : elle est obtenue en sommant les durées de traitement de chaque tâche et de chaque machine ; le maximum de ces n+m sommes constitue la limite inférieure de F_{max}. Les ordonnancements qui atteignent cette valeur sont certainement optimaux, ce qui ne signifie pas que les autres ne le sont pas. Comme la valeur de F_{max} du meilleur ordonnancement non-retardé est en moyenne de 12 % supérieur à la limite inférieure correspondante, tandis que le meilleur ordonnancement actif l'est de 19 %, les auteurs en concluent qu'il est plus efficient d'engendrer aléatoirement des ordonnancements non-retardés, bien que les actifs contiennent l'ordonnancement optimal, ce qui n'est pas nécessairement le cas des non-retardés.

2. Jeremiah, Lalchandani et Schrage [1] ont abouti aux mêmes conclusions, mais sans engendrer des ordonnancements de façon aléatoire. Parmi les opérations ordonnançables au moment t , ils opéraient une sélection en se basant sur certains attributs des opérations et/ou des tâches. Une fois une opération sélectionnée, on lui assigne sa date de début de traitement : cette méthode produit des ordonnancements actifs en choisissant une opération parmi celles contenues dans $\{S_{so}\}$.
Les règles de décision utilisées étaient : RANDOM,MOPNR,MWKR-P,MWKR/P,SPT, MWKR et LWKR. Ces ordonnancements furent comparés avec des ordonnancements non-retardés : une machine était d'abord sélectionnée et une opération était sélectionnée parmi celles contenues dans ce $\{S^k_{so}\}$ grâce aux règles de décision suivantes : RANDOM,SPT,MWKR,LPT,FCFS. Nous reproduisons ici une partie des résultats obtenues : 84 problèmes y sont envisagés ; ils se distinguent soit par le nombre de tâche et le nombre de machines, soit par le type de routing :

   - routing de permutation (Perm.) : chaque tâche contient exactement m opérations et chaque machine ne peut traiter qu'une opération d'une tâche.
   - routing aléatoire (Random) : chaque machine a la même probabilité d'être choisie pour traiter une opération quelconque d'une tâche. Le nombre d'opérations par tâche est une variable aléatoire dont la distribution (géométrique)a une espérance mathématique de m ; plusieurs opérations, non consécutives, peuvent donc être traitées par la même machine.

---

[1] JEREMIAH B., LALCHANDANI A. et SCHRAGE L. : "Heuristiv Rules Toward Optimal Scheduling" ; Research Report, Department of Industrial Engineering ; Cornell University 1964.

- routing intermédiaire (Fix R) : chaque tâche contient exactement m opérations mais il n'est pas nécessaire que chaque machine n'aie qu'une opération à traiter.

Les critères sont $\overline{F}$ et $\overline{F}_{max}$; les ordonnancements sont de type actif (A) ou non-retardé (N).

Chaque problème contient un certain nombre de sous-problèmes qui varient d'après la durée de traitement des opérations.

problème n°	n	m	routine	sous-problème n°	type cir.	DUREE MOYENNE D'OCCUPATION DE L'ATELIER RANDOM	MOPNR	MWKR-P	MWKR/P	SPT	MWKR	LWKR	LPT	FCFS	DUREE MAXIMALE D'OCCUPATION DE L'ATELIER RANDOM	MOPNR	MWKR-P	MWKR/P	SPT	MWKR	LWKR	LPT	FCFS	problème n°
1	10	1	Random.	1	A	29	32	33	33	29,9	32,4				67	52	52	54	66	52				1
					N					16,3	26,2	17,1	11,6	15,2					52	52	59	58	60	
2				2	A	29,1	33,7	33,8	33,8	29,7	33,1				92	63	63	63	83	63				2
					N					22,3	29,3	18,9	21,5	22					80	70	80	75	75	
3				3	A	59,2	72	78,6	71,6	61,6	75,1				118	96	94	104	118	93				3
					N					47,2	42,7	42,8	54	46,1					113	98	125	122	123	
4				4	A	46,2	54,8	53,4	50	49	52,7				88	92	76	81	103	77				4
					N					36,9	44,8	35,9	42,5	40,5					84	78	91	81	79	
5				5	A	68,5	94,5	88,8	86,1	70,4	85,6				156	136	136	136	199	136				5
					N					44	61,4	39,1	59,6	50,1					186	167	192	173	186	
6				6	A	68,2	89,2	85,4	82,5	68,7	81				155	107	106	111	155	101				6
					N					53,7	73,1	53,8	65,8	67,5					111	99	113	109	115	
7				7	A	48,4	64,9	62,5	59,2	52,2	61,7				116	86	92	94	123	92				7
					N					37,1	44,9	37	48,6	40,2					118	113	116	120	119	
8				8	A	53,5	69	61,9	69	57,9	66,6				190	117	113	124	180	115				8
					N					39	42,4	35,1	48	46,7					158	109	150	140	141	
9				9	A	54,7	70,4	72,2	72,2	59,8	72,6				119	110	110	110	123	110				9
					N					37,1	52,5	37,3	43,7	38,1					116	107	134	135	120	
10				10	A	43,2	51,7	58	57,5	43,1	57,6				97	81	81	87	91	78				10
					N					37,9	46,7	32,7	39,5	39,1					89	83	103	93	100	
11				11	A	28,5	29,2	31	30,1	25,7	34,2				53	53	53	53	54	53				11
					N					24,1	33,2	18,7	28,9	26,1					53	53	64	58	53	
12				12	A	40,4	42,9	43,5	46,4	36,9	42,2				78	62	62	66	88	62				12
					N					24,8	36,7	25	31,6	30,4					88	67	81	74	79	
13				13	A	68,1	91,5	94,1	102,7	75,9	96,2				190	138	122	129	154	124				13
					N					43,9	70,6	49	63,8	46,4					118	111	172	138	128	
14				14	A	40,8	48	44,4	50,3	43,8	49,8				81	78	77	81	88	77				14
					N					31	47,5	39	48	31.7					73	73	91	80	78	
15	20	4	Random	1	A	54,3	64,9	70,6	64,3	47,5	71,2				120	110	110	111	128	110				15
16				2	A	53,5	75,8	77,8	73,2	54,3	74,5				136	113	115	116	126	109				16
17				3	A	91	128,4	126,3	129	89,1	131				185	62	163	170	173	163				17
18				4	A	71,4	103,2	101,8	95,9	61,3	103				162	132	132	128	170	129				18
19				5	A	74,7	97,5	104,1	97,5	65,7	105,5				164	139	139	143	205	138				19
20				6	A	67,4	92,9	88,9	99,5	68,3	96,7				156	127	126	126	132	124				20
21				7	A	87,9	121,6	119,1	119,9	83,9	124				225	167	168	169	202	167				21
22				8	A	51,1	75,9	78,6	82,5	54,5	79				145	111	113	117	132	113				22
23				9	A	109,8	146,4	151,4	156,7	102,3	150,4				252	196	198	213	230	201				23
24				10	A	44,4	54,7	52,9	48,6	43,9	54				102	86	84	78	97	79				24
25	20	6	Random	1	A	100	120,2	125,1	118,3	94,6	126,8				192	177	178	182	220	184				25
26				2	A	95,5	118,8	111,6	109,8	88,1	113,1				184	149	146	147	208	144				26
27				3	A	109,3	125,3	132,1	126,8	101,2	126,4				226	176	185	188	248	174				27
28				4	A	114,5	147,5	152,7	148	117,1	150,5				216	186	185	198	261	186				28
29				5	A	82,5	90,4	87	89,7	74,6	90,6				267	151	155	157	199	142				29
30				6	A	106,2	133,5	139,1	133,2	115,1	136,4				237	187	191	191	249	186				30
31				7	A	121,7	153,8	155,2	157,8	116,4	159,2				247	187	187	193	238	194				31
32				8	A	67,3	94,2	95,6	88,9	71,3	94,5				168	121	116	125	146	119				32
33				9	A	103,7	146,7	148,1	145,2	109,3	148,2				249	181	187	185	284	187				33
34				10	A	107,4	139,5	133,8	137,7	113,4	135,5				250	185	177	188	245	178				34

