

ABDELKADER LAHRICHI

Ordonnancements. La notion de « parties obligatoires » et son application aux problèmes cumulatifs

RAIRO. Recherche opérationnelle, tome 16, n° 3 (1982), p. 241-262

http://www.numdam.org/item?id=RO_1982__16_3_241_0

© AFCET, 1982, 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.
LA NOTION DE « PARTIES OBLIGATOIRES »
ET SON APPLICATION
AUX PROBLÈMES CUMULATIFS (*)**

par Abdelkader LAHRICHI ⁽¹⁾

Résumé. — Pour résoudre les problèmes cumulatifs, où le délai de réalisation est considéré comme le critère de choix le plus important, on introduit une nouvelle méthode basée sur la notion de « parties obligatoires ». Ses avantages sont :

- de ne pas débiter par une heuristique, mais par l'ordonnement définitif et « exact » de certaines parties de tâches bien déterminées;
- d'utiliser un minorant de l'intensité maximale bien plus proche de la solution exacte que l'indicateur généralement utilisé : l'intensité moyenne.

Les résultats numériques obtenus sur plus de cent problèmes réels confirment de façon décisive la supériorité des nouvelles méthodes proposées ici par rapport aux méthodes traditionnelles.

Mots clés : Ordonnement; minimax; cumulatif; heuristique; enveloppe.

Abstract. — In order to solve cumulative scheduling problems, where the completion time range, is considered as the main criterium, we introduce a new method based on the notion of "compulsory parts". Its advantages are:

- not to start with an heuristic, but with an exact and final scheduling of some well determined parts of tasks;
- to use a lower bound of the maximal intensity far closer to the exact solution than the indicator generally used: the average intensity.

The numerical results obtained from more than one hundred examples show clearly the superiority of the new methods over traditional methods.

Keywords: Scheduling; minimax; cumulative; heuristic; envelope.

INTRODUCTION

Dans la réalisation des projets, l'un des principaux soucis des responsables est le problème cumulatif : savoir comment faire cadrer un projet avec les disponibilités budgétaires, les ressources en personnel, etc.

Sur le marché français par exemple, les auteurs de [9] page 300 ont recensé 28 programmes d'ordonnement, dont 23 s'efforcent de tenir compte de ce problème. Dans cet article, notre but est de résoudre les problèmes cumulatifs à

(*) Reçu juillet 1979.

(¹) I.N.S.E.A. Rabat, Maroc et E.D.F., Division Statistiques, Optimisation et Reconnaissance des formes, services I.M.A. D.E.R., 1, avenue du Général-de-Gaulle, 92141 Clamart, France.

un seul moyen, où le délai de réalisation est considéré comme le critère de choix le plus important.

Dans le premier paragraphe, on rappelle quelques définitions sur les problèmes cumulatifs.

Dans le second, on présente une revue des différentes méthodes existant à ce jour, en mettant en évidence les principales difficultés rencontrées par celles-ci.

Dans la troisième, on introduit la notion de « parties obligatoires », ainsi qu'une nouvelle méthode présentant les avantages :

- de ne pas débiter par une heuristique, mais par l'ordonnancement définitif et exact de certaines parties de tâches bien déterminées;
- d'utiliser un minorant de l'intensité maximale bien plus proche de la borne exacte que l'indicateur généralement utilisé : l'intensité moyenne.

Dans le paragraphe 4, on fait la comparaison des résultats obtenus sur plus de 100 exemples réels, par l'ancien programme Planec [8] et par notre méthode. Les solutions trouvées, par notre méthode sont optimales dans 50 % des cas environ, et ne se sont jamais écartées de plus de deux unités du minorant.

Dans le paragraphe 5, on propose des extensions et une généralisation des parties obligatoires.

1. RAPPELS ([3, 4, 8, 10, 12])

1.1. Définitions

(1) On appelle *problème cumulatif*, tout problème d'ordonnancement soumis aux deux types de contraintes potentielles et cumulatives. Dans cette note, on se restreindra au cas où il existe un seul type de moyens. Si n est le nombre de tâches, chaque tâche i aura une durée d_i , une intensité M_i (nombre de moyens nécessaires à son exécution) et une charge $c_i = M_i d_i$.

La charge peut être assimilée à une quantité de travail. On peut la représenter schématiquement par un rectangle (cf. fig. 1).

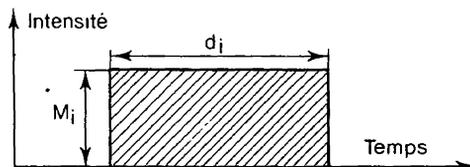


Figure 1

- La charge totale sera : $CHT = \sum c_i$, pour i variant de 1 à n .
- t_i^- , t_i^+ et m_i seront les dates de début au plus tôt et au plus tard, et la marge totale de la tâche i .

- T_c représentera le temps critique.

(2) Pour chaque ordonnancement réalisable (respectant les contraintes de précédence), on définit :

- t_i : début réel d'exécution de la tâche i ;
- T_r : durée réelle d'exécution de l'ensemble du projet ($T_r \geq T_c$).
- La fonction de charge : $I(k)$; k variant de 1 à T_r .

$I(k)$ est le nombre cumulé de moyens nécessaires durant la k -ième unité de temps.

- L'intensité maximale est : $I_{\max} = \max(I(k))$ pour $k \in [1, T_r]$.
- L'intensité moyenne est :

$$I_m = \left\lceil \frac{CHT}{T_r} \right\rceil;$$

($\lceil \cdot \rceil$: partie entière supérieure).

- L'écart partiel durant la k -ième unité de temps est : $e_k = |I(k) - I_m|$.
- L'écart total est : $E = \sum e_k$ pour $k \in [1, T_r]$.

1.2. Quelques problèmes cumulatifs.

(1) PROBLÈME 1 : Trouver un ordonnancement réalisable minimisant l'intensité maximale I_{\max} sous la contrainte $T_r = T_c$.

(2) PROBLÈME 2 : Trouver un ordonnancement réalisable minimisant l'écart total E sous la contrainte $T_r = T_c$.

(3) PROBLÈME 3, [1] : Trouver un ordonnancement réalisable minimisant le temps réel T_r sous la contrainte $I_{\max} \leq M$, M étant une constante donnée.

Ces trois problèmes sont les plus représentatifs parmi les problèmes cumulatifs à un seul type de moyens. Dans ce qui suit on s'intéressera plus particulièrement au problème 1.

