RAIRO. RECHERCHE OPÉRATIONNELLE

JEAN-PAUL SOUBRIER

Un algorithme de résolution de problèmes d'ordonnancement dynamiques

RAIRO. Recherche opérationnelle, tome 16, nº 3 (1982), p. 219-239

<http://www.numdam.org/item?id=RO_1982__16_3_219_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 16, n° 3, août 1982, p. 219 à 239)

UN ALGORITHME DE RÉSOLUTION DE PROBLÈMES D'ORDONNANCEMENT DYNAMIQUES (*)

par Jean-Paul SOUBRIER [1]

Résumé. — *Nous proposons une extension dynamique du problème d'ordonnancement d'atelier noté n/m/G/T par Conway (n tâches/m machines/flot quelconque dans l'atelier G/critère des délais T) où les n tâches ne sont pas connues simultanément. Afin de mieux approcher la réalité nous nous sommes placés dans le contexte d'un ordonnancement dynamique et la méthode utilisée se décompose en un pré-ordonnancement par un algorithme de type glouton et un lissage par une méthode de pénalisation modifiée.*

Mots clés : Ordonnancement; atelier; contexte dynamique.

Abstract. — *We propose a dynamic generalization to the Conway's n/m/G/T job-shop scheduling problem (n jobs/m machines/a general flow pattern G/delay criterion T) in which the n jobs are not simultaneously available. To be more realistic, we consider a dynamic scheduling context the method of which is decomposed into a sequencing by a "greedy" algorithm and a regulation by a penalty's modified method.*

Keywords: Scheduling; job-shop; dynamic context.

I. INTRODUCTION

Le problème d'ordonnancement d'atelier est le suivant : un ensemble de machines doit exécuter un ensemble de tâches affectées d'un délai de réalisation et constituées d'opérations élémentaires ordonnées. Il faut déterminer les dates de passage des opérations sur les machines de façon à respecter la charge des machines et si possible les délais fixés. C'est, suivant Conway, le « $n/m/G/T$ job-shop scheduling problem » : n tâches, m machines, un atelier à flot quelconque G (l'ordre de passage des opérations sur les machines n'étant pas le même pour toutes les tâches), le critère d'évaluation de l'ordonnancement T étant ici le respect des délais par la mesure du retard des tâches.

Le problème ainsi défini est le plus souvent résolu par des méthodes combinatoires telles que « Branch and Bound » (*cf.* par exemple [5, 6, 7]) ou arbitrage sur les disjonctions dans des graphes (*cf.* [2, 3]). Toutefois il faut citer la démarche de Nepomiastchy [11] qui utilise une méthode de pénalisation. L'inconvénient majeur de ces méthodes est la nécessité de connaître au même

(*) Reçu novembre 1980.

([1]) E.N.S.E.E.I.H.T., 2, rue Charles-Camichel, 31071 Toulouse Cedex.

instant toutes les tâches. Ce sont des méthodes d'ordonnancement de type statique où la production est fixée à l'avance, ce qui paraît peu réaliste.

C'est pourquoi nous introduisons ici une méthode d'*ordonnancement dynamique* qui pallie cet inconvénient majeur : l'ensemble des tâches peut évoluer dans le temps. Ce modèle permet ainsi de prendre en compte le cas d'un atelier fonctionnant avec une production « à la demande ».

Notre problème prolonge donc celui étudié par Nepomiastchy [11]. L'extension fondamentale est la connaissance des tâches évoluant dans le temps. Le modèle reprend la notion de groupes de machines à composition variable dans le temps mais il est étendu au cas de graphe quelconque des opérations d'une tâche (i. e. plusieurs machines différentes peuvent être nécessaires simultanément pour la même tâche).

L'approche de tels problèmes concrets interdit l'utilisation des méthodes de type précédent. En effet toute adaptation de ces méthodes à une production à la demande paraît irréalisable par son coût prohibitif dû au très grand nombre de variables à introduire. L'algorithme que nous proposons est donc nouveau dans sa conception et paraît particulièrement efficace pour traiter des problèmes réels.

Un ordonnancement *périodique* des tâches connues est réalisé. La méthode se décompose en deux phases se répétant périodiquement : un *chargement* et une *régulation*. Le chargement des machines avec toutes les opérations connues utilise un système de priorités dynamiques suivant un algorithme heuristique inspiré de [10]. Les dates de début des opérations sont ainsi assignées de manière *semi-définitive*. La régulation de la charge des machines est réalisée par une méthode de pénalisation semblable à celle référencée en [11]. L'ordonnancement définitif de la période immédiate est ainsi réalisé.

Notre méthode permet de fournir avec une relative rapidité une solution qui, dans tous les cas, *respecte la contrainte de charge* maximale des machines. De plus cette solution est *convenable du point de vue de la tenue des délais fixés*. Les temps de calcul étant relativement faibles, notre méthode peut être utilisée en tant qu'outil d'aide à la décision en permettant des essais successifs. Une utilisation en temps réel peut être envisagée, ce qui permettrait de traiter les aléas de fabrication.

II. LE PROBLÈME DYNAMIQUE

1. Les données

Les machines

On dispose de m groupes de machines (identiques dans un même groupe). Dans chaque groupe j ($j=1$ à m) il y a $M(j, \theta)$ machines dépendant du temps θ. Ceci permettra de prendre en compte des *modifications du parc* des machines et en particulier les aléas de fabrication tels que *panne* ou *maintenance.*

La capacité de production est donc connue dans le temps. Elle doit être en accord global avec les charges à absorber. Nous verrons qu'un des résultats de l'ordonnancement doit être de permettre l'ajustement des moyens de production.