problème n° :	n	m	routing	sous-problème n° :	type ord.	DUREE MOYENNE D'OCCUPATION DE L'ATELIER									DUREE MAXIMALE D'OCCUPATION DE L'ATELIER									problème n° :
						RANDOM	MOPNR	MWKR-P	MWKR/P	SPT	MWKR	LWKR	LPT	FCFS	RANDOM	MOPNR	MWKR-P	MWKR/P	SPT	MWKR	LWKR	LPT	FCFS	
35	10	4	Fix R	1	A					133,2	146,2	123,5							206	164	222			35
36				2	A					116,5	187	110,1							253	219	279			36
37				3	A					158	170,9	150,5							219	205	263			37
38				4	A					131,7	113,1	128,9							247	164	273			38
39				5	A					123,2	115,8	110,2							248	195	229			39
40				6	A					121,4	152	130,7							214	193	225			40
41				7	A					118	152,4	126,3							234	193	222			41
42				8	A					150,7	201,6	134,7							373	243	333			42
43				9	A					159,8	179,8	151,3							300	210	263			43
44				10	A					150,1	181,7	129,4							258	248	239			44
45	20	6	Fix-R	1	A	138,3	130,9	132	142	133,2	156				199	218	183	233	206	249				45
46				2	A	165,2	168,4	172,8	176,5	146,5	187,8				219	204	205	217	253	219				46
47				3	A	163,1	161,5	174,8	164,9	160,9	195,3				247	214	229	231	280	241				47
48				4	A	149,6	153,5	152	163,7	145,2	157,4				247	189	177	186	305	180				48
49				5	A	174,4	146	164,7	170,5	163,8	181,9				253	242	253	248	270	247				49
50				6	A	207,8	203,7	221,8	201,3	168,2	230				339	259	258	264	352	292	....	....	....	50
51				7	A	162	162	150,5	146	132,7	171,7				228	207	194	198	299	200				51
52				8	A	148,3	151,8	153,5	153,4	139,4	149,1				225	162	177	172	210	167				52
53				9	A	163,8	168,2	161,9	172	142,2	176,7				251	198	195	233	259	201				53
54				10	A	117,3	126	125,9	120,3	113,3	135,9				197	163	168	167	245	171				54
55				11	A	176	202	193,3	182,3	171,7	202,4				254	235	235	246	327	241	....	....	....	55
56				12	A	213,3	197,2	224,9	214,9	187,2	231,9				310	279	315	287	334	298				56
57				13	A	180,7	189,1	189,6	173,1	165,9	172,7				310	251	271	247	292	236				57
58				14	A	131	124	140,1	129,7	110	163,2				226	214	236	220	214	225				58
59				15	A	191,7	211.2	203,9	209,4	192,8	211,4				324	255	251	268	277	251				59
60				16	A	139,3	142,3	147	143,9	125,5	137,8				219	178	182	175	255	165	....	....	....	60
61				17	A	162	179,7	185.2	154,4	140,6	172,6				298	219	212	231	331	194				61
62				18	A	174,9	194,5	182,4	181,8	162,4	177,4				223	227	204	241	266	196				62
63				19	A	141,9	161,6	177,9	164	132	171,7				209	201	212	218	278	205				63
64				20	A	140,5	142,2	138,2	138,1	106,3	132,8				189	165	169	187	194	154				64

Problème n° :	n	m	routing	sous-problème n° :	type ord.	DUREE MOYENNE L'OCCUPATION DE L'ATELIER RANDOM	MOPNR	MWKR-P	MWKR/P	SPT	MWKR	LWKR	LPT	FCFS	DUREE MAXIMALE D'OCCUPATION DE L'ATELIER RANDOM	MOPNR	MWKR-P	MWKR/P	SPT	MWKR	LWKR	LPT	FCFS	problème n° :
65	10	4	Perm	1	A					130,3									198					65
					N	113,3				80,3	101,8	101;3	125,4	115,3	189				161	168	219	238	162	
66				2	A					157,6								223						66
					N	158,1				119	174,6	123,6	161,1	160	254				228	231	226	255	238	
67				3	A					151,3										251				67
					N	162,7				129	170	121,4	184,9	169,2	280				270	251	270	274	251	
68				4	A					136,7							203							68
					N	138,1				113,1	146,8	131,6	135,6	152,2	218				183	168	234	233	187	
69				5	A					145,1										203				69
					N	134,5				121,9	142,1	129	152,9	166,8	206				177	196	204	258	228	
70				6	A					154,4										254				70
					N	165,7				129,1	195,3	109,3	191,1	167,6	266				291	241	265	277	271	
71				7	A					127,7							193							71
					N	135,6				105,2	158,9	99,5	155	159,6	195				207	202	229	214	202	
72				8	A					137,1						178								72
					N	139,1				102,4	138,4	103,3	127,6	138,4	216				167	172	232	204	177	
73				9	A					167,3										225				73
					N	139,4				118,9	167,5	120;6	176,2	159,9	225				225	225	239	236	225	
74				10	A					116,9						170								74
					N	132,2				97	134,3	102,8	151,9	124,3	192				204	178	207	216	170	
75				11	A					162,4							245							75
					N	141,6				140,4	170,9	129	178	168,6	264				262	241	269	268	264	
76				12	A					156							236							76
					N	160,9				127,5	189,8	135,4	182,6	172,5	223				223	248	225	241	237	
77				13	A					156,6										214				77
					N	151,7				132,1	170,6	159,8	172,2	170,8	238				206	214	275	251	214	
78				14	A					103,8						135								78
					N	101,8				98,2	120,6	103,1	129,1	122,7	145				148	139	158	182	132	
79				15	A	155,4												257						79
					N	174,1				137,3	189,2	120,5	179,8	170,9	310				241	268	267	293	290	
80				16	A					130,3								171						80
					N	127,7				113	158,8	111,9	158,5	127,1	171				186	199	213	199	197	
81				17	A					148,6												221		81
					N	148,2				112,6	158,5	120,4	164,5	160,2	227				209	202	310	240	221	
82				18	A					167,3						217								82
					N	145,3				113	169,1	126,8	169,6	170,1	230				231	226	272	230	216	
83				19	A					123,4								235						83
					N	174,1				121,7	176,7	125,6	171,2	146,9	287				232	235	255	254	244	
84				20	A					101,6								171						84
					N	97,1				96,9	130,9	85,8	136,9	110	194				144	150	188	181	167	

De toutes ces données on peut tirer les conclusions suivantes :

- quant à $\overline{F}$:

  - on insistera sur la supériorité des ordonnancements non-retardés : pour les 14 problèmes de type 10/4/Random/$\overline{F}$, le meilleur ordonnancement non-retardé est supérieur au meilleur ordonnancement actif (une valeur est soulignée lorsqu'elle est minimale quant à $\overline{F}$ ou à $\overline{F}_{max}$ pour les ordonnancements actifs ou non-retardés). Le contraste est particulièrement clair lorsqu'on compare les procédures SPT et MWKR qui sont utilisées dans les deux types d'ordonnancements. Les mêmes conclusions peuvent être tirées pour les problèmes 10/4/Perm/$\overline{F}$.
  - Pour les ordonnancements actifs, trois règles de décision sont dominantes : SPT, RANDOM et LWKR. Il n'y a que deux problèmes sur les 84, où le meilleur ordonnancement actif est trouvé par une règle différente : il n'est cependant pas possible d'effectuer un choix entre ces trois règles pour en retirer une règle dominante. Aucune conclusion ne peut être tirée quant à l'effet de la dimension du problème et du type routing sur l'efficacité de certaines règles de décision.

- quant à F_{max} :
  - Les conclusions divergent dans ce sens qu'il est impossible de trancher entre les ordonnancements actifs et non-retardés. Sur les 14 problèmes 10/4/RANDOM/F_{max}, 5 ordonnancements non-retardés sont meilleurs (bien que deux d'entre eux aboutissent au même résultat que dans l'ordonnancement actif correspondant).
  - Parmi les 20 problèmes 10/4/Perm/F_{max}, 14 ordonnancements non-retardés fournissent de meilleurs résultats (5 d'entre eux fournissent des résultats équivalents à ceux des ordonnancements actifs).
  - Des indications claires sont données quant à l'efficacité des règles : il semble extrêmement intéressant de considérer le total du travail restant à effectuer (MWKR) ; lorsque cette règle n'aboutit pas au meilleur résultat, il suffit d'utiliser une variante de cette règle (MWKR-P ou MWKR/P).

