

JACQUES CARLIER

Ordonnements à contraintes disjonctives

RAIRO. Recherche opérationnelle, tome 12, n° 4 (1978),
p. 333-350

http://www.numdam.org/item?id=RO_1978__12_4_333_0

© AFCET, 1978, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

ORDONNANCEMENTS A CONTRAINTES DISJONCTIVES (*)

par Jacques CARLIER ⁽¹⁾

Résumé. — *Dans cet article, nous proposons des méthodes pour les problèmes d'ordonnement à contraintes disjonctives : l'étude du graphe disjonctif nous amène à proposer une méthode arborescente utilisant la matrice des chemins de valeur maximale entre toute paire de sommets. De nouveaux algorithmes rendent cette méthode efficace. Un programme a été fait pour les problèmes d'ateliers; des exemples ayant 8 pièces et 9 machines, 14 pièces et 4 machines... ont été résolus optimalement.*

I. INTRODUCTION

La réalisation d'une œuvre importante telle que la construction d'un immeuble, la détermination du cycle de fabrication d'un atelier, ou l'établissement d'un emploi du temps, impose généralement l'accomplissement de tâches nombreuses et variées qui sont soumises à des contraintes de différents types. Établir un ordonnancement, c'est déterminer un ordre d'exécution de l'ensemble des tâches qui satisfait aux contraintes. Parmi tous les ordonnancements admissibles on devra en trouver un qui optimise un critère fixé à l'avance: minimisation d'un délai, maximisation d'un profit, lissage des effectifs de la main d'œuvre, étalement des charges financières, etc.

Jusqu'en 1958, le procédé de résolution le plus couramment utilisé par les praticiens était le diagramme de Gantt, ou planning à barres; cette année-là se sont développées parallèlement deux types de méthodes fondées sur le calcul des chemins ou tensions de valeur maximale d'un graphe valué: les méthodes C. P. M., P. E. R. T., et, la méthode des potentiels de Bernard Roy [13]. Ces méthodes ont apporté une amélioration considérable par rapport au planning à barres, mais permettent uniquement de minimiser le délai dans les problèmes d'ordonnement n'ayant que des contraintes potentielles. Dès qu'il y a des contraintes cumulatives, ce qui signifie que les moyens sont en quantité limitée, ou, disjonctives, ce qui entraîne que deux tâches doivent s'exécuter dans des intervalles de temps disjoints, les techniques existantes

(*) Manuscrit reçu Mars 1976, révisé Mai 1978.

(1) Institut de programmation. Université Paris VI.

sont défaillantes. Nous proposons d'y remédier dans le cas des disjonctions, nous nous sommes fixés comme objectif d'*optimiser la durée des ordonnancements comportants des contraintes disjonctives*.

Ce problème est difficile à résoudre en raison de son caractère fortement combinatoire: il est *NP-complet*; les spécialistes proposent des méthodes arborescentes. Notre méthode, par séparation et évaluation progressive, reprend la séparation sur les disjonctions de Balas [1], Tabourier [14] et Roy [12], et, utilise une fonction d'évaluation proche de celles de Bratley [3] et Florian [6] [7]. Elle est rendue très puissante par l'exploitation en chaque sommet de l'arborescence des propriétés communes de l'ensemble des solutions de ce sommet. Ceci permet d'améliorer grandement la fonction d'évaluation par défaut et de séparer efficacement, à l'aide, d'une part, d'une matrice des chemins de valeur maximale, et d'autre part des notions originales d'entrée et de sortie de clique de disjonction. Elle a été testée avec succès sur des problèmes d'ateliers de taille déjà importantes: 14 pièces usinées sur 4 machines différentes, par exemple.

II. FORMULATION MATHÉMATIQUE (Roy [13])

Tout au long de l'article, d_i désigne la durée de la tâche i , t_i sa date de début d'exécution.

Nous associons à tout problème d'ordonnement à contraintes potentielles un graphe conjonctif, à tout problème d'ordonnement à contraintes disjonctives un graphe disjonctif.

1. Graphe conjonctif

DÉFINITION : Un graphe conjonctif est un graphe valué $G = (X, \mathcal{U})$, comportant une seule entrée notée 1 et une seule sortie n , tel qu'il existe un chemin de valeur positive, entre l'entrée 1 et tout autre sommet, et, entre tout sommet différent de la sortie, n , et cette sortie.

On obtient X en ajoutant à l'ensemble des tâches, deux tâches fictives, 1 et n ; \mathcal{U} , en associant à chaque contrainte potentielle un arc valué.

Ensemble de potentiels

Un ensemble de potentiels sur un graphe conjonctif $G = (X, \mathcal{U})$ est une application t de X dans \mathbb{R} telle que, pour tout arc conjonctif (i, j) de valuation a_{ij} , on ait: $t_j - t_i \geq a_{ij}$. On pose habituellement: $t_1 = 0$.

Théorème d'existence

Une condition nécessaire et suffisante pour qu'il existe un ensemble de

potentiels sur un graphe conjonctif $G = (X, \mathcal{U})$ est que ce graphe n'ait pas de circuit absorbant, c'est-à-dire de circuit de valuation strictement positive.