2. ANALYSE CRITIQUE DES MÉTHODES EXISTANTES

2.1. Méthodes sérielles ([2, 10])

2.1.1. Généralités

La majorité des méthodes commence par résoudre le problème central de l'ordonnancement ([6, 10]) et établir la fonction de charge au plus tôt ($t_i = t_i^-$

pour i variant de 1 à n). Cela permet d'obtenir des bornes pour les solutions optimales des problèmes cités ci-dessus. Considérons l'exemple 1 suivant, représenté par son graphe potentiel-étape (fig. 2) :

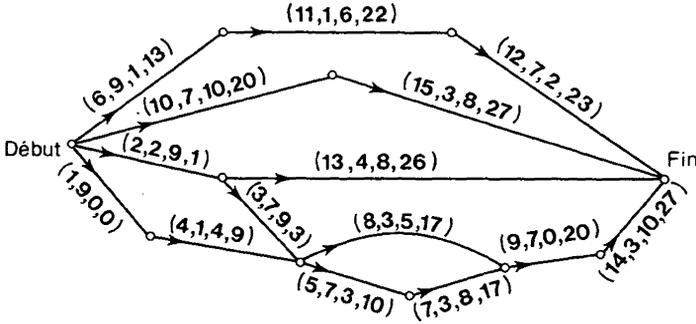


Figure 2

Chaque arc i comporte quatre valeurs : son numéro i , sa durée d_i , son intensité M_i et sa date au plus tard t_i^+ : (i, d_i, M_i, t_i^+) .

La fonction de charge au plus tôt est représentée sur la figure 3.

Les valeurs prises par la fonction de charge au plus tôt sont : (20, 20, 28, 28, 28, 28, 20, 18, 18, 18, 10, 10, 10, 5, 5, 5, 5, 8, 8, 8, 0, 0, 0, 0, 0, 0, 10, 10, 10). La charge totale (CHT) est égale à 330, le temps critique T_c est égal à 30; d'où :

$$I_m = \left\lceil \frac{330}{30} \right\rceil = 11.$$

Si I_{\max}^* , E^* et T_r^* sont les solutions optimales des problèmes 1, 2 et 3; et si I_{\max}^0 et E^0 sont les valeurs de l'intensité maximale et de l'écart total relatifs à la fonction de charge au plus tôt, on a alors :

$$11 = I_m \leq I_{\max}^* \leq I_{\max}^0 = 28$$

et :

$$E^* \leq E^0,$$

T_c minore T_r^* . Dans le cas où M est strictement inférieur à I_m , on peut donner un meilleur minorant pour T_r^* :

$$T_c < \left\lceil \frac{CHT}{M} \right\rceil \leq T_r^*.$$

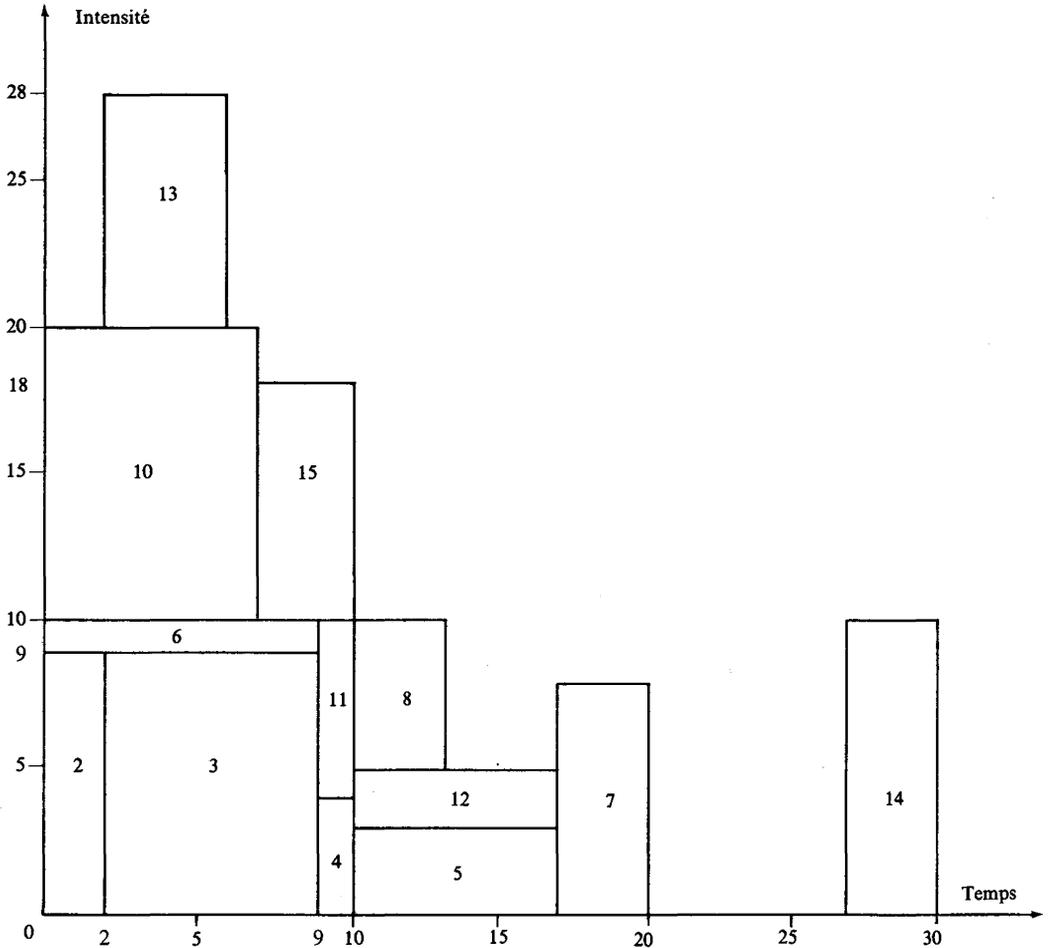


Figure 3

2.1.2. La méthode Milord [2] pour résoudre le problème 1

(1) *Principe* : Cette méthode se base sur le fait que : la solution du problème 1 est le plus petit entier compris au sens large entre l'intensité moyenne et l'intensité maximale de la fonction de charge au plus tôt, pour lequel la solution optimale du problème 3 est : $T_r = T_c$. Avec les données de l'exemple 1, par exemple, on a vu ci-dessus que : $11 \leq T_{\max}^* \leq 28$. Donc, pour trouver la solution du problème 1, on cherche le plus petit entier M , compris entre 11 et 27 pour lequel le problème 3 donne une solution satisfaisant $T_r = T_c$.

(2) *La méthode Milord pour résoudre le problème 3*

(a) On résout le problème central de l'ordonnancement.

(b) *Liste d'appel* : On fait un tri de toutes les tâches suivant les dates au plus tard croissantes. Si deux tâches ont leurs dates au plus tard égales, on les départage suivant leurs marges totales croissantes, etc. Remarquons que ce tri est un tri *topologique* (ordre total respectant l'ordre partiel du début) : s'il existe la contrainte de précédence $i < j$, alors dans cette liste on aura encore i placée avant j .