Section 4 : Règles de décision probabilistes

Jusqu'à présent, lorsqu'on utilisait une règle de décision, un seul ordonnancement en résultait ; si nous répétions l'application de la règle au même problème, nous aboutirions au même résultat, car les opérations gardent la même priorité. Certains chercheurs ont essayé d'éviter une application aussi automatique de ces règles en joignant à chaque système de priorité une distribution de probabilité.

Exemple : Supposons qu'au moment t , l'ensemble $\{S_{SO}\}$ se compose d'opérations dont les durées de traitement sont 4,7,3,9. Si nous utilisons la règle SPT comme nous l'avons fait jusqu'à présent, la 3ème opération serait prioritaire, et le resterait quelque soit le nombre de répétitions du problème. Si maintenant, nous attribuons à la priorité 1, la probabilité 0,6 à la priorité 2 la probabilité 0, 2 ... , il n'est plus certain, que la troisième opération sera toujours ordonnancée. Une répétition de la règle probabiliste à un même problème est possible sans qu'il en résulte le même ordonnancement. Remarquons que nous donnons à la priorité 1 la plus grande probabilité, ce qui donne à la tâche sélectionnée par la règle de décision la plus grande chance d'être sélectionnée. Dans le tableau suivant, nous reproduisons quelques résultats dûs à Jeremiah, Lalchandani et Schrage.
Le nombre d'application de la même règle au même problème est de 50 et donne 50 ordonnancements ; si, par contre la règle était strictement utilisée, nous n'en aurions qu'un.

* Pour les ordonnancements actifs, nous reprenons les 10 problèmes 10/4/Fix R//.
Le tableau reprend, pour $\bar{F}$ et $\bar{F}_{max}$, le résultat obtenu par l'application stricte de la règle ainsi que la valeur minimale trouvée par application de la règle probabiliste.

	DUREE MOYENNE D'OCCUPATION DE L'ATELIER							DUREE MAXIMALE D'OCCUPATION						
	SPT		MWKR		LWKR		RANDOM	SPT		MWKR		LWKR		RANDOM
prob.	pur	min.	pur	min.	pur	min.	min.	pur	min	pur	min	pur	min	min
1	133,2	123,4	146,2	133,3	123,5	109,0	124,0	206	171	164	160	222	201	188
2	146,5	136,4	187,0	171,8	140,1	131,6	152,4	253	221	219	213	279	237	214
3	158,0	141,0	170,9	161,3	150,5	130,1	142,8	219	206	205	196	263	224	235
4	131,7	121,8	143,4	139,8	128,9	[illegible]	131,0	247	228	164	164	273	207	180
5	123,2	110,8	145,8	138,6	110,2	[illegible]	115,7	148	196	195	195	229	209	191
6	121,4	112,6	152,0	139,4	130,7	110,4	117,0	214	173	193	170	225	187	170
7	148,0	125,3	152,4	144,9	126,3	120,1	124,3	234	188	193	185	222	197	191
8	150,7	132,7	204,6	186,5	134,7	125,1	144,4	373	288	243	237	333	274	249
9	159,8	146,9	179,8	175,7	151,3	141,6	153,9	300	227	210	204	263	231	218
10	150,1	131,1	181,7	153,7	129,4	122,8	144,1	258	207	248	216	239	234	210

* Pour les ordonnancements non-retardés, le critère est F_{max} ; le nombre d'ordonnancements engendrés par la règle probabiliste varie de 40 à 1000. Lorsque le nombre 1 est indiqué, cela suppose que la règle a été strictement appliquée (un seul ordonnancement en résulte).

problème	n	m	routing	nombre d' ordonnancement	RANDOM	SPT	MWKR	FCFS
85	20	9	random	1		20,7	28,3	22,5
				100	21,2	19,9	21,2	
86				1		15,0	19,0	16,8
				100	15,2	14,1	15,2	
87				1		24,3	29,5	25,8
				100	24,3	22,2	24,3	
88				1		23,1	30,3	25,0
				100	23,3	22,7	24,3	
89	60	9	random	1		32,5	53,3	32,8
				50	30,8	29,7	30,8	
90				1		30,9	55,5	33,8
				50	33,0	29,7	33,0	
91	100	9	random	1		45,7	93,4	58,0
				40	56,0	44,8	56,0	
92				1		47,8	90,9	54,3
				40		46,9	73,2	
93	6	6	perm.	1		52,7	55,8	54,7
				100	48,2	46,7	48,2	
94	10	10	perm.	1		834	101	1036
				100	875	791	875	
95	10			1		68,9	73,5	74,5
				100	68,0	66,8	68,0	
				1000		66,4		

Il ressort clairement de ces tableaux que l'application de règles probabilistes est <u>dans tous les cas</u> meilleure que l'application stricte d'une règle. Il faut également noter une systématicité : la règle déterministe qui donne le meilleur ordonnancement le donne aussi lorsque cette règle est utilisée de façon probabiliste.

Nugent [1] s'est intéressé à ces problèmes mais il conclut d'un ensemble important de recherches que l'amélioration apportée par les règles probabilistes est trop faible pour mériter les efforts à fournir pour y parvenir. Il insiste néanmoins sur la valeur de ces règles, en signalant que ses ordonnancements étaient parfois meilleurs que ceux obtenus par la programmation linéaire. Cependant la procédure est trop lourde et il apparait que l'amélioration devient d'autant plus faible que le problème est de plus grande dimension.

---

[1] NUGENT C.E. : "On Sampling Approaches to the Solution of the n by m Static Sequencing Problem" ; Ph.-D.Thesis ; Cornell University septembre 1964.

ANNEXE VI - LES 42 ORDONNANCEMENTS OBTENUS

Dans les tableaux qui suivent, nous reprenons les 42 ordonnancements différents que nous avons obtenus en appliquant les règles heuristiques Au lieu de tracer 42 diagrammes-machine, nous avons préféré indiquer la date à laquelle chaque opération commençait. Les valeurs des critères sont repris à la suite de la description de ces ordonnancements. Nous concluons par quelques remarques.

Les données du problème (voir Titre III, chapitre 5) sont les suivantes :

- le premier chiffre indique le numéro d'identification de la tâche
- le deuxième indique le numéro d'ordre de l'opération
- le troisième indique le numéro d'identification de la machine
- les deux derniers, la durée de traitement de l'opération.

(Il s'agit d'un problème de type 6/6/G/...)

Tâche 1	Tâche 2	Tâche 3	Tâche 4	Tâche 5	Tâche 6
11301	21208	31305	41205	51309	61203
12103	22305	32404	42105	52203	62403
13206	23510	33608	43305	53505	63609
14407	24610	34109	44403	54604	54110
15603	25110	35201	45508	55103	65504
16506	26404	36507	46609	56401	66301

date de livraison :

56	56	50	55	57	61

Remarques :

- Tous les ordonnancements sont de type non-retardés, excepté ceux qui utilisent l'heuristique de projection (ils sont transformés en ordonnancements actifs)
- toutes les tâches sont disponibles au moment 0.