Les tâches

Elles sont au nombre de n, mais ces n tâches ne sont pas connues au même instant. Pour chaque tâche i on connaîtra sa date d'arrivée a_i, sa date de début au plus tôt c_i et sa due-date d_i. Chaque tâche est composée d'opérations élémentaires $O_{i,j,k}$ (*k-ième* opération de la tâche i sur le groupe de machines j). L'ordre technologique des différentes opérations pour une tâche i est connu : il peut être représenté par un graphe quelconque. De même est connue la durée $f_{i,j,k}$ de l'opération $O_{i,j,k}$. Pour chaque opération $O_{i,j,k}$ il doit s'écouler un temps $t(O_{i,j,k}, O_{i,r,s})$ entre la fin de $O_{i,j,k}$ et le début de $O_{i,r,s}$ (même tâche i) et ceci pour tout r et tout s tel que $O_{i,j,k}$ précède immédiatement $O_{i,r,s}$. Ceci permet de prendre en compte les temps de transit dans l'atelier, même dans le cas d'une tâche non linéaire.

2. Les hypothèses

(*a*) *Hypothèses liées à la réalité de l'atelier*

Une opération commencée sur une machine ne peut être interrompue et doit être menée à son terme. A tout instant une machine ne peut être chargée avec plus d'une opération. Chaque tâche peut avoir plusieurs opérations sur un même groupe de machines (indice k). L'ordre technologique des opérations pour une tâche n'est pas supposé linéaire : une opération peut avoir plusieurs prédécesseurs et plusieurs successeurs.

(*b*) *Hypothèses qui pourront être levées sans difficulté*

On considérera que les machines peuvent travailler en *continu* afin de pouvoir effectuer une opération sur deux périodes consécutives. Les temps de transit *seront prix égaux à zéro.*

(*c*) *Hypothèses qui sont éminemment restrictives*

Nous supposons qu'il n'y aura pas d'aléas de fabrication concernant la période ordonnancée définitivement (période immédiate). Lorsque deux opérations se succèdent sur une même machine nous faisons l'hypothèse qu'il n'y a pas de temps d'adaptation de la machine.

3. Objectifs de l'ordonnancement

Le critère que l'on rencontre le plus souvent ([1, 3, 4, 6], ...) est celui de la minimisation du temps total d'exécution des tâches par les machines (C_{MAX} de Conway). Ce critère n'est pas du tout approprié dans le cas dynamique puisque le processus est ici continu.

Les critères fondamentaux envisageables pour évaluer un ordonnancement dynamique sont les suivants :

(*a*) *Critère du temps total réel de passage dans l'atelier*

Si D_i est sa date de fin réelle, f_i la durée minimale de réalisation de la tâche i si elle était seule, W_i son temps d'attente et C_i la date de début de la première opération de la tâche i alors le temps total réel de passage de la tâche i dans l'atelier est :

$$P_i = D_i - C_i = f_i + W_i.$$

En moyenne on considèrera $\bar{P} = \bar{f} + \bar{W}$ (1) où $\bar{f}$ est la durée minimale moyenne d'une tâche et $\bar{W}$ son temps d'attente moyen.

(*b*) *Critère des délais*

Le respect des délais paraît être un des objectifs les plus importants par le fait même qu'il peut y avoir une pénalité de retard associée au délai.

Le retard R_i de la tâche i est défini par $R_i = \text{SUP}(O, D_i - d_i)$. La moyenne des retards $\bar{R}$ est particulièrement importante. Elle apparaît comme une fonction de pénalisation à minimiser. Une pondération adéquate permettrait de tenir compte de cas où les retards des tâches ne sont pas d'égale importance.

(*c*) *Critère des en-cours*

Si on se limite à mesurer les en-cours par le nombre moyen de tâches en fabrication $\bar{N}$, alors, réduire les en-cours équivaut à réduire le temps moyen de passage $\bar{P}$ des tâches dans l'atelier.

En effet $\bar{N} = \lambda \bar{P}$ (2) où λ est le taux moyen des arrivées (la démonstration rigoureuse est donnée par Little [9] dans le cas général des files d'attente).

(*d*) *Critère des moyens de production*

Deux politiques différentes peuvent être envisagées :

– soit on considère le nombre de machines de chaque groupe fixé définitivement (y compris dans le temps) par les $M(j, \theta)$;

– soit ce nombre peut être dépassé : c'est le cas où l'on s'autorise la sous-traitance ou des heures supplémentaires.

Alors l'utilisation moyenne des machines est suivant le cas une donnée ou une variable du problème puisque le taux moyen d'utilisation $\bar{U}$ d'une machine est :

$$\bar{U}=\frac{\lambda \bar{f}}{\bar{m}} \qquad \text{avec} \quad \bar{m}=\lim_{t\to\infty}\frac{1}{t}\left(\sum_{j=1}^{m}\sum_{\theta=0}^{t}M(j,\theta)\right), \tag{3}$$

où $\bar{m}$ est le nombre moyen total de machines disponibles par unité de temps.

D'où d'après (1), (2) et (3) les relations :

$$\bar{P}=\bar{W}+\bar{f}=\frac{\bar{N}\bar{f}}{\bar{m}\bar{U}}.$$

4. Les choix possibles

Puisque les tâches arrivent dans l'atelier de façon continue l'ordonnancement devra être fait périodiquement (période T) à des époques $pT-t$ (p entier). Il ne concernera évidemment que l'ensemble des tâches connues à cet instant là où un sous-ensemble de ces tâches si on se limite à réaliser l'ordonnancement de l'intervalle $[pT, (p+1)T]$.

Le problème est de trouver un ordonnancement admissible (A) qui sera déterminé par la fixation des dates de début effectives $C_{i,j,k}$ des opérations des différentes tâches. Il n'est pas nécessaire de donner l'affectation sur les machines d'un même groupe : le fait qu'une opération commencée sur une machine ne se terminera pas sur une autre du même groupe est la conséquence du mode de fixation des $C_{i,j,k}$ et de la relation $D_{i,j,k}=C_{i,j,k}+f_{i,j,k}$ où $D_{i,j,k}$ est la date de fin effective de l'opération $O_{i,j,k}$. (Il suffit de charger sur la même machine du groupe les couples $(O_{i,j,k}, O_{l,j,n})$ tels que $D_{i,j,k}=C_{l,j,n}$.)