Matrice des chemins de valeur maximale

A un graphe conjonctif $G = (X, \mathcal{U})$, on associe une matrice L , d'élément générique $l(i, j)$, où $l(i, j)$ est la valeur du chemin de valeur maximale allant de i à j .

2.-Problème disjonctif

Contrainte disjonctive

Une contrainte disjonctive $[i, j]$, entre deux tâches i et j , exprime le fait que i et j doivent s'exécuter dans des intervalles de temps disjoints. Si i précède j : $t_i + d_i \leq t_j$; si j précède i : $t_j + d_j \leq t_i$; on a donc : $t_j - t_i \geq d_i$ ou $t_i - t_j \geq d_j$.

De même qu'à une contrainte conjonctive correspond un arc conjonctif, à une contrainte disjonctive correspond un couple d'arcs disjonctifs : les arcs (i, j) et (j, i) valués par d_i et d_j .

Arbitrer une contrainte disjonctive $[i, j]$, c'est choisir un ordre d'exécution des tâches i et j , ce qui équivaut au choix de l'arc (i, j) ou bien de l'arc (j, i) et donc à l'insertion de la contrainte conjonctive correspondante.

Graphe disjonctif

Un graphe disjonctif est un graphe $\mathcal{G} = (X, \mathcal{U} \sqcup \mathcal{D})$, où $G = (X, \mathcal{U})$ est un graphe conjonctif et \mathcal{D} , un ensemble de contraintes disjonctives (fig. 1).

Un ensemble de potentiels sur un graphe disjonctif est une application t de X dans \mathbb{R} telle que l'on ait :

- pour toute contrainte conjonctive (k, l) : $t_l - t_k \geq a_{kl}$;
- pour toute contrainte disjonctive $[i, j]$: $t_j - t_i \geq d_i$ ou $t_i - t_j \geq d_j$.

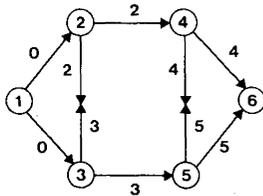


Figure 1. — Graphe disjonctif.

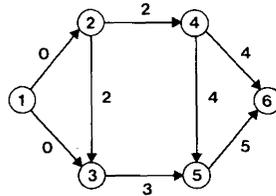


Figure 2. — Graphe conjonctif associé à \mathcal{A}_0 .

Arbitrage

Un arbitrage est un ensemble \mathcal{A} d'arcs disjonctifs; l'appartenance de (i, j) à \mathcal{A} impose d'exécuter la tâche i avant la tâche j .

A l'arbitrage \mathcal{A} , on associe le graphe conjonctif $G = (X, \mathcal{U} \sqcup \mathcal{A})$.

Un arbitrage est dit *complet* si toutes les contraintes disjonctives sont arbitrées.

Un arbitrage est dit *compatible* si le graphe conjonctif associé n'a pas de circuit absorbant.

On appellera *solution*, par abus de langage, un arbitrage complet compatible. La durée d'une solution est la valuation du chemin critique du graphe conjonctif associé.

Exemple : Considérons le graphe disjonctif de la figure 1; l'arbitrage $\mathcal{A}_0 = \{(2, 3), (4, 5)\}$ (fig. 2) est complet et compatible; c'est une solution; l'ensemble des potentiels minimaux associé est : $t_1 = t_2 = 0$; $t_3 = t_4 = 2$; $t_5 = 6$; $t_6 = 11$.

III. CLIQUES DE DISJONCTIONS

$\mathcal{G} = (X, \mathcal{U} \sqcup \mathcal{D})$ désigne un graphe disjonctif; \mathcal{A}_0 , un arbitrage complet compatible, \mathcal{A}_0 est donc une « solution »; L_0 , la matrice d'élément générique $l_0(i, j)$ où $l_0(i, j)$ est la valeur du chemin de valeur maximale allant de i à j dans le graphe $G_{\mathcal{A}_0} = (X, \mathcal{U} \sqcup \mathcal{A}_0)$.

1. Définition

Une *clique de disjonction* \mathcal{C} est un ensemble de tâches, de cardinal supérieur à 2, tel que deux quelconques d'entre elles soient en disjonction.

La durée $D(\mathcal{C})$ d'une clique est la somme des durées des tâches de la clique.

Dans un problème d'atelier, l'ensemble des tâches exécutées sur une machine forme une clique.

2. Ordre total sur \mathcal{C} associé à \mathcal{A}_0

Relation ρ

On écrit ipj s'il existe dans $G_{\mathcal{A}_0}$ un chemin de i à j , de valeur supérieure à d_i , ou bien si $i = j$.

LEMME : La relation ρ induit un ordre total.

Preuve : ρ est réflexive par définition.

ρ est antisymétrique : ipj , jpi et $i \neq j$ entraînent l'existence d'un circuit de $G_{\mathcal{A}_0}$, de valeur supérieure à $d_i + d_j$, donc absorbant, ce qui contredit la compatibilité de \mathcal{A}_0 .