DATE A LAQUELLE UNE OPERATION DOIT COMMENCER

ORDONNANCEMENT N°

Opération	1	2	3	4	5	6	7	8	9	10	11	12
113	0	0	0	0	24	24	5	9	9	5	0	0
121	1	1	1	1	25	25	6	13	13	6	1	1
132	8	8	16	16	28	28	16	19	19	16	19	19
144	14	14	22	22	34	34	22	25	25	22	25	25
156	21	21	29	29	63	50	29	32	32	29	32	32
165	30	30	42	50	66	53	50	36	49	47	46	49
212	0	0	0	0	0	0	0	8	8	0	8	8
223	15	15	15	15	9	9	15	20	20	15	20	20
235	20	20	20	20	14	14	20	49	32	20	32	32
246	37	37	32	32	31	31	32	59	44	32	44	44
251	47	47	42	42	41	41	42	69	54	42	54	54
264	57	57	52	52	51	51	52	79	64	52	64	64
313	1	1	1	1	14	14	0	10	10	0	1	1
324	6	6	6	6	19	19	5	15	15	5	6	6
336	10	10	10	10	23	23	9	19	19	9	15	15
341	18	18	18	18	31	31	18	27	27	18	25	25
352	27	27	27	27	40	40	27	36	36	27	34	34
365	36	36	35	43	41	41	43	42	42	53	52	42
412	14	14	8	8	8	8	8	3	3	8	3	3
421	27	27	13	13	13	13	13	8	8	13	8	8
433	32	32	20	20	19	19	20	15	15	20	15	15
444	37	37	29	29	26	26	29	20	20	29	20	20
455	43	48	48	35	29	29	35	24	24	35	24	24
466	51	56	56	46	41	57	46	35	35	43	35	35
513	6	6	6	6	0	0	6	0	0	6	6	6
522	19	19	22	22	13	13	22	16	16	22	16	16
535	51	43	30	30	24	24	30	19	19	30	19	19
546	60	48	42	42	59	53	42	27	27	42	24	24
551	64	57	52	52	69	61	52	36	36	52	34	34
564	67	61	56	56	72	64	56	39	39	56	37	37
612	22	22	13	13	16	16	13	0	0	13	0	0
624	25	25	16	16	23	23	16	3	3	16	3	3
636	28	28	19	19	50	41	19	6	6	19	6	6
641	37	37	28	28	59	51	28	16	16	28	15	15
655	56	56	56	56	72	61	56	32	55	43	42	55
663	60	60	60	60	76	65	60	36	59	47	46	59
S'applique pour les règles et heuristiques :	1R	1A	2R	2A	3R	3A	4R	5R	5A	6R	7R	7A
(A = Alternance	1P	1AP	1A'		3P	3AP	4A					15A
P = Projection + Insertion							6A					
R = sans heuristique							12R					
AP = toutes les heuristiques							12A					
A' = Alternance élargie)							13R					
							13A					

DATE A LAQUELLE UNE OPERATION DOIT COMMENCER

ORDONNANCEMENT N°

Opération	13	14	15	16	17	18	19	20	21	22	23	24
113	0	0	0	5	5	0	9	0	0	5	5	5
121	1	1	1	10	6	1	13	1	1	6	10	6
132	19	19	19	19	19	16	19	19	19	13	13	16
144	25	25	25	25	25	22	25	25	25	19	19	22
156	32	32	41	41	41	40	41	41	44	26	26	40
165	53	49	49	49	49	50	55	51	50	50	48	50
212	0	0	8	8	8	0	8	8	0	0	5	0
223	15	15	20	20	20	15	16	16	15	15	13	15
235	24	24	32	32	32	20	21	21	20	20	18	20
246	35	35	44	44	44	30	31	31	30	30	29	30
251	45	45	54	54	54	40	41	41	40	40	39	40
264	55	55	64	64	64	50	51	51	50	50	49	50
313	1	1	1	0	0	1	10	1	1	0	0	0
324	6	6	6	5	6	6	15	6	6	5	5	5
336	10	10	10	9	15	10	19	15	10	9	9	9
341	18	18	18	17	25	18	27	25	18	18	17	18
352	27	27	27	26	34	27	36	34	27	27	26	27
365	34	34	42	42	42	43	48	44	43	43	28	30
412	8	8	0	0	3	8	3	3	8	8	0	8
421	13	13	5	5	9	13	8	8	13	13	5	13
433	20	20	15	15	15	20	21	21	20	20	18	20
444	32	32	20	20	20	29	32	32	32	26	26	29
455	41	41	25	24	24	35	35	36	35	30	35	37
466	49	49	32	32	32	47	44	48	47	53	52	45
513	6	6	6	6	6	6	0	6	6	6	23	6
522	16	16	16	16	16	22	16	16	16	22	32	22
535	19	19	19	19	19	30	43	31	30	38	43	45
546	28	28	27	26	24	43	53	44	40	49	48	54
551	28	38	37	36	34	50	57	51	50	60	59	58
564	41	41	40	39	37	54	60	55	54	63	62	61
612	13	13	5	5	0	13	0	0	13	19	19	13
624	16	16	10	9	3	16	3	3	16	29	29	16
636	19	19	18	17	6	19	6	6	19	40	39	19
641	28	28	27	26	15	28	16	15	28	49	49	28
655	49	55	55	55	55	56	31	57	56	60	59	56
663	53	59	59	59	59	60	35	61	60	64	63	60
S'applique pour les règles et heuristiques :	8R	8A 10A'	2A' 4A' 6A' 8A' 9A' 11A' 12A'	3A'	5A' 7A'	2AP 4AP 6AP	5AP	7AP	8AP	9A 9AP	10AP	11P 11AP

(A = Alternance
P = Projection + Insertion
R : sans heuristique
AP : toutes les heuristiques
A' : Alternance élargie

DATE A LAQUELLE UNE OPERATION DOIT COMMENCER

ORDONNANCEMENT N°

Opérations	25	26	27	28	29	30	31	32	33	34	35	36
113	5	5	0	0	0	5	9	5	0	0	5	5
121	6	6	1	1	1	6	13	6	1	1	6	10
132	16	16	19	19	16	16	19	16	19	19	13	13
144	22	22	25	25	22	22	25	22	25	25	19	19
156	44	40	40	41	40	40	45	40	41	44	26	26
165	50	50	50	55	54	50	55	47	48	47	45	43
212	0	0	0	8	0	0	8	0	8	0	0	5
223	15	15	15	16	15	15	16	15	16	15	15	13
235	20	20	20	21	20	20	21	20	21	20	20	18
246	30	30	30	31	30	30	31	30	31	30	30	29
251	40	40	40	41	40	40	41	40	41	40	40	39
264	50	50	50	51	50	50	51	50	51	50	50	49
313	0	0	1	1	1	0	10	0	1	1	0	0
324	5	5	6	6	6	5	15	5	6	6	5	5
336	9	9	10	15	10	9	19	9	15	10	9	9
341	18	18	21	25	18	18	27	18	25	18	18	17
352	27	27	30	34	27	27	36	27	34	27	27	26
365	43	43	43	48	35	43	40	53	54	53	38	28
412	8	8	11	3	8	8	3	8	3	8	8	0
421	13	13	16	8	13	13	8	13	8	13	13	5
433	20	20	21	21	20	20	21	20	21	20	20	18
444	29	29	32	32	29	29	32	29	32	32	26	26
455	35	35	35	35	42	35	47	35	36	35	30	35
466	47	47	47	44	50	43	55	47	48	47	40	48
513	6	6	6	6	6	6	0	6	6	6	6	23
522	22	22	16	16	22	22	16	22	16	16	22	32
535	30	30	30	43	30	30	35	30	31	30	51	49
546	40	43	43	53	43	52	41	43	44	40	58	57
551	50	50	50	57	50	56	51	50	51	50	68	61
564	54	54	54	60	54	59	55	54	55	54	71	64
612	13	13	8	0	13	13	0	13	0	13	19	19
624	16	16	11	3	16	16	3	16	3	16	29	29
636	19	19	18	6	19	19	6	19	6	19	49	43
641	28	28	30	15	28	28	16	28	15	28	58	49
655	56	56	56	31	50	56	31	43	44	43	68	59
663	60	60	60	35	54	60	35	47	48	47	72	63
S'applique pour les règles et heuristiques	12P 12AP	13AP	14P 14AP	15AP	2P	4P 13P	5P	6P	7P	8P	9R 9P	10P

(A = Alternance
P = Projection + Insertion
R = sans heuristique
AP = toutes les heuristiques
A' : Alternance élargie)

DATE A LAQUELLE UNE OPERATION DOIT COMMENCER

ORDONNANCEMENT N°

Opération	37	38	39	40	41	42
113	0	5	5	5	0	0
121	1	10	10	6	1	1
132	8	13	13	16	19	8
144	14	19	19	22	25	14
156	23	26	26	29	32	23
165	46	42	50	50	49	41
212	14	5	5	0	0	14
223	22	20	15	15	15	22
235	27	48	20	20	24	54
246	37	58	30	32	35	64
25[illegible]	47	68	40	42	45	74
264	57	78	50	52	55	84
313	1	0	0	0	1	1
324	6	5	5	5	6	6
336	15	9	9	9	10	15
341	25	17	17	18	21	25
352	34	26	26	27	30	34
365	52	27	30	30	34	47
412	3	0	0	8	11	3
421	8	5	5	13	16	8
433	15	15	20	20	21	15
444	21	26	26	29	32	21
455	59	34	37	37	41	24
466	67	42	49	45	49	32
513	6	6	6	6	6	6
522	22	19	19	22	16	22
535	41	22	45	45	19	36
546	47	29	58	54	27	41
551	57	33	62	58	40	45
564	61	36	65	61	43	48
612	0	22	22	13	8	0
624	3	29	29	16	11	3
636	6	33	40	19	18	6
641	15	42	50	28	30	15
655	37	58	60	56	55	32
663	41	62	64	60	59	36
S'applique pour les règles et heuristiques (A = Alternance P = Projection + Insertion R = aucun heuristique AP : toutes les heuristiques A' = Alternance élargie)	15P	10R	10A	11R 11A	14R 14A	15R