Définition d'un ordonnancement admissible (A)

(1) Toute tâche i ne commence qu'à une date C_i telle que $c_i \leqq C_i$.

(2) Toute tâche i finit à une date D_i telle que $D_i \leqq d_i$.

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

(4) Pour tout θ le nombre des opérations sur le groupe j est inférieur ou égal à $M(j, \theta)$.

(5) Pour chaque tâche, toute opération $O_{i,j,k}$ doit vérifier avec ses prédécesseurs immédiats $O_{i,r,s}$:

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

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

Il se peut que trouver (A) soit impossible à cause de la violation soit de (2) (les délais ne sont pas tous tenus), soit de (4) (la charge est dépassée sur certains groupes). Le choix de ne pas respecter (2) ou (4) dans ce cas dépend essentiellement du contexte dans lequel on cherche l'ordonnancement. Pour notre part nous nous sommes fixés pour but de respecter absolument la contrainte (4).

III. LE MODÈLE

1. Formalisation des conditions

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

Sans que cela entraîne une perte de généralité du problème nous supposerons dans la suite que toutes les données sont entières (a_i, c_i, d_i, f_i, ...,) : on peut donc se limiter à la recherche de solutions entières.

Pour la tâche i posons :

$$E_{i1} = \{ O_{i,j,k} / O_{i,j,k} \text{ sans prédécesseur} \},$$
$$E_{i2} = \{ O_{i,j,k} / O_{i,j,k} \text{ sans successeur} \},$$

alors :

$$\text{tâche } i = \{ O_{i,j,k} / O_{i,j,k} \in E_{i1} \cup E_{i2} \cup E_{13} \}.$$

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

— condition (1) $\Leftrightarrow \forall i$ et $\forall (j, k) / O_{i,j,k} \in E_{i1}$: $c_i \leqq C_{i,j,k}$;

— condition (2) $\Leftrightarrow \forall i$ et $\forall (j, k) / O_{i,j,k} \in E_{i2}$: $D_{i,j,k} \leqq d_i$ ($D_{i,j,k}$ étant la date de fin effective de l'opération $O_{i,j,k}$);

— (1) et (2) $\Leftrightarrow \forall i$ et $\forall$ $(j, k) / O_{i,j,k} \in E_{i3}$: $C^-_{i,j,k} \leqq C_{i,j,k} \leqq C^+_{i,j,k}$, $C^-_{i,j,k}$ et $C^+_{i,j,k}$ étant respectivement les dates de début au plus tôt et au plus tard et dépendant des opérations précédentes et suivantes :

$$C^-_{i,j,k} = \operatorname*{Sup}_{r,s} (C^-_{i,r,s} + f_{i,r,s}) \qquad \text{avec } r \text{ et } s \text{ tels que } O_{i,r,s} < O_{i,j,k},$$

$$C^+_{i,j,k} = \operatorname*{Inf}_{r,s} (C^+_{i,r,s}) - f_{i,j,k} \qquad \text{avec } r \text{ et } s \text{ tels que } O_{i,j,k} < O_{i,r,s};$$

— condition (3) $\Leftrightarrow \forall i, j, k$: $D_{i,j,k}=C_{i,j,k}+f_{i,j,k}$;

— condition (4) : θ étant entier, soit $M(j, \theta)$ le nombre de machines du groupe j disponibles pendant l'intervalle de temps $[\theta, \theta+1[$. Si nous posons :

$$F_j(\theta)=\{O_{i,j,k}/C_{i,j,k}\leqq\theta\leqq D_{i,j,k}-1\},$$

alors (4) $\Leftrightarrow \begin{pmatrix} \forall j \\ \forall \theta \end{pmatrix}$, $\operatorname{Card}(F_j(\theta))\leqq M(j, \theta)$

ou à l'aide de la variable booléenne :

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

et en posant :

$$J(j, \theta)=\sum_i \sum_k J_{i,k}(j, \theta),$$

alors $J(j, \theta)$ représente la charge du groupe de machine j sur $[\theta, \theta+1[$ et (4) $\Leftrightarrow \underline{J(j, \theta) \leqq M(j, \theta)}$, $\forall j=1, \ldots, m$ et $\forall \theta$ entier;

— condition (5) : nous n'avons envisagé pour l'instant que le cas où les temps de transit entre opérations étaient nuls.

Donc (5) $\Leftrightarrow \forall i$: $\forall (j, k)$ tels que $O_{i,j,k} \notin E_{i1}$, $\forall (r, s)$ tels que: $O_{i,r,s} < O_{i,j,k}$:$C_{i,r,s} + f_{i,r,s} \leqq C_{i,j,k}$.

2. Périodicité de l'ordonnancement et unité de temps

La résolution de l'ordonnancement sera effectuée périodiquement avec la période T : chaque résolution se décomposera en deux étapes que nous avons appelées « chargement » et « régulation ». A la date $t\,(t<q\,T)$ on ordonnancera définitivement (chargement et régulation) des opérations sur l'intervalle $[q\,T, (q+1)\,T[$, les périodes suivantes étant ordonnancées provisoirement (chargement uniquement) : les tâches concernées seront telles que $t-T\leqq a_i<t$.