ρ est transitive : démonstration analogue.

L'ordre ρ est total car \mathcal{A}_0 est un arbitrage complet.

Conséquence

\mathcal{C} a un plus petit élément, E , et un plus grand élément S . E sera appelé entrée de la clique \mathcal{C} et S , sortie.

THÉORÈME 1 : *Il existe dans $G_{\mathcal{A}_0}$ un chemin, de 1 à n , de valeur supérieure ou égale à :*

$$l_0(1, E) + D(\mathcal{C}) + l_0(S, n) - d_S.$$

Preuve : soit $C = \{x_1, x_2, \dots, x_c\}$ l'ensemble des éléments de \mathcal{C} , indicé dans l'ordre de ρ , avec : $x_1 \equiv E$ et $x_c \equiv S$. On a, par définition de ρ : $l_0(x_i, x_{i+1}) \geq d_{x_i}$ (on peut avoir l'inégalité stricte, par exemple, s'il existait dans G un chemin allant de x_i à x_{i+1}). D'où en sommant :

$$\sum_{x_i \in C - \{x_c\}} l_0(x_i, x_{i+1}) \geq \sum_{x_i \in C - \{x_c\}} d_{x_i} = D(\mathcal{C}) - d_S.$$

Il en résulte immédiatement que :

$$l_0(1, E) + \sum_{x_i \in C - \{x_c\}} l_0(x_i, x_{i+1}) + l_0(S, n) \geq l_0(1, E) + D(\mathcal{C}) + l_0(S, n) - d_S.$$

Le chemin associé est obtenu en concaténant le chemin de valeur maximale de 1 à E aux chemins de valeur maximale de x_i à x_{i+1} ($1 \leq i \leq c - 1$) puis au chemin de valeur maximale de $x_c = S$ à n .

IV. PROPRIÉTÉS DES SOLUTIONS DE DURÉE INFÉRIEURE A f_0 (Carlier [5])

Nous nous intéressons dans la suite, uniquement aux *solutions de durée inférieure à f_0* ; f_0 peut être la durée de la meilleure solution connue (obtenue par exemple heuristiquement). Afin d'évaluer efficacement par défaut la durée optimale, nous déterminons certaines propriétés communes à ces solutions : contraintes conjonctives sous-jacentes et arbitrages triviaux.

1. Fonction d'évaluation par défaut

Soit \mathcal{E}_1 (resp : \mathcal{S}_1) l'ensemble des éléments E (resp : S) tels qu'il existe une solution de durée inférieure à f_0 pour laquelle E (resp : S) est entrée (resp : sortie) de la clique \mathcal{C} .

Soit \mathcal{E} (resp : \mathcal{S}) un sous-ensemble de \mathcal{C} contenant \mathcal{E}_1 (resp : \mathcal{S}_1).

THÉORÈME 2 : $\text{Min}_{e \in \mathcal{E}} l(1, e) + D(\mathcal{C}) + \text{Min}_{s \in \mathcal{S}} (l(s, n) - d_s)$ est une évaluation par défaut de la durée optimale.

Preuve : Il suffit de montrer que ce nombre est plus petit que :

$$l_0(1, E) + D(\mathcal{C}) + l_0(S, n) - d_S$$

(Cf : Théorème 1); $E \in \mathcal{E}_1$ et $\mathcal{E}_1 \subseteq \mathcal{E}$ entraîne : $E \in \mathcal{E}$; d'où $l_0(1, E) \geq \text{Min}_{e \in \mathcal{E}} l_0(1, e)$; or $l_0(1, e) \geq l(1, e)$; par transitivité : $l_0(1, E) \geq \text{Min}_{e \in \mathcal{E}} l(1, e)$; on démontre de même que : $l_0(S, n) - d_S \geq \text{Min}_{s \in \mathcal{S}} (l(s, n) - d_s)$; de plus $D(\mathcal{C}) \geq D(\mathcal{C})$; le résultat est obtenu en sommant les trois inégalités.

2. Amélioration de l'évaluation

Plus faibles seront les cardinaux de \mathcal{E} et \mathcal{S} , meilleure sera l'évaluation du théorème 2; deux tests permettent d'enlever de \mathcal{E} et \mathcal{S} certains sommets; initialement on pose : $\mathcal{E} = \mathcal{S} = \mathcal{C}$.

Soient i et j deux éléments de \mathcal{C} , h et k deux éléments de X .

Test 1

Si il y a un chemin dans $G = (X, \mathcal{U})$ de i à j de valeur strictement supérieure à $-d_j$, poser : $\mathcal{E} = \mathcal{E} - \{j\}$ et $\mathcal{S} = \mathcal{S} - \{i\}$.

En effet si la disjonction entre i et j était arbitraire de telle sorte que j précède i , l'arc ainsi créé fermerait, joint au dit chemin, un circuit de valeur strictement positive, c'est impossible. Donc nécessairement i précède j : i ne pourra être sortie, ni j entrée; il en est ainsi par exemple quand l'arc disjonctif (i, j) a été choisi.