VALEURS DES CRITERES

ORDONNANCEMENT N°

Critère		1	2	3	4	5	6	7	8	9	10	11	12
Achèvement de la tâche :	1	36	36	48	56	72	59	56	42	55	53	52	55
	2	61	61	56	56	55	55	56	83	68	46	68	68
	3	43	43	42	50	48	48	50	49	49	60	59	59
	4	60	65	65	55	50	66	55	44	44	55	44	44
	5	68	62	57	57	73	65	57	40	40	57	38	38
	6	61	61	61	61	77	66	61	37	60	48	47	60
F_{max}		68	65	65	61	77	66	61	83	68	60	68	68
$\bar{F}$		54,83	54,67	54,83	55,83	62,5	59,83	55,83	49,17	52,67	54,83	51,33	52,33
Avance (retard) de la tâche	1	20	20	8	0	-16	-3	0	14	1	3	4	1
	2	-5	-5	0	0	1	1	0	-27	-12	0	-12	-12
	3	7	7	8	0	2	2	0	1	1	-10	-9	1
	4	-5	-10	-10	0	5	-11	0	11	11	0	11	11
	5	-11	-5	0	0	-16	-8	0	17	17	0	19	19
	6	0	0	0	0	-16	-5	0	24	1	13	14	1
Avance (retard) min.max.		-11	-10	-10	0	-16	-11	0	-27	-12	-10	-12	-12
Avance (retard) moyenne		1	1,17	1	0	-6,67	-4	0	6,67	3,17	1	4,50	3,50
Somme des retards :		21	20	10	0	48	27	0	27	12	10	21	12
Somme des avances :		27	27	16	0	8	3	0	67	31	16	48	33
Attente totale de la tâche	1	10	10	22	30	46	33	30	16	30	27	26	30
	2	14	14	9	9	8	8	9	36	21	9	21	21
	3	9	10	8	16	14	14	16	15	16	26	25	16
	4	25	30	30	20	15	31	20	9	9	20	9	9
	5	43	37	32	32	48	40	32	15	15	32	13	13
	6	31	31	31	34	47	36	31	7	30	18	17	30
Total des attentes tâches :		132	132	132	138	178	162	138	98	121	132	111	119
Attente max. de la tâche	1	6	6	12	18	24	24	18	9	14	15	15	15
	2	7	7	7	7	7	7	7	24	8	7	8	8
	3	8	8	7	15	14	14	15	10	10	25	17	7
	4	14	14	16	8	8	20	8	3	3	8	3	3
	5	29	21	7	7	30	24	7	7	7	7	6	6
	6	22	22	18	18	24	16	18	6	29	13	17	30
Nombre max. de tâches en attente dont la machine	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	3	3	3	3	2	2	2	2	2	2	2	2
	3	2	2	2	2	3	3	2	2	2	2	2	2
	4	0	1	1	1	1	1	1	0	0	1	0	0
	5	3	3	3	3	1	1	3	2	3	3	3	3
	6	1	1	1	1	3	3	1	1	1	1	1	1
Inoccupation totale de la machine	1	27	20	15	15	32	24	15	39	24	15	24	24
	2	2	2	2	2	15	15	2	11	11	2	9	9
	3	35	35	35	35	51	40	35	11	34	22	21	34
	4	46	40	35	35	51	43	35	61	46	35	46	46
	5	20	20	20	20	36	25	20	19	19	20	19	19
	6	21	22	22	12	23	23	12	26	11	12	11	11
Total inoccupation machine		151	139	129	119	208	170	119	167	145	106	130	143
Inoccupation max de la machine	1	14	14	9	9	13	13	6	30	15	6	17	17
	2	2	2	2	2	9	9	2	11	11	2	9	9
	3	23	23	35	35	51	40	35	11	34	22	21	34
	4	17	17	20	20	19	19	20	39	24	20	26	26
	5	20	20	20	20	18	14	20	19	19	20	19	19
	6	10	10	10	10	23	23	9	15	6	9	6	6

VALEURS DES CRITERES

ORDONNANCEMENT N°

CRITERE		13	14	15	16	17	18	19	20	21	22	23	24
Achèvement de la tâche	1	59	55	55	55	55	56	61	57	56	56	54	56
	2	59	59	68	68	68	54	55	55	54	54	53	54
	3	41	41	49	49	49	50	55	51	50	50	35	37
	4	58	58	41	41	41	56	53	57	56	62	61	54
	5	42	42	41	40	38	55	61	56	55	64	63	63
	6	54	60	60	60	60	61	36	62	61	65	64	61
F_{max}		59	60	68	68	68	61	61	62	61	65	64	62
$\bar{F}$		52,17	52,50	52,33	52,17	51,83	55,33	53,50	56,33	55,33	58,50	55	54
Avance (retard) de la tâche :	1	-3	1	1	1	1	0	-5	-1	0	0	2	0
	2	-3	-3	-12	-12	-12	2	1	1	2	2	3	2
	3	9	9	1	1	1	0	-5	-1	0	0	15	13
	4	-3	-3	14	14	14	-1	2	-2	-1	-7	-6	1
	5	15	15	16	17	19	2	-4	1	2	-7	-6	-5
	6	7	1	1	1	1	0	25	-1	0	-4	-3	0
Avance (retard) min(max)		-3	-3	-12	-12	-12	-1	-5	-2	-1	-7	-6	-5
Avance (retard) moyenne		3,67	3,33	3,50	3,67	4	0,50	2,33	-0,50	0,50	-2,67	0,83	1,83
Somme de retards :		9	6	12	12	12	1	14	5	1	18	15	5
Somme des avandes		31	26	33	34	36	4	28	2	4	2	20	16
Attente totale de la tâche :	1	33	30	30	30	30	31	35	32	31	30	28	31
	2	12	12	21	21	21	8	8	9	8	7	6	7
	3	7	7	16	16	16	17	22	18	17	17	2	4
	4	23	23	7	7	7	21	19	22	21	27	26	20
	5	17	17	16	15	13	30	36	31	30	39	38	37
	6	24	30	30	30	30	31	6	32	31	35	34	31
Total des attentes tâches		116	119	120	119	117	138	126	144	138	155	134	130
Attente maximum de la tâche :	1	18	15	15	9	10	12	11	15	15	21	19	11
	2	7	7	8	8	8	7	8	8	7	7	5	7
	3	6	6	14	15	7	15	11	9	15	15	1	2
	4	8	8	5	5	3	8	8	8	8	15	9	8
	5	6	6	6	6	6	8	24	12	11	13	23	20
	6	13	17	18	19	30	18	5	32	18	19	19	18
Nombre max de tâches en attente devant la machine	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	3	3	3	2	2	3	2	2	3	2	2	2
	3	2	2	2	2	2	2	2	2	2	2	2	2
	4	1	1	1	1	1	1	1	1	1	2	2	1
	5	3	3	2	2	2	2	2	3	2	4	2	3
	6	1	1	1	1	1	2	1	2	2	2	1	1
Inoccupation totale de la machine	1	15	15	24	24	24	13	20	14	13	23	22	21
	2	2	2	2	1	9	2	11	9	2	2	9	2
	3	28	34	34	34	34	35	10	36	35	39	38	35
	4	37	37	46	46	46	33	39	34	33	42	41	40
	5	19	19	19	19	19	20	21	21	20	24	23	20
	6	15	15	11	11	11	13	14	14	13	19	18	15
Total inoccupation machines		116	122	136	135	143	116	115	128	116	149	151	133
Inoccupation max de la machine	1	9	9	14	15	17	9	8	7	9	13	13	8
	2	2	2	2	1	9	2	11	9	2	2	5	2
	3	28	34	34	34	34	35	9	35	35	39	31	35
	4	13	13	23	24	26	18	16	16	15	18	17	18
	5	19	19	19	19	19	20	21	21	20	20	18	20
	6	10	10	10	9	6	10	6	8	10	9	9	9