(c) *Affectation des tâches* : On prend les tâches dans l'ordre de la liste d'appel et on les affecte dès que les moyens restants disponibles le permettent.

(3) *Application sur l'exemple 1*

(a) Les résultats de la résolution du problème central sont illustrés sur la figure 3 et l'exemple 1.

(b) Nous avons numéroté les arcs suivant les dates au plus tard croissantes. Donc, la liste d'appel est : 1, 2, 3, . . . , 15.

(c) Affectation des tâches

Cas où $11 \leq M < 18$

Dans ce cas, les 8 premières tâches se placent à leurs dates au plus tôt. La fonction de charge partielle prend les valeurs : (10, 10, 10, 10, 10, 10, 10, 10, 10, 4, 8, 8, 8, 3, 3, 3, 3, 8, 8, 8, 0, 0, . . . 0). La tâche 10 aura pour début 20. Car, si elle commençait plus tôt, l'effectif maximal du vecteur de charge dépasserait strictement 17. Le fait d'imposer à la tâche 10, un début réel égal à 20, revient à imposer à la tâche 15, un début réel supérieur ou égal à sa date au plus tard : 27.

Pour satisfaire la contrainte $T_r = T_c = 30$, on devrait alors imposer à la tâche 15 comme début réel la valeur 27. Et on devrait faire de même pour la tâche 14 (tâche critique). Donc pour satisfaire la contrainte $T_r = T_c$, la décision d'imposer à la tâche 10, un début réel égal à 20, nous oblige à faire les deux tâches 15 et 14 en même temps, ce qui nous demanderait une intensité de 18 durant l'intervalle de temps]27, 30]. On en déduit que pour les cas : $11 \leq M < 18$, la méthode Milord ne peut trouver de solution satisfaisant la contrainte $T_r = T_c$.

Cas où $M = 18$

Dans ce cas la méthode Milord trouvera une solution satisfaisant la contrainte $T_r = T_c$; et-cela consiste à adopter l'ordonnancement réalisable suivant :

– les tâches 10, 11, 12 et 15 auront respectivement comme débuts réels les dates 9, 16, 17 et 16.

– et toutes les autres tâches commenceront à leurs dates de début au plus tôt (il est aisé de vérifier ce résultat en s'aidant de la figure 1).

Pour le problème 1, la solution proposée par la méthode Milord serait d'adopter l'ordonnement réalisable ci-dessus, avec une intensité maximale égale à 18

2.1.3. L'ancienne méthode Planec [11]

(1) La seule différence notable entre cette méthode et la méthode Milord réside dans l'élaboration de la liste d'appel. Cette liste se fait en deux étapes :

— on fait un premier tri topologique de toutes les tâches suivant les niveaux croissants;

— on affecte à chaque tâche i , une valeur $k_i = m_i / M_i d_i$. Puis au sein de chaque niveau, on réordonne les tâches suivant leurs k_i croissants.

Ainsi, pour l'exemple 1, la liste d'appel définitive serait :

	1	2	3	4	5	6	7	8	9	10	11	12	13
Liste	2	10	6	4	3	13	15	11	5	8	12	7	14
k_i	$\frac{1}{18}$	$\frac{20}{70}$	$\frac{13}{9}$	0	$\frac{1}{63}$	$\frac{24}{32}$	$\frac{20}{24}$	$\frac{13}{6}$	0	$\frac{7}{15}$	$\frac{13}{14}$	0	0
Niveau	1	1	1	2	2	2	2	2	3	3	3	4	6

Comme nous le verrons au paragraphe 3, nous adopterons partiellement cette liste, car à notre avis elle est de loin la plus élaborée des listes que nous connaissons. Elle exploite à la fois les quatre idées heuristiques suivantes :

— *le critère d'intensité* : il est évident qu'il faut placer en priorité les tâches de grandes intensités : une tâche de grande intensité mal placée a toutes les chances d'augmenter démesurément l'intensité maximale du tableau de charge;

— *le critère de durée* : les tâches de courtes durées ont de fortes chances de trouver des « créneaux » pour se placer même si on les laisse pour la fin. Donc il vaut mieux commencer par les tâches de longues durées;

— *le critère de marge* : de même, on doit placer le plus tôt possible les activités de faibles marges, car on a peu de choix pour les emplacements de ces tâches;

— *le critère des niveaux* : placer les tâches suivant les niveaux croissants (ou décroissants) nous permet :

- d'éviter de placer une tâche avant une ou plusieurs de ses antécédents,
- et surtout, de calculer très facilement les modifications sur certaines dates et certaines marges entraînées par le recul éventuel de cette tâche.

(2) *Application sur l'exemple 1*

Pour résoudre le problème 1, l'ancienne méthode Planec procède de la même manière que la méthode Milord : elle cherche le plus petit entier M pour lequel le problème 3 a une solution satisfaisant la contrainte $T_r = T_c$.

Cas où $11 \leq M < 19$

Pour résoudre un problème 3 avec un M compris entre 11 et 18, l'ancienne méthode Planec imposerait aux quatre premières tâches de la liste d'appel (tâches 2, 10, 6, 4) de débiter respectivement aux instants 0, 2, 0, 9. A ce stade, les valeurs prises par la fonction de charge partielle seraient :

$$(10, 10, 11, 11, 11, 11, 11, 11, 11, 4, 0 \dots 0).$$

Maintenant, pour que le maximum de la fonction de charge ne dépasse pas strictement 18, il faudrait imposer à la cinquième tâche de la liste d'appel (tâche 3), un début réel supérieur ou égal à 9. Comme sa date de début au plus tard est 3, on transgresserait la contrainte $T_r = T_c$.

Donc l'ancienne méthode Planec ne peut fournir de solution au problème 3, satisfaisant à la fois : $10 < M < 19$ et $T_r = T_c$. On vérifie aisément qu'il en est de même pour $M = 19$.

Cas où $M = 20$

Dans ce cas, l'ancienne méthode Planec donne une solution satisfaisant la contrainte $T_r = T_c$. Cela consiste à imposer aux tâches 8, 11, 12, 13 et 15 de débiter aux instants 12, 11, 12, 7 et 9 et à placer toutes les autres tâches à leurs dates de débuts au plus tôt.

Donc pour le problème 1, la méthode Planec propose l'ordonnancement réalisable ci-dessus, avec une intensité maximale égale à 20.

2.1.4. *Critique des méthodes sérielles*

(1) Ces méthodes débutent avec *une fonction de charge nulle* et placent les tâches suivant leur ordre dans la liste d'appel. Cela présente un grand risque : les premières tâches placées définitivement (heuristiques de type glouton [6], ch. 11, § 2), peuvent être mal affectées et occuper une bonne partie des meilleurs emplacements d'autres tâches encore non affectées.