Test 2

Si $l(1, i) + D(\mathcal{C}) + \text{Min}_{s \in \mathcal{S} - i} (l(s, n) - d_s)$ est strictement supérieure à f_0 ; poser $\mathcal{E} = \mathcal{E} - \{i\}$;

Si $\text{Min}_{e \in \mathcal{E} - j} l(1, e) + D(\mathcal{C}) + l(j, n) - d_j$ est strictement supérieure à f_0 , poser $\mathcal{S} = \mathcal{S} - \{j\}$.

Si i était entrée (resp : j sortie), la durée de la solution serait plus grande que f_0 car on aurait des chemins de valeur supérieure à f_0 .

De fait $l(h, k)$ est une évaluation par défaut de $l_0(h, k)$; trois tests permettent d'affiner cette évaluation et d'introduire un arc conjonctif (h, k) ; cela est particulièrement utile quand $h=1$ ou $k=n$.

Test 3

Si la sortie S est déterminée pour la clique \mathcal{C} , introduire l'arc conjonctif $(1, S)$ valué par : $\text{Min}_{e \in \mathcal{E}} l(1, e) + D(\mathcal{C}) - d_S$;

Si l'entrée E est déterminée, introduire l'arc conjonctif (E, n) valué par : $D(\mathcal{C}) + \text{Min}_{s \in \mathcal{S}} (l(s, n) - d_s)$.

Ce dernier test est applicable lorsque le cardinal de \mathcal{E} est 1 ; il permet d'introduire une contrainte conjonctive entre E et n très contraignante pour les prédecesseurs de E .

Posons :

$$\Delta 1 = \text{Min}_{e \in \mathcal{E}} (l(1, e) + d_e), \quad \Delta 2 = \text{Min}_{s \in \mathcal{S}} l(s, n)$$

Test-4

Si i n'appartient pas à \mathcal{E} , introduire l'arc conjonctif $(1, i)$ valué par $\Delta 1$; si j n'appartient pas à \mathcal{S} , introduire l'arc conjonctif (j, n) valuée par $\Delta 2 + d_j$.

Test 5

Introduire, si elle n'existe pas, une contrainte conjonctive (h, k) valuée par $\text{Min} [l(h, i) + d_i + l(j, k), l(h, j) + d_j + l(i, k)]$.

3. Arbitrages triviaux

Deux tests détectent des disjonctions qui seront arbitrées dans le même sens pour toutes les solutions :

Test 6

Si $l(1, i) + d_i + l(j, n)$ est strictement supérieure à f_0 , (j, i) appartient à toutes les solutions de durée inférieure à f_0 s'il en existe.

Test 7

Si E (resp : S) a été déterminé pour \mathcal{C} , alors (E, i) (resp : (j, S)) appartient à toutes les solutions de durée inférieure à f_0 .

De plus si \mathcal{E} ou \mathcal{S} est vide, le problème n'a pas de solution.

V. RÉOLUTION D'UN PROBLÈME D'ATELIER

1. Définition des problèmes d'ateliers

p pièces doivent être usinées par m machines ; les machines sont chacune d'un type différent ; une pièce doit passer sur la machine 1 avant la machine 2, puis sur la machine 2 avant la machine 3 Un ordonnancement ω est un m -uplet $(01, 02, \dots, 0m)$ où $0q$ désigne l'ordre de passage des pièces sur la machine q ; on cherche l'ordonnancement de durée minimale.

A ce problème, on associe un graphe disjonctif, les sommets sont les $p \times m$

tâches d'usinage ; les contraintes conjonctives résultent du fait que le passage d'une pièce sur les différentes machines se fait dans un ordre imposé ; les contraintes disjonctives, du fait qu'il y a une machine de chaque type.

2. Exemple

Trois pièces doivent passer sur quatre machines ; les durées des tâches valent les arcs de la figure 3 ; afin d'expliquer le mécanisme de l'algorithme proposé, on cherche par exemple des ordonnancements de durée inférieure à 33. On va en trouver un seul ; ce sera l'optimum. La solution optimale serait obtenue directement par application de la méthode de construction d'une solution (Cf : VI).

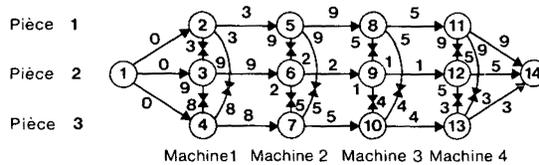


Figure 3

A chaque machine est associée une clique de trois éléments : l'ensemble des tâches exécutées par cette machine.

Le tableau ci-dessous permet d'appliquer les tests.