Donc à des dates $t_0, t_0+T, \ldots, t_0+p\,T$ on effectuera le chargement des opérations connues à ces dates là, les $C_{i,j,k}$ étant fixés définitivement pour les intervalles respectifs (avec $t_0<\theta_0$) :

$$[\theta_0, \theta_0+T[, [\theta_0+T, \theta_0+2\,T[, \ldots, [\theta_0+p\,T, \theta_0+(p+1)\,T[.$$

Nous aurions pu nous limiter à effectuer l'ordonnancement en une seule étape sans faire de chargement préalable sur ces intervalles de temps. Ceci équivaudrait à résoudre un problème d'ordonnancement dynamique comme une série de problèmes statiques. Nous n'avons pas choisi cette approche qui implique un choix très délicat des opérations à ordonnancer définitivement sur la période immédiate.

La seule hypothèse que nous faisons sur la période T est qu'elle soit entière : il suffit pour cela de choisir une unité de temps convenable. Nous supposerons que tel est le cas et rappelons que toutes les données du problème sont des entiers.

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

3. Objectifs du modèle

Le principe du modèle a été défini en tenant compte que l'on ignore s'il existe une solution admissible (A) au problème et donc *a fortiori* une solution optimale.

Nous nous limitons donc à la recherche d'une solution convenable respectant la contrainte sur les moyens de production (4) mais pouvant ne pas tenir certains délais [partiellement (2)]. De plus, à l'aide de prises de décisions logiques, nous essaierons d'atteindre les objectifs tels que montant réduit des en-cours et bonne utilisation des machines. Le but du modèle est ainsi de tendre vers un ordonnancement admissible.

4. Principe de la méthode

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

$$C_{i,j,k} = C^{+}_{i,j,k} \quad \text{(ordonnancées au plus tard).}$$

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

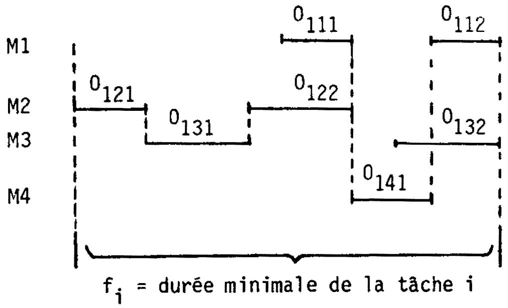
4.1. ÉTUDE DU MODE DE CHARGEMENT

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

4.1.1. *Discussion sur le choix du cadrage initial : marges et distances*

Pour chaque tâche i on connaît, à partir de son ordre technologique, l'ordonnancement au plus tard respectant le délai minimal de réalisation f_i (donné par un P.E.R.T.).

Exemple : 4 machines distinctes/1 tâche/7 opérations.



La marge totale de la tâche est $d_i - c_i - f_i$.

Pour une opération $O_{i,j,k}$ nous distinguerons les notions de marge et de distances :

la marge amont : durée totale dont on peut avancer une opération sans déborder en amont de la date de début au plus tôt c_i de la tâche;

la marge aval : durée totale dont on peut retarder une opération sans dépasser la date de fin au plus tard fixée;

la distance amont: $\operatorname{Inf}_{r,s}$ (durée séparant l'opération d'une de ses précédentes immédiates $O_{i,r,s}$);

la distance aval : $\operatorname{Inf}_{r,s}$ (durée séparant l'opération d'une de ses suivantes immédiates $O_{i,r,s}$).

Les marges nous permettront de déterminer $C^-_{i,j,k}$ et $C^+_{i,j,k}$ lorsqu'on chargera l'opération $O_{i,j,k}$.

Les distances serviront à fixer l'intervalle sur lequel il est possible d'ordonnancer une opération sans influer sur l'ordonnancement des opérations de la même tâche qui l'encadrent.

Il faut cadrer ce diagramme minimal dans le temps. Deux grandes possibilités existent pour ce choix de la répartition des marges : marge totale en aval ou marge totale en amont ce qui correspond respectivement aux cadrages au plus tôt et au plus tard (la 3[e] grande possibilité, marge uniformément répartie entre les diverses opérations, paraît sans intérêt puisqu'elle restreint le champ de manœuvre).

Il nous est apparu préférable de partir du cadrage minimum au plus tard car cela permet de prendre en compte plus facilement les délais. Initialement on part d'un montant d'en-cours minimal. D'autre part la charge des machines sera globalement moins élevée pour le très court terme, d'où la possibilité de charger plus facilement des opérations arrivées plus tard avec des marges faibles. Par contre l'inconvénient sera une sous-utilisation des machines : c'est pourquoi nous avons prévu une « régulation » de la charge pour la période ordonnancée définitivement.

4.1.2. *Mode de chargement*

Étant donné le choix fait précédemment il paraît logique de réaliser le chargement des opérations suivant un ordre de priorité décroissante, donc en commençant par l'opération la plus contraignante. Le système de priorité sera dynamique c'est-à-dire que la priorité de chaque opération restante est recalculée après la fixation de la date de début de l'opération la plus pénalisée par une fonction donnant la liberté de manœuvre.

Considérons l'ordonnancement effectué à l'instant $t_0+q\,T$. Soit E l'ensemble des opérations $\leqq a_i < t_0+q\,T$: appartenant à des tâches telles que $t_0+(q-1)\,T$

$$E_0=\{O_{i,j,k}\in E/O_{i,j,k} \text{ a tous ses successeurs ordonnancés}\},$$

$$E_0=\bigcup_j E_{0,j}$$

avec $E_{0,j}$ le sous ensemble concernant le groupe j de machines.

Nous pouvons raisonner indépendamment pour le groupe j (il faudra bien sûr choisir ensuite l'ordre de prise en compte des diverses machines).

Soit j *fixé*, $\forall\, i, k$ tel que $O_{i,j,k}\in E_{0,j}$ il faut fixer $C_{i,j,k}$ tel que :

$$C^-_{i,j,k}\leqq C_{i,j,k}\leqq C^+_{i,j,k}. \tag{1}$$

Partons de la solution initiale consistant à ordonnancer $E_{0,j}$ au plus tard. La charge globale est alors décrite par $\sum_\theta J(j,\theta)$.