C'est d'ailleurs ce qui s'est produit pour l'exemple précédent :

– pour $M \geq 11$, la méthode Planec place la tâche 10 (2^e tâche de la liste d'appel) dans l'intervalle]2, 9]. Ainsi, on ne peut plus débiter la tâche 8 avant l'instant 9, sans que l'intensité maximale de la fonction de charge ne dépasse 20. Dans ce cas, la tâche 10 a occupé entièrement l'emplacement idéal de la tâche 8 : l'intervalle]2, 9].

— comme on le verra au paragraphe 3, le meilleur emplacement de la tâche 10 est l'intervalle]10, 17]. Or pour $11 \leq M \leq 17$, la méthode Milord place la tâche 8 dans l'intervalle]10, 13]. En occupant partiellement la « place idéale » de la tâche 10, la tâche 8 rejette cette dernière dans l'intervalle]20, 27].

(2) Les solutions proposées par ces méthodes dépendent essentiellement de la liste d'appel choisie. Parmi les critères élémentaires suivant lesquels ces listes sont établies, citons :

- dates de début au plus tôt ou au plus tard croissantes ou décroissantes;
- niveaux croissants ou décroissants, etc.

A notre avis, puisque l'optimisation s'effectue sur l'intensité maximale en fonction du temps, une liste d'appel idéale serait établie en fonction des *intensités décroissantes*.

Parmi les raisons pour lesquelles aucune méthode sérielle n'utilise cette liste citons :

- placer la tâche de plus grande intensité avec une fonction de charge initiale nulle, revient pratiquement à affecter cette tâche au hasard;
- contrairement aux listes d'appels courantes, la liste d'appel suivant les intensités décroissantes n'est pas un tri topologique.

Ainsi elle s'adapte très mal aux heuristiques gloutonnes :

- à la fin de chaque phase i , on doit remettre à jour toutes les dates de début au plus tôt ou au plus tard des tâches ayant une relation d'antériorité avec la i -ième tâche de la liste d'appel.

(3) Ces méthodes utilisent un minorant — intensité moyenne — qui dans la pratique reste très loin de la solution proposée.

Malgré tous ces inconvénients, il est rare qu'un problème cumulatif soit résolu par une méthode non sérielle.

2.2. La méthode « Molos » pour résoudre le problème 2 [9]

2.2.1. Principe

Nous allons exposer brièvement le principe sur l'exemple 2 déduit de l'exemple 1 du § 21 comme suit : dans l'exemple 1, les tâches 5 et 7 avaient pour intensité les valeurs 3 et 8. Pour l'exemple 2, ces deux tâches auront respectivement les intensités suivantes : 0 et 5 (la nécessité de cette variante du premier exemple du § 21, est que l'exemple 2 donne une solution idéale pour le problème 2 avec $E^* = 0$ (Remarque 3 du § 4); par contre l'exemple 1 est tel que : $E^* > 0$ car $I_{\max} > I_m$, § 3.3).

Avec ces deux modifications (sur les intensités), les résultats de la résolution du problème central (2.1.1) restent valables. Les nouvelles valeurs prises par la fonction de charge au plus tôt sont :

$$(20, 20, 28, 28, 28, 28, 20, 18, 18, 18, 7, 7, 7, 2, 2, \\ 2, 2, 5, 5, 5, 0, 0, 0, 0, 0; 0, 0, 10, 10, 10).$$

L'intensité moyenne est de 10 et l'écart total de la fonction de charge au plus tôt est de 252.

Pour éviter le risque de débiter avec une fonction de charge nulle (méthodes sérielles), cette méthode débutera avec une fonction de charge « pleine » : la fonction de charge au plus tôt. Comme le but est de lisser, la méthode Molos se propose de combler les trous (créneaux) à partir de la droite. Pour notre exemple, il y a un « vide » dans l'intervalle]20, 27] et la tâche la plus efficiente pour combler ce vide est la tâche 10. La méthode Molos imposera *définitivement* à cette tâche de s'exécuter dans l'intervalle]20, 27] (heuristique gloutonne [4]).

Au début l'écart total était de 252; après cette première amélioration, l'écart total est de : $252 - (2 \times 70) = 112$.

Comme nous l'avons fait remarquer pour la méthode Milord, la décision d'imposer à la tâche 10, de débiter à l'instant 20, revient à imposer aux deux tâches 14 et 15, de s'exécuter en même temps dans l'intervalle]27, 30]. Donc dans l'intervalle]27, 30], il y aura un écart d'au moins $(18 - 10) \times 3 = 24$. S'il y a une « bosse » d'au moins 24, dans cet intervalle, il y aura un trou d'au moins 24 dans l'intervalle]0, 27]. Donc cette première décision pour la tâche 10, implique que l'écart total que propose la méthode Molos soit d'au moins : $2 \times 24 = 48$. Or nous verrons au paragraphe 3 que ce problème a une solution idéale avec un écart total nul.

2.2.2. Critique de la méthode « Molos »

Le fait de débiter avec une fonction de charge « pleine » ne nous permet pas d'échapper aux difficultés des méthodes sérielles (fonction de charge initiale nulle) : des tâches de charges relativement grandes peuvent se placer mal et de façon *définitive*. C'est ce qui s'est produit dans l'exemple précédent pour la tâche 10. Il y a un autre risque : dans la pratique, les problèmes peuvent comporter des milliers de tâches. Supposons que l'on veuille combler un créneau par la tâche i . En général, cardinal de $\hat{\Gamma}_{(i)}$ (tâches précédées par i) est de l'ordre de quelques centaines, et on ne peut décider pour la tâche i sans avoir trouvé une bonne répartition de $\hat{\Gamma}(i)$. On risque ainsi de se trouver face à un problème combinatoire inextricable.

3. LA NOTION DE « PARTIE OBLIGATOIRE » ET SON APPLICATION

Dans ce paragraphe, on s'impose la contrainte : $T_r = T_c$.

3.1. Définition [8]

Pour tout ordonnancement réalisable vérifiant la contrainte $T_r = T_c$, on a :

Si une tâche i , d'intensité M_i est telle que : $t_i^+ < t_i^- + d_i (\Leftrightarrow m_i < d_i)$, alors dans l'intervalle de temps $]t_i^+, t_i^- + d_i]$, nous avons obligatoirement M_i machines occupées avec la tâche i . Le nombre $M_i(t_i^- + d_i - t_i^+)$ sera dit « la charge obligatoire » de la tâche i (fig. 4).

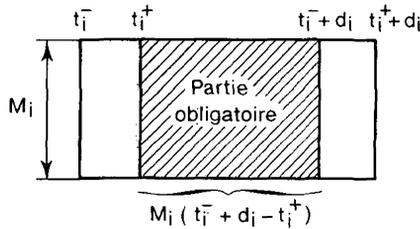


Figure 4. — Définition de la notion de partie obligatoire.