Clique \mathcal{C}	\mathcal{C}_1	\mathcal{C}_2	\mathcal{C}_3	\mathcal{C}_4
$D(\mathcal{C})$	20	16	10	17
$\text{Min}_{e \in \mathcal{C}} l(1, e) =$	0	3	11	12
$\text{Min}_{s \in \mathcal{C}} (l(s, n) - d_s) =$	8	6	3	0
Évaluation (Théorème 2)	28	25	24	29

Appliquons, en le justifiant, le test 2 ; si 13 est entrée de la clique \mathcal{C}_4 , alors : $l_0(13, n) \geq D(\mathcal{C}_4) + \text{Min}_{s \in \mathcal{C}_4} (l(s, n) - d_s)$; $l_0(13, n) \geq 17 + 0 = 17$. Or, $l_0(1, n) \geq l_0(1, 13) + l_0(13, n)$, donc, $l_0(1, n) \geq l(1, 13) + 17 = 17 + 17 = 34$. Ceci contredit le fait qu'on ne s'intéresse qu'aux solutions de durée inférieure à 33 ; 13 n'est pas entrée de la clique \mathcal{C}_4 . On montre de même que 11 n'est pas entrée de la clique \mathcal{C}_4 .

12 est donc entrée de la clique \mathcal{C}_4 ; nous ajoutons les contraintes conjon-

tives (12, 11) et (12, 13) valuées par 5 (Test 7); le test 3 fait passer la valuation de la contrainte conjonctive (12, 14) de 5 à 17.

Le test 6 montre que les autres arbitrages sont imposés :

[3, 4] est arbitrée dans le sens (3, 4); si on l'arbitrait dans le sens (4, 3), on créerait un chemin de valeur 37;

[6, 7] est arbitrée dans le sens (6, 7) (sinon, (7, 6) : 42);

[2, 3] est arbitrée dans le sens (2, 3) (sinon, (3, 2) : 35);

[5, 6] est arbitrée dans le sens (5, 6) (sinon, (6, 5) : 37);

[11, 13] est arbitrée dans le sens (11, 13) (sinon, (13, 11) : 41);

[9, 8] est arbitrée dans le sens (9, 8) (sinon, (8, 9) : 35);

[8, 10] est arbitrée dans le sens (8, 10) (sinon, (10, 8) : 34).

La figure 4 décrit l'unique solution de durée inférieure à 33; sa durée est 32; elle est optimale (On voit que $01 = 02$, $03 = 04$).

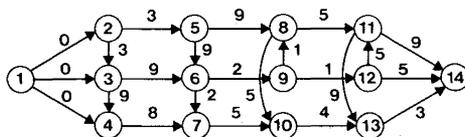


Figure 4

3. Discussion

Lorsque dans un problème d'atelier, on ne retient que les contraintes disjonctives entre les tâches d'une même machine, on s'intéresse à un problème plus facile quoiqu'encore *NP*-complet; Rinnooy Kan [10] le résoud pour calculer une borne inférieure du problème initial; Florian [7] abandonnait en plus les contraintes de précédence entre les tâches de la machine restante; pour ces deux auteurs, la borne de l'exemple ci-dessus est 29.

Dans notre méthode, avant d'appliquer le théorème 2, qui associe à une machine une borne inférieure rapide à calculer (de l'ordre de n opérations), on tient compte, par l'ajout de contraintes conjonctives de l'existence d'autres disjonctions et d'une borne supérieure; ces nouvelles contraintes résultent de l'application des tests; notre borne est 32.

Les tests nous ont d'autre part permis de prouver à la main l'optimalité de la meilleure solution connue (durée 1 165) du célèbre problème de Muth et Thompson [15] à 5 machines et 20 pièces.

VI. CONSTRUCTION D'UNE SOLUTION

Les tests sont d'autant plus puissants que f_0 est plus proche de l'optimum ; nous montrons sur un exemple comment construire une solution, puis nous proposons une méthode systématique.

1. Exemple pratique de Rinnooy Kan [10]

L'auteur propose à la page 136 de sa thèse un problème pratique formulé en termes d'ordonnancements à contraintes disjonctives et cumulatives. 35 pièces doivent être usinées successivement par les machines M1, M2, M3, M4, puis à nouveau par M2 ; les capacités des quatre machines sont

i	P_{i1}	P_{i2}	P_{i3}	P_{i4}	P_{i5}
1	4	4	8	2	6
2	2	2	8	1	4
3	1	1	8	1	2
4	—	2	8	,5	6
5	6	2	8	2	7
6	1	1	8	1	3
7	1	1	8	,5	4
8	—	5	8	,5	7
9	1	1	8	,5	4
10	—	—	8	2	6
11	1	1	8	,5	2
12	1	1	8	,5	2
13	1	1	8	1	2
14	2	1	8	1	4
15	—	5	8	1	9
16	7	10	8	2	16
17	—	—	—	1,5	6
18	1	1	8	1,5	6
19	—	—	8	2	8
20	6	6	8	3	6
21	—	—	—	2,5	12
22	—	—	8	2	7
23	—	—	—	2,5	11
24	2	2	8	1,5	3
25	—	—	—	—	4
26	6	3	8	2,5	7
27	—	—	8	1	5
28	1	1	8	1	3
29	—	—	—	1,5	6
30	1	1	8	1	2
31	1	1	8	1	2
32	—	3	8	1	7
33	6	6	8	3	7
34	—	6	8	1,5	6
35	4	2	8	1,5	10

respectivement 1, 6, ∞ et 1 ; les tâches sur M3 ne peuvent débuter qu'aux dates multiples de 8 (début de journée) ; deux fonctions économiques à minimiser sont proposées : la durée totale T , et, la somme Δ des temps libres sur M4

Pour les problèmes de grande taille, il existe un nombre considérable de solutions optimales, ce qui permet de prendre en compte une seconde fonction économique quand on connaît les caractéristiques communes de ces solutions.