Remarque : *Supprimons l'opération $O_{i,j,k}$ de cet ordonnancement initial. Si cette opération était chargée en dernier (la dernière de $E_{0,j}$) il serait impossible de trouver un ordonnancement admissible (A) si :*

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

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

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

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

La charge, décrite par les $J(j, \theta)$, est modifiée après le chargement de chaque opération.

Pour pouvoir tenir compte à la fois de la durée de l'opération et s'assurer que l'opération peut respecter la condition (4) nous évaluerons la « liberté de manœuvre » $l_{i,j,k}$ de $O_{i,j,k}$:

avec $e_\alpha \begin{cases} = 1 & \text{si } J(j,\theta) < M(j,\theta), \quad \forall \theta \in \alpha, \alpha + f_{i,j,k}], \\ = 0 & \text{sinon.} \end{cases}$

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

Sur la machine j on ordonnancera donc d'abord $O_{i,j,k}$ telle que $l_{i,j,k}$ associé soit minimale. Parmi les emplacements possibles on choisira celui qui donne à $C_{i,j,k}$ la valeur la plus grande (chargement au plus tard si $l_{i,j,k} \neq 0$).

Cas où $l_{i,j,k} = 0$

Cela correspond au cas où il est impossible de charger $O_{i,j,k}$ en respectant (2).

Sur $[C^-_{i,j,k}, C^+_{i,j,k}]$ on recherche une solution compatible de déplacement d'autres opérations en envisageant pour $O_{i,j,k}$ les positions successives à partir de $C^+_{i,j,k}$. La recherche s'arrête dès qu'un emplacement libéré a été trouvé pour $O_{i,j,k}$ (l'ordonnancement systématique au plus tard pouvant permettre d'en trouver un). Pour ne pas engendrer trop de calculs nous nous limitons au déplacement d'opérations consécutives (auxquelles il faut rajouter les prédécesseurs éventuels de ces opérations).

S'il n'y a pas de solution, l'opération $O_{i,j,k}$ sera en retard (ceci implique que la tâche concernée sera en retard). Alors on ordonnancera $O_{i,j,k}$ au plus tôt après $C^+_{i,j,k}$.

Donc à chaque itération on fixe un $C_{i,j,k}$ et l'ensemble E_0 est modifié ainsi :

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

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

L'ordre dans lequel on envisage les différentes machines est l'ordre décroissant de la charge : ce qui revient à calculer $l_{i,j,k}$ pour tout $O_{i,j,k} \in E_0$ (donc $\forall j$) et à choisir $O_{i,j,k} \in E_0$ telle que $l_{i,j,k}$ minimale.

Lorsque E_0 est vide, toutes les opérations connues à la date $t_0 + q\,T$ ont été chargées. La charge des machines est décrite par $J(j, \theta)$. Puisque l'ordonnancement suivant aura lieu à $t_0 + (q+1)\,T$ et chargera sur l'intervalle $[\theta_0 + (q+1)\,T, \infty[$, la période $[\theta_0 + q\,T, \theta_0 + (q+1)\,T[$ doit être ordonnancée définitivement : c'est l'objet de la régulation de la charge.

4.1.3. *Exemple de chargement*

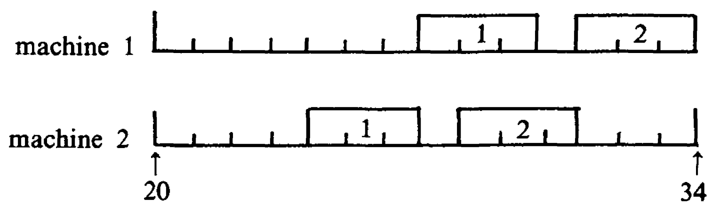
Tâche_i	a_i	c_i	d_i	
1	0	16	30	Chargement à la date 5
2	4	23	34	
3	14	24	34	Chargement à la date 19
4	18	22	32	
5	17	20	34	

Ordre technologique

« < » désigne la relation « précède immédiatement » et entre parenthèses figurent les durées des opérations :

tâche 1 : $O_{1,2,1}(3) < O_{1,1,1}(3)$;
tâche 2 : $O_{2,2,1}(3) < O_{2,1,1}(3)$;
tâche 3 : $O_{3,1,1}(3) < O_{3,2,1}(2)$;
tâche 4 : $O_{4,1,1}(2) < O_{4,2,1}(3)$;
tâche 5 : $O_{5,1,1}(1) < O_{5,2,1}(3) < O_{5,1,2}(1)$.

Le chargement définitif effectué à la date 5 a donné le résultat suivant :

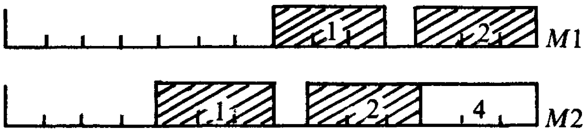


Nous nous intéressons alors au chargement à la date 19 (période suivante) concernant les tâches 3, 4 et 5. Les différentes étapes sont les suivantes :

Étape a

$O_{i,j,k}$	$O_{3,2,1}$	$O_{4,2,1}$	$O_{5,1,2}$
$l_{i,j,k}$	2	1	4

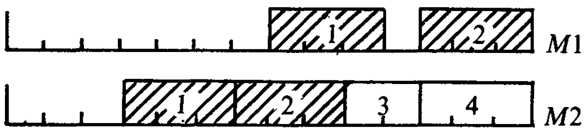
$\Longrightarrow$



Étape b

$O_{i,j,k}$	$O_{3,2,1}$	$O_{4,1,1}$	$O_{5,1,2}$
$l_{i,j,k}$	0	4	4

$\Longrightarrow$



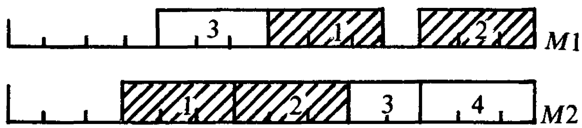
Il n'y a pas d'emplacement possible pour $O_{3,2,1}$ $(l_{3,2,1}=0)$.