VALEURS DES CRITERES

ORDONNANCEMENT N°

CRITERE		25	26	27	28	29	30	31	32	33	34	35	36
Achèvement de la tâche:	1	56	56	56	61	60	56	61	53	54	53	51	49
	2	54	54	54	55	54	54	55	54	55	54	54	53
	3	50	50	50	55	42	50	47	60	61	60	45	35
	4	56	56	56	53	59	52	64	56	57	56	49	57
	5	55	55	55	61	55	60	56	55	56	55	72	65
	6	61	61	61	36	55	61	36	48	49	48	73	64
F_{max}		61	61	61	61	60	61	64	60	61	60	73	65
$\bar{F}$		55,33	55,33	55,33	53,50	54,17	55,50	53,17	54,33	55,33	54,33	57,33	53,83
Avance (retard) de la tâche :	1	0	0	0	-5	-4	0	-5	3	2	3	5	7
	2	2	2	2	1	2	2	1	2	1	2	2	3
	3	0	0	0	-5	8	0	3	-10	-11	-10	5	15
	4	-1	-1	-1	2	-4	3	-9	-1	-2	-1	6	-2
	5	2	2	2	-4	2	-3	1	2	1	2	-15	-8
	6	0	0	0	25	6	0	25	13	12	13	-12	-3
Avance (retard) min (max)		-1	-1	-1	-5	-4	-3	-9	-10	-11	-10	-15	-8
Avance (retard) moyenne		0,50	0,50	0,50	2,33	1,67	0,33	2,67	1,50	0,50	1,50	-1,50	2
Somme des retards		1	1	1	14	8	3	14	11	13	11	27	13
Somme des avances		4	4	4	28	18	5	30	20	16	20	18	25
Attente totale de la tâche :	1	30	30	30	35	34	30	35	27	28	27	25	23
	2	8	8	8	8	8	7	9	8	9	8	7	6
	3	16	16	16	22	8	16	13	26	27	26	11	2
	4	21	21	21	19	24	17	29	21	22	21	14	22
	5	30	30	30	36	30	35	31	30	31	30	47	40
	6	31	31	31	6	25	31	6	18	19	18	43	34
Total des attentes tâches :		136	136	136	126	126	129	123	130	136	130	147	127
Attente max de la tâche	1	15	11	15	15	12	11	13	11	15	15	16	14
	2	7	7	7	8	7	7	8	7	8	7	7	5
	3	15	15	12	13	7	15	10	25	19	25	10	1
	4	8	8	11	8	10	8	12	8	8	8	8	8
	5	7	8	11	24	8	17	16	8	12	11	26	23
	6	18	18	16	6	13	18	5	13	19	13	19	19
Nombre max de tâches en attente devant la machine	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	2	2	3	2	3	2	2	2	2	3	2	2
	3	2	2	2	2	2	2	2	2	2	2	2	2
	4	1	1	1	1	1	1	1	1	1	1	2	2
	5	2	2	2	2	2	2	2	2	3	2	4	2
	6	2	2	2	1	2	2	2	2	2	2	2	1
Inoccupation totale de la machine :	1	13	13	13	20	13	19	14	13	14	13	31	24
	2	2	2	5	9	2	2	11	2	9	2	2	9
	3	35	35	35	10	29	35	10	22	23	22	47	38
	4	33	33	33	39	33	38	34	33	34	33	50	43
	5	20	20	20	21	20	20	21	20	21	20	32	23
	6	13	13	13	14	16	13	21	13	14	13	19	18
Total inoccupation machine		116	116	119	113	113	127	111	103	115	103	181	155
Inoccupation maximum de la machine :	1	6	6	12	7	9	6	8	6	7	9	13	13
	2	2	2	5	9	2	2	11	2	9	2	2	5
	3	35	35	34	9	29	35	9	22	22	22	47	31
	4	18	18	15	16	18	18	16	18	16	15	18	17
	5	20	20	20	21	20	20	21	20	21	20	20	18
	6	9	9	10	8	10	9	7	9	8	10	9	9

VALEUR DES CRITERES

ORDONNANCEMENT N°

CRITERE		37	38	39	40	41	42
Achèvement de la tâche :	1	52	48	56	56	55	47
	2	61	82	54	56	59	88
	3	59	34	37	37	41	54
	4	76	51	58	54	58	41
	5	62	37	66	62	44	49
	6	42	63	65	61	60	37
F_{max}		76	82	66	62	60	88
$\bar{F}$		58,67	52,50	56	54,33	52,83	52,67
Avance (retard) de la tâche :	1	4	8	0	0	1	9
	2	-5	-26	2	0	-3	-32
	3	-9	16	13	13	9	-4
	4	-21	4	-3	1	-3	14
	5	-5	20	-9	-5	13	8
	6	19	-2	-4	0	1	24
Avance (retard) min. (max.)		-21	-26	-9	-5	-3	-32
Avance (retard) moyenne		-2,83	3,33	-0,17	1,50	3	3,17
Somme des retards		40	28	16	5	6	36
Somme des avances		23	48	15	14	24	55
Attente totale de la tâche :	1	26	22	30	30	29	21
	2	14	35	7	9	12	41
	3	25	0	4	3	7	20
	4	41	16	23	19	23	6
	5	37	12	41	37	19	24
	6	12	33	35	31	30	7
Total des attentes tâches		155	118	140	129	120	119
Attente maximum de la tâche	1	20	13	21	18	15	15
	2	14	23	5	7	7	27
	3	17	0	3	2	3	12
	4	35	16	10	8	11	3
	5	16	6	23	20	9	11
	6	22	22	22	18	15	7
Nombre maximum de tâches en attente devant la machine	1	1	1	1	1	1	1
	2	2	2	2	2	3	2
	3	2	2	2	2	2	2
	4	1	2	2	1	1	1
	5	5	3	4	3	3	4
	6	1	1	1	1	1	1
Inoccupation totale de la machine :	1	20	38	25	21	15	44
	2	9	1	1	2	5	9
	3	16	37	39	35	34	11
	4	40	60	44	40	37	66
	5	27	22	24	20	19	24
	6	33	25	19	15	15	31
Total inoccupation machines		145	183	152	133	125	185
Inoccupation maximum de la machine	1	13	16	14	6	12	26
	2	9	1	1	2	5	9
	3	14	37	39	35	33	9
	4	33	41	18	20	11	35
	5	27	27	20	20	19	24
	6	16	9	9	9	10	19

Remarques :

- Il n'y a pas de corrélation entre certaines variables ; exemples :

a. critères, fonctions de la durée d'occupation de l'atelier (F)

ordonnancement n° :

	8	12	14
F_{max} :	83	68	60
$\overline{F}$:	49,17	52,33	52,50

Ce n'est donc pas parce que l'atelier se vide plus rapidement de toutes les tâches que la durée moyenne d'occupation d'une tâche sera moins élevée ;

b. critères, fonctions des retards (L)

ordonnancement n°

	1	8	14
avance minimum :	-11	-27	-3
écart moyen :	1	6,67	3,33
somme retards :	21	27	6
somme avances :	27	67	26

Ce n'est pas parce que le retard maximum (avance minimum) subi par une tâche est moins élevé, que l'avance (retard) moyenne sera moins élevée : il suffirait que toutes les tâches soient en avance excepté une, dont le retard serait important (cas de l'ordonnancement 8). De plus, ce n'est parce que la somme des retards diminue que la somme des avances va augmenter (il faudrait pour cela que les dates de livraison soient identiques).

c. critères, fonction de $\overline{F}$ et de l'inoccupation des machines

ordonnancement n° :

	1	4	5
$\overline{F}$:	54,83	55,83	62,50
Somme des inocupations :	151	119	208

Ce n'est pas parce que les tâches quittent plus rapidement l'atelier, que les machines resteront moins longtemps inoccupées.