2. Méthode systématique pour obtenir une première solution

Déterminer sur l'ensemble des disjonctions $[i, j]$ non arbitrées, l'une $[i, j]_0$ qui rende maximal :

$$M_{ij} = \text{Max} [l(1, i) + d_i + l(j, n), l(1, j) + d_j + l(i, n)].$$

Déterminer sur l'ensemble des tâches susceptibles d'être entrées de cliques, l'entrée e_0 de regret maximal, c'est-à-dire maximisant :

$$M_e = \text{Min}_{i \in \mathcal{E} - e} l(1, i) + D(\mathcal{C}) + \text{Min}_{s \in \mathcal{S}} (l(s, n) - d_s).$$

Déterminer la sortie s_0 de regret maximal N_{s_0} .

Poser : $T = \text{Max} (M_{e_0}, N_{s_0}, M_{[i,j]_0})$; si $M_{[i,j]_0} = T$, arbitrer la disjonction $[i, j]_0$ dans le sens opposé à celui créant un chemin de longueur T ; sinon soient : $\mathcal{E}_0 = \{e_0 / M_{e_0} = T\}$, $\mathcal{S}_0 = \{s_0 / N_{s_0} = T\}$, séparer sur l'ensemble \mathcal{E}_0 ou \mathcal{S}_0 de cardinal minimal; recommencer tant qu'il y a des arbitrages à effectuer.

VII. MÉTHODE ARBORESCENTE POUR TRAITER LE PROBLÈME GÉNÉRAL

Nous associons à chaque sommet d'une arborescence une matrice des plus longs chemins qui nous permet de détecter les circuits absorbants. Nous introduisons la notion de recouvrement pour appliquer le théorème 2 à plusieurs cliques et séparer efficacement.

1. Recouvrement

Une famille de cliques $\{\mathcal{C}_r\} r \in R$, est un recouvrement si pour toute contrainte disjonctive $[i, j]$ non arbitrée, il existe r tel que $\{i, j\} \subset \mathcal{C}_r$.

Recouvrement initial

Le recouvrement initial est l'ensemble des cliques maximales (au sens de l'inclusion).

2. Initialisations

Initialiser f_0 en construisant une bonne solution (VI-2). Calculer \mathcal{E}_r et \mathcal{S}_r pour chaque clique \mathcal{C}_r en appliquant les deux premiers tests.

Calculer la matrice L associée à G .

Utiliser la matrice L et les tests pour calculer une évaluation matricielle M de L_0 .

Fixer l'évaluation initiale par défaut, comme le maximum de $M(1, n)$ et de l'évaluation fournie par le théorème 2.

3. Séparation

Nous séparons, à chaque étape, l'espace des solutions en deux espaces : d'une part, les solutions pour lesquelles la tâche i précède la tâche j , d'autre part, les solutions pour lesquelles la tâche i suit la tâche j , d'où une arborescence pour laquelle, à chaque sommet, correspond un ensemble d'arcs disjonctifs, c'est-à-dire un arbitrage \mathcal{A} .

A chaque sommet, nous associons une matrice appelée, par abus de notation, L (initialement $L := M$ et L dépend évidemment de \mathcal{A}). L est une matrice de valuations maximales en ce sens que pour tout $i, j, k : l(i, j) + l(j, k) \leq l(i, k)$ (Inégalité triangulaire).

A chaque séparation, il faut calculer une nouvelle matrice des chemins de valeur maximale : celle-ci est obtenue en ajoutant un arc disjonctif (i, j) . L'algorithme VII-3 permet de n'effectuer en moyenne qu'un nombre d'opérations proportionnel au nombre n de sommets de X .

Dans le plus mauvais cas, ce nombre est toutefois multiple de n^2 .

Algorithme VII-3

- i) Calculer l'ensemble K des sommets k tels que : $l(k, i) + d_i > l(k, j)$.
- ii) Rechercher les sommets t tels que : $d_i + l(j, t) > l(i, t)$; un sommet t trouvé, reprendre les sommets k de K pour déterminer si : $l(k, i) + d_i + l(j, t) > l(k, t)$; s'il en est ainsi poser : $l(k, t) := l(k, i) + d_i + l(j, t)$.

4. Application des tests

La matrice L , nouvellement calculée, engendre un nouveau graphe disjonctif auquel on applique la théorie générale.

Amélioration du recouvrement

Si un ensemble $\mathcal{D}_r = \mathcal{E}_r$, ou \mathcal{S}_r , est de cardinal 1, on modifie la clique $\mathcal{C}_r : \mathcal{C}_r := \mathcal{C}_r - \mathcal{D}_r$.