La solution est ici obtenue par déplacement d'autres opérations déjà chargées.

Étape c

$O_{i,j,k}$	$O_{3,1,1}$	$O_{4,1,1}$	$O_{5,1,2}$
$l_{i,j,k}$	1	4	4

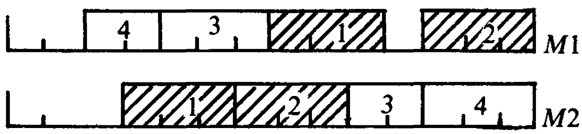
$\Longrightarrow$



Étape d

$O_{i,j,k}$	$O_{4,1,1}$	$O_{5,1,2}$
$l_{i,j,k}$	1	1

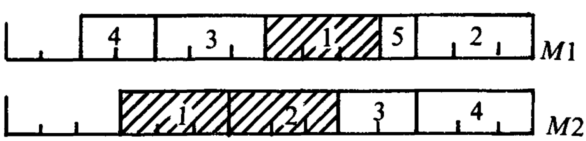
$\Longrightarrow$



(choix arbitraire).

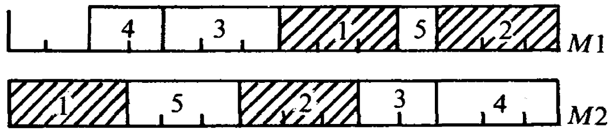
Étape e

Chargement de $O_{5,1,2}$ $\Longrightarrow$



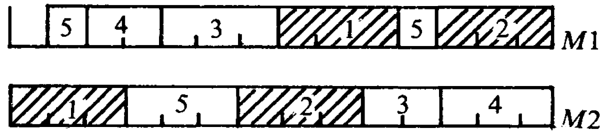
Étape f

Chargement de $O_{5,2,1}$ $\Longrightarrow$
par déplacement de $O_{1,2,1}$



Étape g

Chargement de $O_{5,1,1}$ $\Longrightarrow$



Ce dernier diagramme représente la solution de chargement obtenu dans ce cas par notre méthode.

4.2. RÉGULATION DE LA CHARGE

Le fait de cadrer des opérations le plus tard possible implique une sous-utilisation relative des machines en début de période : il faut donc avancer des opérations sur cette période à ordonnancer définitivement. (Les autres périodes ne sont pas concernées puisque d'autres tâches peuvent arriver et y être chargées ultérieurement).

4.2.1. *Problème R*

Soit l'ensemble des opérations pouvant être avancées sur la période $[\theta_0+q\,T, \theta_0+(q+1)\,T[$. Le problème est de choisir un sous-ensemble qui maximise l'utilisation des machines. Dans l'état actuel de la résolution nous nous contentons de considérer que les opérations déjà ordonnancées sur cette période, le sont définitivement. Nous essayons alors de combler les trous restants avec les opérations candidates de telle sorte que l'utilisation soit maximale. En pratique il s'avère que l'algorithme est peu combinatoire.

4.2.2. *Algorithme de régulation*

Sur $I_q=[\theta_0+q\,T, \theta_0+(q+1)\,T]$ l'utilisation de la machine j est :

$$U_j=\sum_{\theta \in I_q} J(j, \theta).$$

Soient les ensembles suivants :

$A=\{O_{i,j,k}/D_{i,j,k}\in I_r \text{ avec } r>q, C_{i,j,k}^{-}<\theta_0+(q+1)\,T\}$;

$A_1=\{O_{i,j,k}/O_{i,j,k}$ sans prédécesseurs dans $A\}$;

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

On ordonnance les opérations sur les intervalles libres $(J(j, \theta) < M(j, \theta))$ sans tenir compte de la contrainte (4) sur la limitation des moyens. (L'ordre est celui des θ croissants et les opérations appartenant à A_1, puis A_2, etc. cadrées au plus tôt). Il faut remarquer que seulement un sous-ensemble A_0 de A est envisagé puisque les $f_{i, j, k}$ interviennent.

Soit $n(j, \theta) = \mathrm{SUP}(J(j, \theta) - M(j, \theta), O)$.

Si $N = \sum_{j} \sum_{\theta \in I_q} n(j, \theta)$ alors (4) $\Leftrightarrow N = 0$ pour le problème R.

Principe de la régulation

Pour chaque opération de A_0 on essaie toutes les positions possibles sur $[C_{i, j, k}^{-}, \theta_0 + (q+1) T]$. On garde la position pour laquelle N est minimum. En cas d'indétermination on garde la position au plus tôt. Si l'opération est éliminée, elle revient à sa place initiale avec $C_{i, j, k} > \theta_0 + (q+1) T$.

On itère jusqu'à obtenir $N = 0$ i. e. la contrainte (4) est vérifiée.

Exemple de régulation

Pour illustrer notre modèle nous donnons ci-dessous un exemple simple du principe de la régulation. Cet exemple concerne deux machines distinctes $M1$ et $M2$ et une période immédiate de 14 unités de temps. Après chargement nous obtenons le diagramme (*fig.* 1) correspondant aux opérations ordonnancées définitivement.

