RAIRO. RECHERCHE OPÉRATIONNELLE

RAYMOND TRÉMOLIÈRES
JEAN-CLAUDE AUBERT

Détermination d'une flotte minimale et ordonnancement des taches dans la livraison du béton prêt à l'emploi

RAIRO. Recherche opérationnelle, tome 11, n° 3 (1977), p. 305-321

<http://www.numdam.org/item?id=RO_1977__11_3_305_0>

R.A.I.R.O. Recherche Opérationnelle/Operations Research
(vol. 11, n° 3, août 1977, p. 305 à 321)

DÉTERMINATION D'UNE FLOTTE MINIMALE ET ORDONNANCEMENT DES TACHES DANS LA LIVRAISON DU BÉTON PRÊT A L'EMPLOI [1]

par Raymond TRÉMOLIÈRES (*) et Jean-Claude AUBERT (**)

Résumé. — *L'article présente un modèle et une méthode de détermination d'une flotte minimale de camions pour le transport du béton. La durée des voyages est supposée non aléatoire. Le problème tient compte d'une contrainte cumulative sur les chargements. La méthode est basée sur l'utilisation d'un algorithme dynamique, direct dans un premier temps, inverse ensuite. Des résultats provenant de l'application de la méthode sur des problèmes concrets sont présentés.*

PRÉSENTATION

Nous donnons dans cette note une méthode permettant la détermination d'une flotte minimale pour la livraison du béton prêt à l'emploi. La méthode a été testée sur un grand nombre de situations concrètes et s'est révélée particulièrement intéressante par rapport au procédé habituellement utilisé par les « dispatcheurs ». Elle a permis en effet de démontrer qu'il était possible de réaliser une économie de l'ordre de 5 à 10 % sur le parc habituellement défini, et de l'ordre de 6 à 7 % sur les coûts de livraison.

D'un point de vue mathématique la méthode repose sur deux algorithmes liés, l'un de type programmation dynamique « directe », l'autre de type programmation dynamique « inverse ».

---

[1] Reçu février 1976.

(*) Maître de Conférences. Institut d'Administration des Entreprises Aix-en-Provence.

(**) Assistant de recherche LABOSID-GRASCE, Université de Droit, d'Économie et des Sciences d'Aix-Marseille.

I. INTRODUCTION

Le problème que nous étudions concerne la détermination d'une flotte minimale pour la livraison de béton prêt à l'emploi par camions-malaxeurs [1]. Ce problème se pose chaque soir aux « dispatcheurs » une fois qu'ils sont en possession de toutes les commandes du lendemain.

Afin de comprendre la nature du problème nous allons brosser un rapide tableau de la situation.

Précisons tout d'abord de quoi il s'agit: d'après la Norme Française (NF-P 18305) publiée en avril 1966, le BÉTON PRÊT A L'EMPLOI (B.P.E.), est défini comme un « matériau dont tous les composants (granulat, sable, ciment, eau, et éventuellement adjuvants) sont dosés dans une installation fixe appelée *CENTRALE*, malaxé pour être livré aux clients utilisateurs, avant début de prise, prêt à être mis en place sans autre traitement préalable ».

Le B.P.E. est donc fabriqué dans des CENTRALES, – sortes de grands silos sous lequel les camions viennent se charger –, et est livré soit par camions-bennes (essentiellement dans les cas où les clients viennent chercher eux-mêmes leur béton) soit par camions-malaxeurs (ceux-ci appartiennent soit aux Centrales, soit à des organismes les regroupant, soit à des « locatiers » qui louent leur parc). Il nous faut préciser ici que la fréquence de chargement des camions-bennes est assez faible par rapport à celle des camions-malaxeurs; d'autre part leur temps de chargement est très réduit. Par ailleurs comme ils sont chargés dans les « creux » du planning des bétonnières nous n'aurons pas à en tenir compte dans l'étude qui suit.

Les camions-malaxeurs ont une forme assez particulière (*cf. fig.* 1) et leur nombre croissant est lié à une industrialisation en pleine expansion du secteur de la construction.

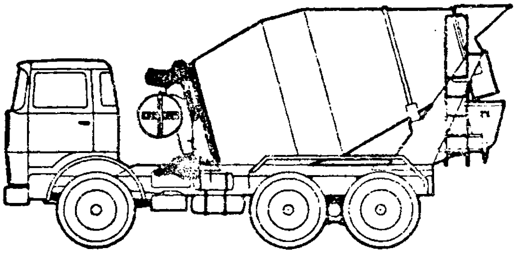

Figure 1.