- <u>Il y a corrélation entre certaines variables</u> ; exemples :

  Il y a corrélation parfaite entre la somme des durées d'attente des tâches, la durée moyenne d'occupation de l'atelier par une tâche et l'écart moyen (différence entre la date d'achèvement d'une tâche et sa date de livraison) ; en effet :

$$W_i = F_i - \sum_{j=1}^{g_i} p_{ij}$$

$$L_i = F_i - d_i$$

  Or, la somme des durées de traitement d'une tâche et sa date de livraison sont des constantes. Il est donc logique d'aboutir à une corrélation parfaite ; cela rejoint d'ailleurs ce que nous disions dans le premier chapitre de la deuxième partie à propos des mesures régulières.

- Ce manque de corrélation rend le <u>jugement de valeur</u> très difficile, lorsque deux ordonnancements sont presque identiques. Prenons les ordonnancements n° 12 et 15.

  La valeur de F_{max} (68), de $\overline{F}$ (52,33), de l'avance minimum (-12), de la somme des retards (12) et de la somme des avances (33) est identique dans les deux cas.

  A l'actif de l'ordonnancement 12, mentionnons :
  - le total des heures attendues par les tâches (119 contre 120),
  - la machine 4 ne nécessite aucun emplacement de stocks intermédiaires, toutes les tâches y étant traitées tout de suite ; ce n'est pas le cas de l'ordonnancement 15.

  A l'actif de l'ordonnancement 15, mentionnons :
  - une opération n'attendra jamais plus de 18 périodes (30 périodes dans l'ordonnancement n° 12)
  - la durée totale d'inoccupation machine s'élève à 136 (143 dans l'autre cas)
  - la durée moyenne pendant laquelle une machine est inoccupée (calculée sur les durées maxima des arrêts-machine) est de 17 contre 18,5.

Cet exemple témoigne de la difficulté de choisir un seul critère et de la nécessité d'en prendre plusieurs. Un exemple encore plus frappant se trouve dans la comparaison des ordonnancements 7 et 8 :

ordonnancement n° 7

- aucune tâche en retard
- $\bar{F} = 55,83$
- $F_{max} = 61$
- somme des avances : 0
- somme des retards : 0
- total attente tâche : 98

ordonnancement n° 8

- 1 tâche très en retard (27)
- $\bar{F} = 49,17$
- $F_{max} = 83$
- somme des avances : 67
- somme des retards : 27
- total attente-tâche : 121
- stocks intermédiaires moins importants.

Lequel choisir ?

APPENDICE

A) Modèles étudiés dans la seconde partie

Dans le tableau suivant, les deux premières colonnes indiquent le nombre de tâches et de machines. Les 11 colonnes suivantes reprennent les hypothèses citées dans le premier chapitre de la seconde partie. Rappelons-les :

1) Une machine est un simple intervalle de temps.

2) Il n'existe qu'une machine de chaque type.

3) Une machine ne peut traiter qu'une seule opération à la fois.

4) Une opération commencée doit être terminée et ne peut être interrompue : dans le cas où le nombre de machines s'élève à un, où la mesure d'évaluation est régulière et où toutes les tâches sont disponibles au même moment, le théorème 1 (Titre II, chap. 2) s'applique et exclut cette hypothèse ; ce cas est noté (1).

5) Une opération consistant dans la fabrication de pièces identiques ne peut être fractionnée.

6) Les durées de traitement sont connues soit avec exactitude, soit en probabilités.

7) Les durées de préparation-machine sont indépendantes de l'ordonnancement ; dans ce cas elles sont inclues dans la durée de traitement.

8) Le routing est strict et l'atelier uniforme ; sans signification, dans le cas d'un seul type de machine, nous l'avons noté (2).

9) Une opération doit être ordonnancée le plus rapidement possible et ne peut être retardée que pour des raisons d'ordonnancement ou de routing.

10) Les tâches sont connues dès leur arrivée : même si leur durées de traitement sont connues en probabilité, leurs dates d'arrivées sont connues ; selon que cette hypothèse est respectée ou non, nous avons affaire à un problème statique ou dynamique.

11) Les durées de transport sont négligées, à moins qu'elles ne soient indépendantes de l'ordonnancement ; dans ce cas, comme dans celui des durées de préparation-machines, ces durées seront inclues dans la durée de traitement.

Les deux colonnes suivantes précisent deux points :

A) Aucune tâche n'est prioritaire a priori : aucun coefficient d'importance n'est assigné aux différentes tâches.

B) Toutes les tâches arrivent au même moment ; si ce n'est pas le cas, on suppose que les dates d'arrivées sont connues (hypothèse 10).

TACHES	MACHINES	1	2	3	4	5	6	7	8	9	10	11	A	B	CRITERE
		HYPOTHESES ET PRECISIONS													
n	1	V	V	V	(1)	V	V	V	(2)	V	V	V	V	V	$\min F_{max}$
n	1	V	V	V	(1)	V	V	V	(2)	V	V	V	V	V	$\min(\max)\bar{F}$
n	1	V	V	V	(1)	V	V	V	(2)	V	V	V	V	V	$\min L_{max}$; $\min T_{max}$
n	1	V	V	V	(1)	V	V	V	(2)	V	V	V	V	V	$\max L_{min}$; $\max T_{min}$
n	1	V	V	V	(1)	V	V	V	(2)	V	V	V	V	V	$\min \bar{T}$
n	1	V	V	V	(1)	V	V	V	(2)	V	V	V	V	V	$\min T_{max}$ ET $\min \bar{F}$
n	1	V	V	V	(1)	V	-	V	(2)	V	V	V	V	V	$E(\bar{F})$
n	1	V	V	V	(1)	V	V	V	(2)	V	V	V	-	V	$\min(\max)\bar{F}_u$
n	1	V	V	V	(1)	V	-	V	(2)	V	V	V	-	V	$E(\bar{F}_u)$
n	1	V	V	V	(1)	V	V	V	(2)	V	V	V	-	V	$\min \bar{T}_u$
n	1	V	V	V	(1)	V	V	-	(2)	V	V	V	V	V	$\min F_{max}$
n	1	V	V	V	-	-	V	V	(2)	-	V	V	V	-	$\min F_{max}$; $\min C_{max}$; $\min \bar{F}$
n	1	V	V	V	-	-	V	V	(2)	-	V	V	V	-	$\min \bar{F}$
n	1	V	-	V	(1)	V	V	V	(2)	V	V	V	V	V	$\min F_{max}$; $\min C_{max}$; $\min \bar{F}$
n	1	V	-	V	(1)	-	V	V	(2)	V	V	V	V	V	$\min \bar{F}$
n	1	V	-	V	(1)	-	V	V	(2)	V	V	V	V	V	$\min F_{max}$; $\min C_{max}$
n	1	V	-	V	(1)	-	V	V	(2)	V	V	V	V	V	$\min \bar{F}$, $\min \bar{C}$, $\min \bar{I}$ et $\neq \bar{I}$
n	1	V	-	V	(1)	V	V	V	(2)	V	V	V	V	V	$\min \bar{F}$, $\min \bar{W}$, $\min \bar{I}$ et $\neq \bar{I}$
n	1	V	-	V	(1)	-	V	V	(2)	V	V	V	-	V	$\min \bar{F}_u$
n	2	V	V	V	V	V	V	V	V	V	V	V	V	V	$\min F_{max}$
n	2	V	V	V	V	V	V	V	V	-	V	V	V	V	$\min F_{max}$
n	2	V	V	V	V	V	V	V	V	V	V	V	V	V	$\min \bar{F}$
n	3	V	V	V	V	V	V	V	V	V	V	V	V	V	$\min F_{max}$
n	3	V	V	V	V	V	V	V	V	V	V	V	V	V	$\min F_{max}$
n	n + 2	V	V	V	V	V	V	V	V	V	V	V	V	V	$\min F_{max}$
2	m	V	V	V	V	V	V	V	V	V	V	V	V	V	$\min F_{max}$
n	m	V	V	V	V	V	V	V	V	V	V	V	V	V	$\min F_{max}$
n	m	V	V	V	V	V	V	V	V	V	V	V	V	V	$\min F_{max}$
n	2	V	V	V	V	V	V	V	-	V	V	V	V	V	$\min F_{max}$
2	m	V	V	V	V	V	V	V	-	V	V	V	V	V	$\min F_{max}$
n	m	V	V	V	V	V	V	V	-	V	V	V	V	V	$\min F_{max}$
n	m	V	V	V	V	V	V	V	-	V	V	V	V	V	quelconque