3.2. Théorème

La partie obligatoire de la fonction de charge partielle, obtenue en cumulant les parties obligatoires de toutes les tâches, est l'enveloppe inférieure de l'ensemble des fonctions de charge de tous les ordonnancements réalisables vérifiant la contrainte : $T_r = T_c$.

Démonstration :

Notons : $I^-(k)$ = partie obligatoire de la fonction de charge durant la k -ième unité de temps :

– il est évident que $I^-(k) \leq I_j(k)$ pour tout $k \in [1, T_c]$, et pour tout ordonnancement réalisable θ_j .

– il reste donc à prouver que pour tout $k \in [1, T_c]$, il existe un ordonnancement réalisable θ_k tel que $I^-(k) = I_k(k)$. Cette propriété est vérifiée par l'ordonnancement réalisable suivant :

- on place au plus tôt, toutes les tâches ayant une fin au plus tôt inférieure ou égale à $k-1$, et toutes les tâches ayant une partie obligatoire dans $[k-1, k]$.
- on place au plus tard toutes les tâches restantes, (toutes ces dernières tâches ont un début au plus tard supérieur ou égal à k).

3.3. Utilisation de la notion de partie obligatoire

A partir de cette notion, nous allons exposer une méthode qui présente les avantages suivants :

— *on ne débute pas avec une heuristique* : pour les problèmes 1 et 2, on a : $T_r = T_c$. En utilisant le théorème du § 3.2, on place de façon définitive et *exacte* la partie obligatoire de la fonction de charge. Cela permet de prévoir d'avance les zones surchargées.

Pour l'exemple 1, on pourrait schématiser la partie obligatoire de la fonction de charge par la figure 5 :

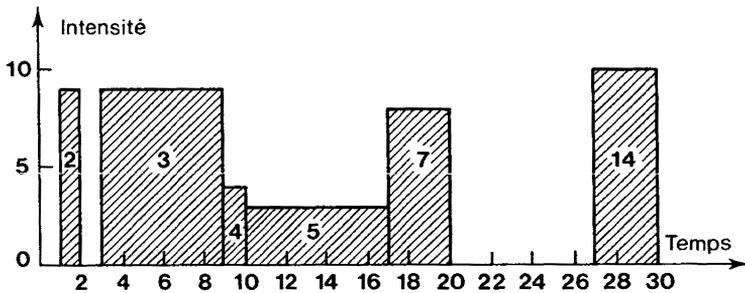


Figure 5

— *on utilise un minorant de l'intensité maximale optimale du problème 1 bien plus proche de la borne exacte que l'indicateur employé précédemment : l'intensité moyenne.*

A partir de la figure 5, étudions tous les emplacements possibles pour la tâche 10 (tâche de plus grande intensité $M_{10} = 10$) :

— $t_{10} = 10$: Dans ce cas la fonction de charge partielle devient : (0, 9, 0, 9, 9, 9, 9, 9, 9, 9, 4, 13, 13, 13, 13, 13, 13, 13, 13, 8, 8, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10, 10, 10). On peut déduire que 13 minore l'intensité maximale de tout ordonnancement réalisable satisfaisant : $t_{10} = 10$;

— $t_{10} = 20$: Comme on l'a vu au 2.1.2.3 c) imposer à la tâche 10 de débiter à l'instant 20, revient à imposer à la tâche 15 de débiter à l'instant 27, donc à s'exécuter en même temps que la tâche 14 dans l'intervalle]27, 30].

Dans cette optique la fonction de charge partielle serait : (0, 9, 0, 9, 9, 9, 9, 9, 9, 4, 3, 3, 3, 3, 3, 3, 3, 8, 8, 8, 10, 10, 10, 10, 10, 10, 10, 18, 18, 18). On peut déduire que 18 minore l'intensité maximale de tout ordonnancement réalisable satisfaisant : $t_{10} = 20$.

– $t_{10} \neq 10$ et $t_{10} \neq 20$: Dans ce cas, il est aisé de voir que l'intensité maximale de tout ordonnancement réalisable dépasse 14.

On peut conclure que 13 est un minorant du problème 1. Les autres méthodes avaient comme minorant 11 (intensité moyenne) et se déroulaient inutilement en vue d'obtenir des solutions avec une intensité maximale $M=11$ et $M=12$.

3.4. Heuristique 1, basée sur la notion de parties obligatoires

Le but principal de cette heuristique est de résoudre le problème 1.

(a) *Initialisation* :

- on résout le problème central Θ
- on dresse la liste d'appel en ordonnant les tâches suivant les intensités décroissantes;
- $m^* = I_m$ (intensité moyenne);
- $i=0$.

(b) *Tant que $i < n$ répéter* :

- $i \leftarrow i+1$;
- $j \leftarrow i$ -ième tâche de la liste d'appel.

S'il existe des dates $t \in [t_j^-, t_j^+]$ telles que le maximum de la nouvelle partie obligatoire de la fonction de charge partielle (obtenue en imposant à la tâche j de commencer à l'un de ces instants) est inférieur à m^ ; imposer définitivement à t_j le premier de ces instants.*

Si non, imposer définitivement à t_j le premier instant $t \in [t_j^-, t_j^+]$ tel que le maximum \bar{m} de la partie obligatoire de la fonction de charge résultante soit minimal;

- $m^* \leftarrow \bar{m}$.

Recalculer les dates de fin au plus tard pour toutes les tâches antérieures à la tâche j et les dates de début au plus tôt pour toutes les tâches postérieures à la tâche j (ce calcul peut être réduit car il n'est pas nécessaire de recalculer les dates pour toutes les tâches).

Fin : l'intensité maximale de la solution proposée par l'heuristique 1 est m^* .

REMARQUE 1 : La détermination de t_j tel que \bar{m} soit minimal à la fin de l'étape b peut se faire par un algorithme linéaire, tel que celui proposé dans [7].

REMARQUE 2 : A la fin de la première phase ($i=1$) m^* indique notre minorant pour le problème 1. Évidemment, on peut obtenir d'autres minorants en plaçant simultanément tous les couples de tâches possibles, ou tous les triplets, etc; ou

bien encore en plaçant en priorité la tâche de plus grande charge, ou de plus grande durée, ou celle dont le déplacement génère le plus grand « volume » de parties obligatoires, etc.