THÉORÈME 3 : *Le nouveau recouvrement est meilleur que les précédents, en ce sens que l'évaluation par défaut qui lui est associée est plus grande.*

On le prouve en comparant la nouvelle évaluation à l'ancienne.

Calcul de \mathcal{E}_r et \mathcal{S}_r

On calcule \mathcal{E}_r et \mathcal{S}_r en appliquant les deux premiers tests.

Modification de L

On applique les tests.

5. Choix des disjonctions

A chaque clique \mathcal{C}_r du recouvrement correspondent les ensembles \mathcal{E}_r et \mathcal{S}_r . Si \mathcal{E}_r est de cardinal α et que l'on arbitre une contrainte disjonctive $[i, j]$ incluse dans \mathcal{E}_r , le cardinal de \mathcal{E}_r devient β , strictement inférieure à α . Nous nous rapprochons ainsi pas à pas du cardinal 0 ou 1; il s'ensuit l'inexistence de solutions ou des arbitrages triviaux.

L'heuristique de base est la suivante :

Arbitrer en priorité les disjonctions d'une clique privilégiée, celle dont l'évaluation initiale est la plus grande.

Choisir une contrainte disjonctive appartenant à des ensembles \mathcal{E}_r ou \mathcal{S}_r de cardinal minimal.

6. Problèmes de grande taille

L'utilisation d'une matrice à n^2 éléments est impraticable pour les problèmes de grande taille; dans ce cas, nous calculons seulement $l(1, i)$ et $l(j, n)$; nous avons procédé ainsi pour traiter manuellement l'exemple de Muth et Thompson.

VIII. RÉSULTATS

1. Programmation

La méthode proposée est tout à fait générale. Toutefois, pour l'expérimenter, il nous a paru intéressant de l'appliquer aux problèmes d'ateliers. Un programme FORTRAN a été écrit pour le terminal de l'UNIVAC 1108 d'Orsay à l'École Normale Supérieure de l'Enseignement Technique. Une première arborescence est décrite pour déterminer l'optimum à ordres identiques sur chaque machine; une deuxième arborescence est ensuite construite pour chercher l'optimum absolu. Dans cette deuxième arborescence, la fonction d'évaluation par excès est initialisée par $f_0 = f_e + 1$ (f_e optimum à ordres égaux).

2. Améliorations

Nous avons utilisé les particularités suivantes des problèmes d'ateliers pour adapter la méthode. Nous cherchons un ordonnancement optimal parmi les $\omega = (01, 02, \dots, 0m)$ tels que $01 = 02$ et $0m = 0m - 1$; le théorème de Roy [11] affirme qu'il en existe. Pour améliorer la fonction d'évaluation par défaut, nous calculons la durée des sous-problèmes d'ateliers, formés par les tâches référénçant 2 ou 3 machines consécutives, en traitant le sous-problème associé.

Nombre de machines	5	4	5	4	6	9	7	8
Nombre de pièces	11	13	12	14	10	8	7	8
Nombre de tâches	57	54	62	58	62	74	51	66
Nombre de disjonctions	275	312	330	364	270	252	144	226
Taille de la première arborescence	249	331	705	1 395	705	259	89	401
Nombre de solutions améliorantes rencontrées	6	17	20	31	7	9	2	4
Temps de calcul : t_1	45	48	110	185	103	68	16	60
Optimum à ordres égaux	7 038	7 166	7 312	8 003	7 720	8 505	6 590	7 819
Taille de la deuxième arborescence	49	41	49	267	1 317	1 185	373	4 501
Première évaluation	6 917	7 019	7 103	8 003	7 118	7 951	5 699	7 100
Nombre de solutions améliorantes	1	1	1	1	3	7	2	7
Temps de calcul : t_2	4	2	7	24	158	209	37	758
Optimum	7 038	7 166	7 312	8 003	7 702	8 313	6 558	7 649
Temps total : $t_1 + t_2$	49	50	117	209	261	277	53	818

3. Résultats

Le tableau résume le traitement de huit exemples ; le premier temps de calcul t_1 , est la durée des initialisations, c'est-à-dire du traitement de $m - 1$ problèmes de Johnson, de $m - 2$ problèmes à trois machines et du problème à ordres égaux ; le temps t_2 est la durée de description de la deuxième arborescence. Le temps machine est exprimé en secondes d'UNIVAC 1108.

4. Données

Nous rapportons ci-dessous les matrices des temps d'usinage.

Exemple 1 : 5 machines, 11 pièces.

375	632	12	460	528	796	532	14	257	896	532
12	452	876	542	101	245	230	124	527	896	302
142	758	124	523	789	632	543	214	753	214	501
245	278	534	120	124	375	896	543	210	258	765
412	398	765	499	999	123	452	785	463	259	988

Exemple 2 : 4 machines, 13 pièces.