V = hypothèse maintenue

\- = hypothèse levée

Méthode pour trouver le meilleur ordonnancement	REMARQUES	Référence (Titre II)
F_{max} est constant		Chap.2,sect.1
règle SPT(LPT)	SPT minimise(LPT maximise)$\bar{L},\bar{W},\bar{C},F_{min},C_{min},W_{max},W_{min} \neq 0,\bar{W}$	Ann.III,sect.1
règle:date de livraison	théorème 2	Chap.2,sect.2 §2
règle:temps restant	théorème 3	Chap.2,sect.2 §2
règle SPT	uniquement si toutes les tâches sont en retard	Chap.2,sect.2 §2
Théorème de Smith		Ann.III,sect.2
formule	théorème 4	Chap.2,sect.3 §1
règle SPT(LPT)pondéré	SPT(LPT)minimise(maximise) aussi $\bar{L}_u$ et $\bar{W}_u$	Ann.III,sect.3 §1
formule		Ann.III,sect.3 §1
diverses	Les méthodes varient selon les conditions particulières	Ann.III,sect.3 §2
Algorithmes	Branch and Bound;programmation dynamique	Ann.III,sect.4
règle SPRT	uniquement pour le premier type de reprise	Chap.2,sect.3 §4
règle SPRT	uniquement pour le second type de reprise	Chap.2,sect.3 §4
$\bar{F}=F_{max}=C_{max}$=constant	durées de traitement identiques; pas de fractionnement	Chap.2,sect.3 §5
$\bar{F}=(m+1)p/2m$	durées de traitement identiques; avec fractionnement	Chap.2,sect.3 §5
$F_{max}=C_{max}$	durées de traitement identiques; avec fractionnement	Chap.2,sect.3 §5
règle : variante SPT	durées de traitement différentes; avec fractionnement	Chap.2,sect.3 §5
règle : variante SPT	durées de traitement différentes; sans fractionnement	Chap.2,sect.3 §5
limite inférieure	formule de Eastman,Even et Isaacs	Chap.2,sect.3 §5
Algorithme de Johnson		Ann.III,sect.5 §1
Algorithme de Mitten	Introduction de délais	Ann.III,sect.5 §1
---	examen par couple	Ann.III,sect.5 §2
Algorithme de Johnson	uniquement si $\min(A_i)$ $\max(B_i)$ou si $\min(C_i)$ $\max(B_i)$	Chap.3,sect.2 §3
Branch and Bound	Ignall et Schrage; Lomnicki	Chap.3,sect.2 §3
Algorithme de Johnson	uniquement si les deuxièmes opérations sont traitées par des machines différentes	Chap.3,sect.2 §3
Solution géométrique	Akers	Chap.3,sect.2 §4
---	Solution de Dudek et Teuton, réfutée par Karush	Chap.3,sect.2 §5
ordonnancements aléatoires	normalité de F_{max} (Heller)	Chap.3,sect.2 §5
Généralisation de Johnson	Jackson	Chap.4,sect.2 §1
Solution géométrique	Akers	Chap.4,sect.2 §2
programmation linéaire	conditions restrictives (Bowman,Wagner,Manne)	Chap.4,sect.3
règles de décision	simulation	Chap.4,sect.4

B) Signification des symboles mathématiques et des abréviations

1) Symboles mathématiques applicables à tous les modèles

n : nombre de tâches à ordonnancer
i : numéro d'identification d'une tâche particulière ($i = 1 \dots n$)
m : nombre de machines
k : numéro d'identification d'une machine particulière ($k = 1 \dots m$)
j : numéro de séquence d'une opération
N : nombre de tâches présentes, à un moment donné, dans l'atelier
t : date à laquelle une décision doit être prise
U : coefficient d'utilisation de l'atelier
G : a) nombre total d'opérations à ordonnancer
b) forme de routing : atelier quelconque
F : forme de routing : atelier uniforme

g_i : nombre d'opérations de la tâche i
g_k : nombre d'opérations à traiter par la machine k
r_i : date d'arrivée de la tâche i
d_i : date à laquelle la tâche devrait être terminée (date de livraison)
a_i : durée pendant laquelle une tâche i peut rester dans l'atelier sans subir de retard : marge de la tâche i
u_i : coefficient d'importance attribué à la tâche i
s_i : (slack) : marge restante de la tâche i (= temps restant) ; durée pendant laquelle le traitement d'une opération de la tâche i peut être différé, sans encourir des retards ; si s_i est négatif, cela signifie que la tâche i subira un retard minimum égal à la valeur absolue de s_i
W_i : durée totale d'attente de la tâche i
F_i : durée pendant laquelle une tâche i est restée dans l'atelier
C_i : date à laquelle la tâche i quitte l'atelier
L_i : différence entre la date à laquelle la tâche i quitte l'atelier et la date à laquelle cette tâche devrait quitter l'atelier
T_i : retard de la tâche i : L_i est plus grand ou égal à 0
E_i : avance de la tâche i : L_i est plus petit ou égal à 0
C_{max} : valeur maximum des C_i : date à laquelle toutes les tâches sont terminées
F_{max} : valeur maximum des F_i : durée maximum pendant laquelle une tâche est restée dans l'atelier
L_{max} : écart maximum (retard ou avance) subi par une (ou plusieurs) tâches
T_{max} : retard maximum subi par une (ou) plusieurs des n tâches
C_{min}, F_{min}, L_{min}, T_{min} représentent les mêmes concepts mais en prenant la valeur minimum
$\overline{F}$: somme des F_i divisée par n : durée moyenne pendant laquelle une tâche est restée dans l'atelier, Lorsque le processus est dynamique ou que les durées de traitement ne sont pas connues avec exactitude, on parlera de l'espérance mathématique de $\overline{F}$: $E(\overline{F})$

$\bar{T}$: retard moyen des n tâches
$\bar{L}$: écart moyen des n tâches
$\bar{N}$: nombre moyen de tâches présentes, à un moment donné, dans l'atelier

p_{ijk} : durée de traitement de l'opération j, appartenant à la tâche i et devant être traitée par la machine k
$[i]$: opération qui doit être traitée en ième lieu
$p_{[i]}$: durée de traitement de l'opération qui doit être traitée en ième lieu

2) Symboles utilisés occasionnellement

A_i, B_i, C_i : durée de traitement des opérations de la tâche i, lorsque la première de ces opérations doit être traitée par la machine A, la deuxième par la machine B, la troisième par la machine C ; cette notation est utilisée dans les modèles se basant sur l'algorithme de Johnson ; l'atelier est uniforme.

Z, S, g', g", g sont uniquement utilisés dans les méthodes de Branch and Bound.

x_i et y_i : délai initial et terminal : uniquement utilisés dans l'algorithme de Mitten.

M, T_{ik}, r_{ijk}, Y_{IJK}, Z_{ij} : ces notations sont uniquement utilisées dans le modèle de programmation linéaire

$\{S_{so}\}$, $\{S_{ip}\}$, $\{S^k_{ip}\}$, $\{S^k_{so}\}$, s_{jk} : ces notations sont uniquement utilisées dans les modèles de simulation.

Les autres symboles n'apparaissent qu'une seule fois et ils ont reçu une signification, lorsqu'ils furent rencontrés.

3) Abréviations

Toutes les abréviations font appel à l'une ou l'autre règle de décision. Qu'il nous suffise de les traduire.

SPT : Shortest Processing Time
LPT : Longest Processing Time
SPRT : Shortest Processing Remaining Time
MWKR : Most Work Remaining
LWKR : Least Work Remaining
MWKR-P : Most Work Remaining (minus) Processing Time
MWKR/P : Most Work Remaining (divided by) Processing Time
MOPNR : Most Operation Remaining
DD : Due-Date
S/OPN : Slack (per) Operation
FCFS : First Come, First Served ; aussi appellé : FIFO : First In, First Out
SLACK : règle basée sur les temps restants (voir s_i)

Remarques :

- Quand on parle de "Processing-time", on envisage la durée de traitement de l'opération ordonnançable d'une tâche.
- Quand on parle de "Work Remaining", il s'agit de la somme des durées de traitement de toutes les opérations non-ordonnancées d'une tâche.