Application de l'heuristique 1 sur l'exemple 1 :

– les résultats de la résolution du problème central sont illustrés dans le graphe de l'exemple 1 et dans la figure 1;

– la liste d'appel suivant les intensités décroissantes est : (10, 3, 2, 13, 15, 11, 8, 12, 6);

– $m^* = 11$;

$i = 1, j = 10$ on va placer la tâche 10. Comme on l'a vu en 3.1, cette tâche sera affectée définitivement dans l'intervalle $]10, 17]$ avec cela, m^* prend la valeur 13;

$i = 2, j = 3$ on va placer la tâche 3. Elle se placera à sa date au plus tôt, sans qu'on ait à changer m^* . Imposer à la tâche 3 de débiter à sa date au plus tôt, revient à imposer à la tâche 2 de débiter aussi à sa date au plus tôt;

$i = 3$, cf. le cas $i = 2$.

La solution finale que proposerait l'heuristique 1 serait d'adopter l'ordonnancement réalisable suivant :

– les tâches 2, 3, 4, 5, 6, 7, 11, 14 commenceront à leurs dates au plus tôt;

– les tâches 8, 10, 12, 13 et 15 commenceront respectivement aux instants 17, 10, 20, 20, 24. La fonction de charge relative à cet ordonnancement prend les valeurs : (10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 13, 13, 13, 13, 13, 13, 13, 13, 13, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10). Le maximum de cette fonction (13) est égal à notre minorant. Donc l'heuristique 1 donne une solution optimale pour l'exemple 1.

REMARQUE 3 : Pour résoudre le problème 1 sur l'exemple 2, l'heuristique 1 donne le même ordonnancement réalisable cité ci-dessus. Ainsi la fonction de charge prend la valeur 10 dans tout l'intervalle $]0, 30]$. Là aussi l'optimum est atteint pour le problème 1. Il l'est aussi pour le problème 2 puisque l'écart total est nul.

REMARQUE 4 : La liste d'appel de l'heuristique 1 n'est pas un tri topologique. Cela rend l'utilisation de cette méthode très coûteuse (2.1.4.2), du fait que l'on doit recalculer les dates au plus tôt ou au plus tard d'un grand nombre de tâches, à chaque itération.

Dans la pratique, nous avons adopté la variante suivante :

3.5. L'heuristique 2 pour résoudre le problème 1

L'heuristique 2 consiste à appliquer l'heuristique 1 pour placer la tâche de plus grande intensité, et à continuer avec cette heuristique tant que les deux conditions suivantes restent vraies :

(C1) $i \leq 4$ (i : nombre de tâches déjà placées);

(C2) la charge de la partie obligatoire de la fonction de charge reste inférieure à la moitié de la charge totale.

Puis on applique toujours l'heuristique 1 mais avec la liste d'appel définie par Planec (cf. 2.1.3) pour les tâches restantes.

Comme on l'a vu dans la remarque 4, les itérations de l'heuristique 1 sont très coûteuses. La condition C1 nous évite de faire plus de quatre de ces itérations.

Illustrons le fonctionnement de l'heuristique 2 sur l'exemple 1.

La tâche 10 (de plus grande intensité) se placera dans l'intervalle [10, 17]. A ce stade la fonction de charge partielle serait : 0, 9, 0, 9, 9, 9, 9, 9, 9, 4, 13, 13, 13, 13, 13, 13, 8, 8, 0, 0, 0, 0, 0, 0, 10, 10, 10). La charge partielle de la fonction de charge est 212. Elle est supérieure à la moitié de la charge totale (330). On dresse la liste d'appel Planec sur les tâches restantes et on applique l'heuristique 1 avec cette liste. On trouve la même répartition que l'heuristique 1.

L'heuristique 2 peut aussi se justifier intuitivement de la façon suivante :

Après avoir placé la tâche 10, la partie obligatoire de la fonction de charge est de 212 et la charge totale est de 330. On a ainsi placé pratiquement les 2/3 des tâches. Les tâches, ou *parties de tâches*, encore non placées sont de charges relativement faibles et ont peu de chances de créer de « grandes bosses ».

REMARQUE 5 : Pour résoudre le problème 1 sur l'exemple 2, l'heuristique 2 donne aussi l'ordonnement réalisable, idéal proposé par l'heuristique 1.

4. RÉSULTATS NUMÉRIQUES

4.1. Le programme Planec ([5, 11])

Planec est un programme (FORTRAN+assembleur) de quelques 10.000 cartes et plus de 300 sous-programmes ⁽²⁾.

Parmi les multiples contraintes dont il peut tenir compte, citons : les contraintes de succession, les exclusives [3], les disjonctives, etc.

Le nombre maximal de tâches dont il peut tenir compte est de 1024.

4.2. Résultats obtenus sur quatre exemples réels

4.2.1. Dans le tableau I, nous comparons les performances de l'ancienne méthode Planec et de l'heuristique 2. Les exemples traités correspondent à la résolution du problème 1 pour l'ordonnement des travaux de révision de quatre centrales électriques.

⁽²⁾ Sur IBM 370/168 de l'E.D.F. Clamart.

TABLEAU I

		Problème			
		Pb1	Pb2	Pb3	Pb4
Nombre de tâches.		997	981	942	978
Nombre de contraintes de succession.		2385	2978	3417	4935
Temps d'exécution du problème central en secondes. . .		0,26	0,29	0,31	0,34
Minorant	Ancien Plan nec.	8	14	12	23
	Heuristique 2.	10	29	17	26
Solution	Ancien Plan nec.	17	37	21	34
	Heuristique 2.	11	29	17	28
Temps d'exécution	Ancien Plan nec.	6,45	9,82	7,93	6,05
	Heuristique 2.	1,34	0,97	0,91	1,94
Charge totale.		2318	4276	3966	6429
P_0 : Somme des parties obligatoires de la fonction de charge.		304	707	618	743
P_1 : Somme des parties obligatoires de la fonction de charge après avoir placé la première tâche.		576	2907	824	993
P_2 : Somme des parties obligatoires de la fonction de charge après avoir placé la deuxième tâche.		719	–	1048	1683
P_3 : Somme des parties obligatoires de la fonction de charge après avoir placé la troisième tâche.		942	–	1107	1934

4.2.2. Commentaires pour le problème Pb2 du tableau I

Ce problème qui n'était pas résolu de façon satisfaisante par Plan nec fut à l'origine de notre étude.

Les utilisateurs n'appréciaient pas que la différence entre la solution donnée par l'ancien Plan nec ($S_0 = 37$) et le minorant ($I_m = 14$) fut de 23. Soit : $S_0/I_m \simeq 2,64$. Comme la puissance de la combinatoire ne permettait pas de connaître la valeur exacte de la solution optimale, des solutions voisines de 14 restaient envisageables. Notre minorant ($\text{Min} = 29$) éviterait à ces méthodes de faire des recherches inutiles en vue d'une solution comprise entre 14 et 28; et permettrait à l'utilisateur d'avoir une meilleure idée de l'écart à l'optimalité de la solution qu'on lui proposait.