654	321	12	345	678	963	25	874	114	785	203	696	302
147	520	147	586	532	145	24	517	896	543	210	784	512
345	789	630	214	275	302	142	24	520	336	699	855	221
447	702	255	866	332	225	589	996	541	234	784	512	345

Exemple 3 : 5 machines, 12 pièces.

456	789	876	543	210	123	456	789	876	543	210	124
537	854	632	145	785	214	752	143	698	532	145	247
123	225	588	669	966	332	144	755	322	100	114	753
214	528	896	325	147	856	321	427	546	321	401	245
234	123	456	789	876	543	210	123	456	789	876	543

Exemple 4 : 4 machines, 14 pièces.

456	789	630	214	573	218	653	214	204	785	696	532	12	457
856	930	214	257	896	532	142	547	865	321	124	12	345	678
963	21	475	320	124	752	147	532	145	763	214	257	854	123
696	320	142	753	214	528	653	214	527	536	214	528	888	999

Exemple 5 : 6 machines, 10 pièces.

333	333	252	222	255	555	558	888	889	999
991	111	222	204	477	566	899	965	588	889
996	663	222	114	123	456	789	876	543	210
123	456	789	876	543	210	124	537	854	632
145	785	214	752	143	698	532	145	247	451
234	532	586	532	142	573	12	14	527	856

Exemple 6 : 9 machines, 8 pièces.

887	799	999	666	663	333	222	114
447	779	999	666	25	558	886	541
234	567	852	140	222	558	965	412
159	267	483	753	420	159	25	863
201	478	520	145	699	875	633	222
555	444	120	142	578	965	412	25
463	123	456	789	876	543	210	123
456	789	630	258	741	36	985	214
753	21	427	520	142	534	157	896

Exemple 7 : 7 machines, 7 pièces.

692	581	475	23	158	796	542
310	582	475	196	325	874	205
832	14	785	696	530	214	578
630	214	578	214	785	236	963
258	147	852	586	325	896	325
147	753	2	356	565	898	800
255	806	699	877	412	302	120

Exemple 8 : 8 machines, 8 pièces.

456	789	654	321	456	789	654	789
654	123	123	456	789	654	321	147
852	369	632	581	472	586	320	120
145	678	965	421	365	824	758	639
632	581	475	32	536	325	863	21
425	396	325	147	852	12	452	863
214	123	456	789	654	321	456	789
654	789	654	123	123	456	789	654

REMERCIEMENTS

Je remercie Jean-Louis Laurière d'avoir bien voulu guider mes recherches.

Je remercie aussi MM. R. Faure, B. Lemaire, P. Tolla ainsi que les rapporteurs pour leurs corrections.

BIBLIOGRAPHIE

1. E. BALAS, Machine sequencing: disjunctive graph and degree constrained sub-graphs, IBM. *New York Scientific Center Technical Report*, n° 320-2971, April 1969.
2. E. BALAS, Machine sequencing via disjunctive graphs: an implicit enumeration algorithm. *Operations Research*, 17, n° 6, 1969.
3. P. BRATLEY, M. FLORIAN and P. ROBILLARD, On sequencing with earliest start and due dates with application to computing bounds for the $(n, m, G/F_{\max})$, Problem. *N.L.R.Q.*, 20, 57, 1973.
4. J. CARLIER, Disjonctions dans les ordonnancements. *R.A.I.R.O.*, vol. 2, 1975.
5. J. CARLIER, *Thèse de troisième cycle*, Paris VI, juin 1975.
6. M. FLORIAN, P. TREPANT and G. MC-MAHON, An implicit enumeration algorithm for the machine sequencing problems. *Management Science*, 17, 782; August 1971.
7. M. FLORIAN, C. TILQUIN and G. TILQUIN, An implicit enumeration algorithm for complex scheduling problems. *International Journal of Production Research*, 13, 1975, 1 (Jan), p. 25-40.
8. S.-B. JOHNSON, Discussion: sequencing n jobs on two machines with arbitrary time lags. *Management Science*, vol. 5, n° 3, 1960, p. 299.
9. J.-F. RAIMOND, Minimaximal paths on disjunctive graphs by direct search. *IBM Journal of research and development*, vol. 13, n° 4, July 1969.
10. (*) AHG RINNOOY KAN, *Machine scheduling problem. Classification complexity and computations*, Nijhoff, The Hague, 1976.
11. B. ROY, Cheminement et connexité dans les graphes; applications aux problèmes d'ordonnement. *METRA*, série spéciale, n° 1, 1962.
12. B. ROY, Prise en compte des contraintes disjonctives dans la méthode du chemin critique. *R.A.I.R.O.*, n° 38, 1966, p. 68-84.
13. B. ROY, *Algèbre moderne et théorie des graphes*. Tome II, Dunod, 1970.
14. Y. TABOURIER, Problèmes d'ordonnement à contraintes purement disjonctives. *R.A.I.R.O.*, Série verte, n° 3, 1969.
15. (*) J. F. MUTH and G. L. THOMPSON, *Industrial Scheduling*, Englewoods Cliffs, New Jersey, Prentice Hall.

(*) Référence parvenue après la rédaction de l'article et ajoutée à la correction.