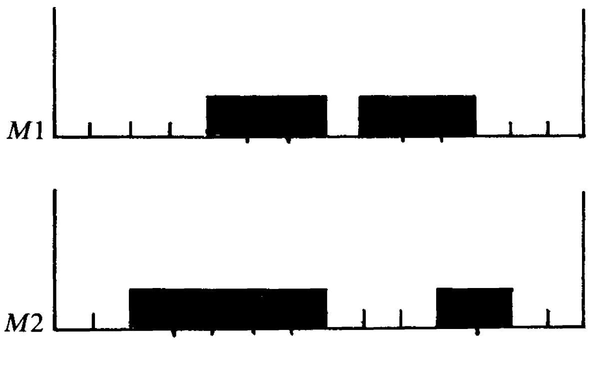
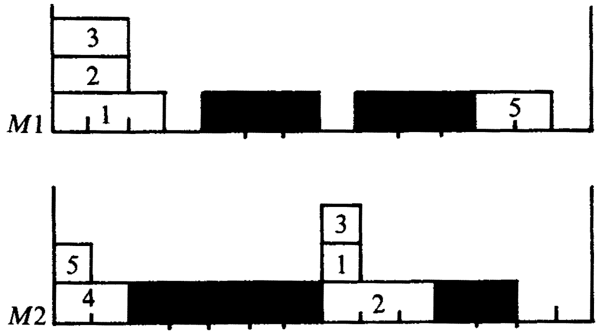


Figure 1

Afin de maximiser l'utilisation des machines il faut chercher à avancer des opérations chargées provisoirement sur des périodes postérieures. Ces opérations candidates sont représentées sur le diagramme de la figure 2 avec les contraintes de précédence suivantes :

$$O_{1,1} < O_{1,2},\ O_{2,1} < O_{2,2},\ O_{3,1} < O_{3,2},\ O_{5,2} < O_{5,1}$$

(le 1er indice désigne la tâche, le 2e la machine).

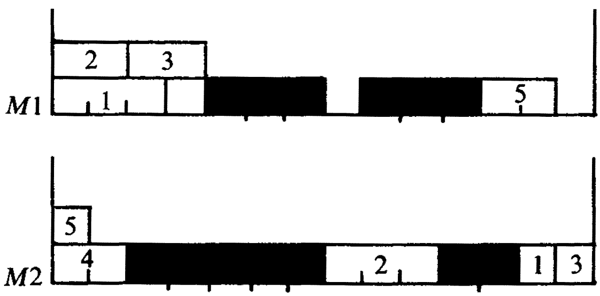


$N=7$
(les numéros figurant sur ce diagramme sont ceux des tâches)

Figure 2

L'algorithme de régulation s'applique de la manière suivante illustrée par les figures 2 à 5. Nous appelons « écrétage » le calcul de la nouvelle valeur de N en essayant toutes les positions possibles pour chaque opération et en gardant celle pour laquelle N est minimal. « Élimination » est le choix de l'opération qui, en revenant à sa place initiale, induit la variation d'utilisation réelle minimale de toutes les machines; *l'utilisation réelle* de la machine j à l'instant θ étant $\min(J(j, \theta), M(j, \theta))$.

Ecrétage (1)



$N=4$

Figure 3

Élimination (1)

Opérations sans successeurs	$O_{1,2}$	$O_{3,2}$	$O_{2,2}$	$O_{5,1}$	$O_{4,2}$
Variation de l'utilisation réelle $= \Delta U$	0 (2)	-1	-2 (2)	-1 (2)	-1

(2) Ces variations ΔU sont obtenues par remplacement ou déplacement d'une autre opération. Par exemple dans l'élimination (1) $O_{5,1}$ est remplacée par $O_{3,1}$; pour $O_{1,2}$, l'opération $O_{1,1}$ peut être déplacée et « superposée » à $O_{5,1}$.

L'opération $O_{1,2}$ est éliminée.

Ecrétage (2)

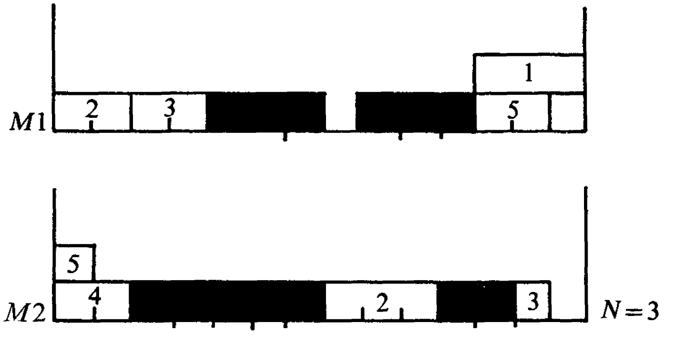


Figure 4

Elimination (2)

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

L'opération $O_{5,1}$ est éliminée.

Ecrétage (3)

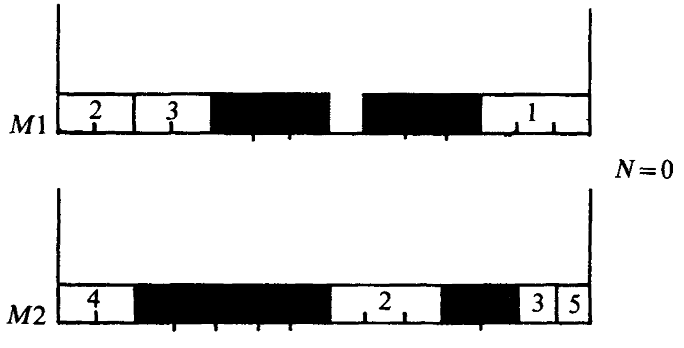


Figure 5

La solution admissible trouvée par notre méthode est celle représentée sur le diagramme de la figure 5.

4.3. L'ALGORITHME GÉNÉRAL de notre modèle est donc le suivant :

Ordonnancement dynamique

$q \leftarrow 0$
Répéter
Réaliser l'ordonnancement à la date $t_0 + q\,T$
$q \leftarrow q+1$
jusque (fin d'arrivée des tâches)

Ordonnancement à la date $t_0 + q\,T$

$E_0 \leftarrow E_2(t_0+q\,T)$
Tant que $(E_0 \neq \varnothing)$ faire
début
— Calcul des $l_{i,j,k}$ pour tout $O_{i,j,k} \in E_0$
— Ordonnancer $O_{i,j,k}$ correspondant à MIN $(l_{i,j,k})$
— $E_0 \leftarrow E_0 - \{O_{i,j,k}\} + \{\text{pred}(O_{i,j,k})\}$
fin
Régulation de la période $[\theta_0 + q\,T,\ \theta_0 + (q+1)\,T]$.