Comme on l'a fait remarquer en 3.6, dès que l'on impose la date de début à la tâche de plus grande intensité, la partie obligatoire du tableau de charge passe de

707 à 2907 et l'intensité maximale de la fonction de charge prend la valeur 29 (minorant). A ce stade, on a déjà fixé pratiquement les 3/4 de la charge totale. Les tâches restantes de charges relativement faibles se placent très facilement sans qu'en aucun moment, l'intensité maximale dépasse 29. Pour ce problème l'heuristique 2 donne la solution optimale.

Les temps d'exécution : l'ancienne méthode Plan nec se déroule plusieurs fois en vue de résoudre le problème 3 avec M compris entre 14 et 28. Cela engendre une perte de temps assez respectable puisque notre minorant est de 29. Pour le même problème, notre méthode (heuristique 2) se déroule une seule fois d'où le gain de temps assez important.

4.3. Résultats obtenus sur 103 problèmes réels

Nous avons travaillé sur la révision de cinq centrales, dont chacune devait résoudre le problème 1 pour une vingtaine de corps de métiers (mécaniciens, électriciens, électroniciens, etc.). Une bonne partie de ces résultats est due à Messieurs J. Rivron ⁽³⁾ et J. M. Flieg ⁽³⁾ (principaux responsables de l'utilisation de Plan nec). Voici les résultats notables :

- on atteint l'optimum pour 53 problèmes (la solution proposée = minorant);
- on est à une unité du minorant pour 16 problèmes;
- et à deux unités du minorant pour les 34 problèmes restants.

En moyenne, sur les 103 exemples, la solution proposée par l'heuristique 2 est de 11% inférieure à la solution proposée par l'ancien Plan nec. Sur les 103 problèmes précédents, notre méthode s'exécute en moyenne cinq fois plus vite que l'ancienne méthode Plan nec. Cela est dû au fait que pour résoudre un problème 1, Plan nec, comme toutes les méthodes sérielles, s'exécute plusieurs fois en vue de résoudre des problèmes 3, tandis que notre méthode s'exécute une seule fois.

5. EXTENSIONS ET GÉNÉRALISATION

La partie obligatoire de la fonction de charge permet de prévoir de manière déterministe les « zones dangereuses ». Nous terminons d'abord par quelques notions permettant de prévoir *heuristiquement* ces zones, puis par une généralisation de la notion de partie obligatoire.

⁽³⁾ Ingénieurs au Service de production thermique E.D.F.

5.1. La notion de présence en moyenne

Chaque tâche i a une charge c_i et doit s'exécuter entièrement et sans interruption dans l'intervalle $]t_i^-, t_i^+ + d_i]$.

Nous pouvons dire qu'il y a en moyenne $c_i/(d_i + m_i)$ machines occupées avec la tâche i dans l'intervalle $]t_i^-, t_i^+ + d_i]$.

Appelons cette quantité la présence en moyenne de la tâche i : $(p_m(i))$.

Exemple : Supposons que l'on ait une tâche i telle que :

$$t_i^- = 5, \quad t_i^+ = 10, \quad d_i = 10, \quad M_i = 6.$$

On a alors :

$$p_m(i) = \frac{6 \times 10}{10 + (10 - 5)} = \frac{60}{15} = 4.$$

On peut donc schématiser la *présence en moyenne* de la tâche i par la figure 6.

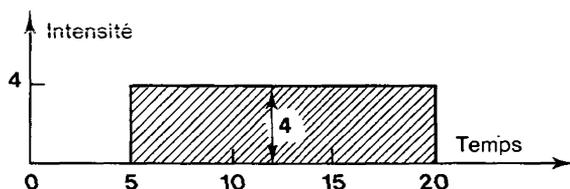


Figure 6

5.2. Heuristique 3

L'heuristique 3 consiste à adapter l'heuristique 1 comme suit : à chaque phase i , au lieu d'utiliser la partie obligatoire de la fonction de charge, on utilise une fonction de charge beaucoup plus « volumineuse ». Cela consiste à placer toutes les tâches n'ayant pas de partie obligatoire suivant leur présence en moyenne, puis pour chaque tâche ayant une partie obligatoire :

- on place sa partie obligatoire;
- on place la « charge restante » suivant sa présence en moyenne dans les deux intervalles restants.

La tâche de la figure 6, par exemple, se placerait comme indiqué à la figure 7.

Dans l'intervalle $]10, 15]$, nous plaçons la partie obligatoire de la tâche i . Cette charge est de : $5 \times 6 = 30$. Donc il reste une charge de $60 - 30 = 30$ à répartir suivant la présence en moyenne dans les deux intervalles $]5, 10]$ et $]15, 20]$.

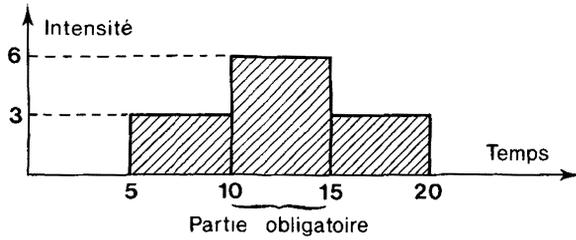


Figure 7

5.3. Présence en probabilité. L'heuristique 4

Cela consiste à adapter l'heuristique 1 avec la fonction de charge suivante :

Si $m_i < d_i$: pour tout $t \in [t_i^- + 1, t_i^+]$ on place dans l'intervalle $]t-1, t]$ un rectangle de hauteur :

$$pp_i(t) = M_i \times \frac{t - t_i^-}{m_i + 1}.$$

Dans $]t_i^+, t_i^- + d_i]$ on place un rectangle de hauteur $pp_i(t) = M_i$. On complète par symétrie pour $t \in [t_i^- + d_i + 1, t_i^+ + d_i]$. Par exemple, la tâche i de la figure 6, sera placée comme indiqué à la figure 8.

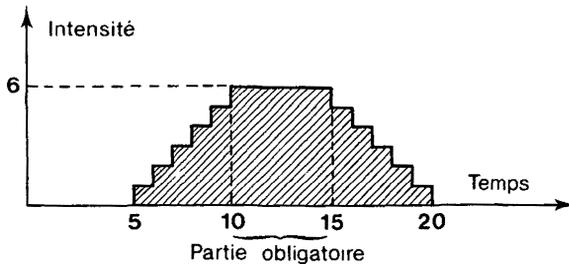


Figure 8

La quantité $pp_i(t)$ s'appellera *présence en probabilité* de la tâche i .

Si l'on note $p_i(t)$, la probabilité pour que i commence à l'instant ($t \in N$), la notion de présence en probabilité découlerait de l'hypothèse :

$$p_i(t_i^-) = p_i(t_i^- + 1) = \dots = p_i(t_i^+) = \frac{1}{m_i + 1} \quad (\text{probabilité uniforme}).$$

Si $m_i \geq d_i$, pour $t \in [t_i^- + 1, t_i^- + d_i - 1]$, on place dans $[t - 1, t]$, un rectangle de hauteur :

$$pp_i(t) = M_i \times \frac{t - t_i^-}{m_i + 1};$$

– pour $t \in [t_i^- + d_i, t_i^+ + 1]$, on place dans $]t - 1, t]$, un rectangle de hauteur :

$$pp_i(t) = M_i \times \frac{d_i}{m_i + 1};$$

– pour $t \in]t_i^+ + 1, t_i^+ + d_i]$ on complète par symétrie.

REMARQUE 6 : Deux variantes plus rapides des heuristiques 3 et 4 consistent à adapter les fonctions de charge établies suivant la présence en moyenne et la présence en probabilité à l'heuristique 2. Ces méthodes sont actuellement en cours d'expérimentation.

5.4. Une généralisation

Soit $O = \{ \theta_i \mid \theta_i \text{ est un ordonnancement réalisable} \}$. Et soient c et c^- deux fonctions telles que :

– $c([a, b], \theta_i) = \sum_{k=a+1}^b I_i(k) = \text{charge cumulée dans l'intervalle de temps } [a, b]$, pour l'ordonnancement réalisable θ_i .

On suppose que $[a, b] \subset [0, T_c]$:

– $c^-([a, b]) = \min_{\theta_i \in O} c([a, b], \theta_i)$: charge minimale de l'intervalle $[a, b]$.

Grâce à la fonction c^- , on peut prévoir les « zones dangereuses » et établir de bons minorants :

1. Soit $\max\{[c^-([a, b])/b - a]\}$ pour $[a, b] \subset [0, T_c]$. C'est un minorant pour le problème 1 (§ 1.2).

2. Si i est la tâche de plus grande charge, elle peut débuter aux instants t_i^- , $t_i^- + 1, \dots, t_i^+$; pour chacune de ces $(t_i^+ - t_i^- + 1)$ possibilités, on calcule un minorant, comme on vient de le faire au 1. Le plus petit de ces minorants est un minorant du problème 1.

On peut améliorer cette recherche du minorant en utilisant la remarque 2 (§ 3.4).

REMARQUE 7 : Ici on va présenter une façon *élémentaire*, d'utiliser nos résultats pour la résolution du problème 3. Soit la suite de problèmes $P(r)$ suivante :

$P(r)$: existe-t-il un ordonnancement réalisable, tel que :

$$T_r = T_c + r \quad \text{et} \quad I_{\max} \leq M.$$

Le plus petit entier r , tel que $P(r)$ ait une solution nous donne la solution du problème 3 : $T_r^* = T_c + r$. Or $P(r)$ est faiblement réductible au problème 1, cf. [6], ch. 10, § 1. Si l'on ajoute au graphe un arc de longueur $T_c + r$ (sans antécédent ni successeur), et d'intensité nulle, résoudre le problème 1 sur ce nouveau graphe, revient à résoudre $P(r)$.

CONCLUSION

La méthode des parties obligatoires, bien que simple dans son concept, a permis d'améliorer notablement la résolution des problèmes d'ordonnement avec contraintes cumulatives, tels que ceux rencontrés dans l'ordonnement des travaux de révision des centrales.

Les recherches actuelles, portent sur l'amélioration du problème cumulatif à un seul type de moyens, et surtout sur le problème crucial : le problème cumulatif à plusieurs types de moyens. Pour attaquer ces problèmes, les idées suivantes pourraient s'avérer très fructueuses :

- utilisation de la fonction c^- et des minorants du paragraphe 5.4. Ces résultats ont déjà été adaptés au problème particulier suivant : résoudre le problème 1 (§ 1.2), dans le cas, où toutes les tâches ont une durée égale à 1 et une intensité égale à 1. Pour les cinq problèmes réels qui ont été traités, l'optimum a été atteint;
- utiliser les tâches dont le déplacement crée une forte quantité de parties obligatoires;
- utiliser les tâches dont le déplacement augmente notablement les rapports $(c^-([a, b])/b - a)$ (§ 5.4);
- etc.

Les notions exposées dans cet article, et celles avancées dans la conclusion devraient permettre de mettre en œuvre des méthodes arborescentes, en vue d'obtenir la solution exacte pour le problème à un seul type de moyens.

REMERCIEMENT

Je tiens à remercier le referee pour ses critiques qui ont permis une nette amélioration de la première version de cet article.

BIBLIOGRAPHIE

1. J. CARLIER, *Problèmes d'ordonnancement à contraintes cumulatives*, Rapport de recherche I. Programmation Paris-VI, octobre 1979.
2. M. L. DIBON, *Ordonnements et potentiels. Méthodes MPM Herman*, Collection actualité scientifique et industrielle, Paris, vol. 70, p. 1334.
3. J. ERSCHLER, G. FONTAN et F. ROUBELLAT, *Potentiel sur un graphe non conjonctif et analyse d'un problème d'ordonnements à moyens limités*, R.A.I.R.O., Recherche opérationnelle, vol. 13, n° 4, novembre 1979.
4. R. FAURE, *Heuristique efficace, la notion de regret en recherche opérationnelle*, Referaty na III symposium. p.n.: Methody Heuresy, in Polscoe Towarzystwo Cybernetyczne (P.T.C.), Varsovie, 1976.
5. M. GONDRAN et M. DOSTATNI, *Le traitement des exclusives dans Planne*, E.D.F. Bulletin de la Direction des Études et Recherches, série C, Mathématiques, Informatique, n° 1, 1977, p. 69-78.
6. M. GONDRAN et M. MINOUX, *Graphes et algorithmes*, Eyrolles, 1979 dans la collection des Études et Recherches E.D.F.
7. A. LAHRICHI, *La notion de partie obligatoire et son application à la recherche des créneaux*, R.A.I.R.O., Informatique Computer Science, vol. 15, n° 3, 1981, p. 13 à 27.
8. A. LAHRICHI, *Ordonnements : la notion de « partie obligatoire » et son application aux problèmes cumulatifs*, Thèse du 3^e cycle. I. Programmation, Université de Paris-VI, mai 1979.
9. F. MAUREL, D. ROUX et D. DUPONT, *Techniques opérationnelles d'ordonnements*, Paris, Eyrolles, 1977.
10. B. ROY, *Algèbre moderne et théorie des graphes*, t. 1 et 2, Dunod, Paris, 1970.
11. *Théorie de la méthode Plannecc*, Service de la production thermique, Direction de la production et du transport, E.D.F.
12. R. SLOWINSKY, *Optimal and Heuristiques Procedures for Project Scheduling with Multiple Constrained Resources*, A survey, Fondation of Control Engineering, vol. 2, n° 1, 1977.