IV. IMPLÉMENTATION ET RÉSULTATS

Ce modèle a été implémenté sur ordinateur CII IRIS 80 en langage PL/1.

Les problèmes traités ont les caractéristiques suivantes : la période d'ordonnancement est le jour, l'unité de temps est l'heure et dans l'état actuel du programme nous nous limitons à envisager les cas où $M(j, \theta)=1$ pour tout j et tout θ.

Toutes les données sont générées aléatoirement : ordre technologique, date d'arrivée, délai, durées des opérations, etc.

Le programme a été testé avec succès sur des ateliers comportant jusqu'à 9 machines et pour des arrivées moyennes jusqu'à 14 opérations par jour et par machine.

Les temps nécessaires à l'obtention d'une solution dépendent essentiellement de la charge des machines sur la période considérée. Comme le montrent les tableaux suivants, ces temps (exprimés en secondes d'U.C. IRIS 80) varient de quelques secondes à quelques dizaines de secondes par période ordonnancée.

Le tableau I donne des exemples de résultats typiques obtenus en fonction du nombre de machines et du nombre d'opérations arrivant chaque jour. Certaines disparités dans les temps s'expliquent par des charges globales des machines plus ou moins importantes.

Le tableau II montre les temps obtenus sur des problèmes avec 5 machines en fonction des charges à absorber ainsi que le nombre de tâches en retard.

D'autres exemples de résultats peuvent être trouvés dans [12].

TABLEAU I

Nombre de machines	Nombre d'opérations par période	Durée d'une période d'ordonnancement (s)
3	15,3	7,56
3	40,9	31,50
5	33,2	31,06
7	34,0	58,44
9	37,2	39,48

Dans ces exemples la capacité de production est suffisante pour réaliser les tâches dans les délais fixés sauf pour les exemples avec 3 machines où dans ce cas seulement des tâches sont en retard.

TABLEAU II

Charge moyenne des machines (h)	Utilisation moyenne des machines (h)	Nombre de tâches en retard	Durée d'une période d'ordonnancement (s U.C.)
22,28	19,64	3	26,28
22,75	19,91	7	28,26
18,77	18,60	2	24,96
14,95	15,33	0	27,06

Les durées des opérations sont les seules données qui varient dans ces 4 problèmes. Leurs caractéristiques communes sont : 5 machines, 39 tâches et 259 opérations. L'ordonnancement concerne 10 périodes consécutives.

V. CONCLUSION

Comme nous l'avons vu précédemment les temps nécessaires à l'obtention d'une solution dépendent surtout de la charge des machines sur la période considérée. La solution trouvée est une solution acceptable vis-à-vis de l'utilisation des machines et de la tenue des délais : le mode d'assignation des dates de début aux opérations ne garantit donc pas une stricte minimisation des retards.

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

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

Cette méthode se prèterait bien à une application en temps réel. La possibilité de faire d'autres essais avec modifications de certains paramètres comme les délais conduirait à considérer la fixation de ceux-ci comme conséquence de l'ordonnancement. Une panne, ou un aléa de n'importe quelle nature, implique de refaire l'ordonnancement de la période immédiate et le chargement des périodes suivantes en prenant en compte la modification du parc des machines (ceci est possible grâce à la notion de groupes de machines variables dans le temps). C'est alors la périodicité de l'ordonnancement qui est remise en cause et on aboutit à réaliser cet ordonnancement en temps réels.

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

– d'une part, l'amélioration de la tenue des délais pour une partie des tâches, sinon toutes, en permettant une modification du mode de fixation;

– d'autre part l'ajustement des capacités des moyens de production au vu du retard des tâches, si cela est techniquement possible.

Ceci est envisageable car notre méthode fournit une solution avec un temps de calcul suffisamment faible pour permettre des essais successifs avec des données éventuellement modifiées tels les délais et la composition des groupes de machines dans le temps.

BIBLIOGRAPHIE

1. Ashour, *Sequencing Theory*, Springer Verlag, 1972.
2. Balas, *Machine Sequencing* via *Disjunctive Graphs: an Implicit Enumeration Algorithm*, Operations Research, n° 6, 1969.
3. Carlier, *Ordonnancement à contraintes disjonctives*, R.A.I.R.O., novembre 1978.
4. Conway/Maxwell, *Theory of Scheduling*, Addison Wesley, 1967.
5. Elmaghraby, *The Machine Sequencing Pb, Reviews and Extensions*, N.R.L.Q., n° 2, 1968.

6. Florian/Bratley/Robillard, *On Sequencing with Earliest Starts and Due-Dates with Applications to Computing Bounds for the $n/m/G/F_{max}$ Problems*, N.R.L.Q., vol. 20, 1973.

7. Ignall/Schrage, *Application to the Branch and Bound Technique to Some Flow-Shop Scheduling Problems*, Operations Research, mai-juin 1965.

8. Jackson, *Some Problems in Queuing with Dynamic Properties*, N.R.L.Q., n° 3, 1960.

9. Little, *A Proof for the Queuing Formula $L=\lambda W$*, Operations Research, n° 3, mai 1961.

10. Minoux, *Algorithmes Gloutons*, E.D.F. Bulletin, 1977.

11. Nepomiastchy, *Résolution d'un problème d'ordonnancement à ressources variables*, R.A.I.R.O., août 1978.

12. Soubrier, *Un modèle de résolution de problèmes d'ordonnancement dynamique*, *Thèse de Docteur-Ingénieur*, U.P.S., Toulouse, 1981.