Les commandes sont en quasi-totalité reçues la veille et une fois celles-ci arrêtées, les « dispatcheurs » déduisent, souvent très approximativement, le parc de camions nécessaire à la satisfaction des commandes.

---

[1] On parle aussi de camion-bétonnière, de bétonnière portée, de « toupie »...

Nous retiendrons ici que chaque commande d'un client peut se décomposer en un certain nombre de N tâches élémentaires pour lesquelles on peut spécifier

- la date de livraison
- le nom du client
- l'adresse du chantier
- l'heure à laquelle le client demande que le camion soit sur le chantier
- les temps aller et retour du camion
- le temps de déchargement sur le chantier
- le nombre de mètres cubes de béton à livrer ($\leqslant 6\ m^3$)
- la qualité du béton demandé
- l'usage auquel ce béton est destiné
- par quel moyen le béton est déchargé (pompe, grue,...).

D'autre part, pour chaque tâche que fait un camion il faut tenir compte

- d'un temps de lavage (de 5 à 10 minutes)
- du temps nécessaire à la manœuvre (environ 30 secondes)
- du temps de chargement qui est une fonction de la qualité du béton et du nombre de mètres cubes à charger : nous avons pris une moyenne de 5 minutes ; plus précisément on peut considérer qu'il faut environ 1 minute par mètre cube.

A partir de ces informations on déduit pour chaque tâche i, $i = 1, \ldots, N$, deux dates importantes :

t_i^- = heure théorique de chargement au plus tard pour effectuer la livraison au temps voulu par le client,

t_i^+ = heure théorique de prochaine disponibilité du camion en Centrale une fois la tâche accomplie en supposant que le camion ait été chargé à t_i^-.

Dans la réalité, et logiquement aussi, il est rare que l'on puisse démarrer les tâches exactement à t_i^-.

Si un camion est chargé à $t_i^- + r$ (avec $r > 0$) on peut admettre que le client sera livré avec un retard de r et le camion ne sera disponible à nouveau qu'à $t_i^+ + r$.

Par contre, si le camion est chargé à $t_i^- - a$ (avec $a > 0$), le client ne sera généralement pas prêt à réceptionner le béton plus tôt que l'heure prévue et le camion sera à nouveau disponible à t_i^+ et non à $t_i^+ - a$.

Nous noterons t_i la date effective de chargement pour la tâche i (on pourra avoir $t_i = t_i^- - a$ ou $t_i^- + r$).

La principale contrainte à prendre en compte ici vient de ce qu'*il n'est pas possible de charger simultanément* 2 camions, en même temps sous une même Centrale; d'autre part un camion ne peut évidemment pas effectuer deux tâches en même temps. Il s'agit donc d'un problème que l'on classe dans les problèmes d'*ordonnancement sous contraintes cumulatives.*

Le problème qui se pose est de déterminer un parc minimal permettant de satisfaire toutes les commandes avec un retard éventuel acceptable tout en tenant compte des temps de chargement c_i.

Dans ce qui suit nous allons définir ce problème de façon précise puis nous indiquerons comment il a été résolu.

Disons préalablement qu'à notre connaissance ce problème n'a pas été étudié tel quel jusqu'à présent. La seule étude que nous connaissons est de J. Colmin [2] dans laquelle il s'agit de l'affectation de camions entre plusieurs centrales et non du dispatching d'une centrale. Pour le lecteur intéressé, l'aspect comptable des entreprises de B.P.E. est étudié dans J. Ayanian [1].

II. DÉFINITION PRÉCISE DU PROBLÈME

Nous supposons que le parc est donné, soit M camions, $k = 1, \ldots, M$, et que pour chaque camion on connaît

- d_k^- = heure de première disponibilité du camion k en Centrale ($d_k^- > -\infty$)
- d_k^+ = heure après laquelle la camion k est considéré comme indisponible

(normalement $d_k^- \simeq 5$ h du matin, $d_k^+ \simeq 7$ h du soir).

Nous supposons d'autre part qu'il y a N tâches élémentaires à effectuer, $i = 1, \ldots, N$ et que pour chaque tâche i nous avons

- t_i^- = heure de chargement théorique,
- t_i^+ = heure de retour théorique,
- c_i = temps de chargement.

Le problème se pose ainsi : étant donnés M; N; d_k^-, d_k^+, $k = 1, \ldots, M$; t_i^-, t_i^+, c_i, $i = 1, \ldots, N$;

– peut-on effectuer les livraisons en temps voulu ?

– sinon quels sont les retards éventuels de chacune des n tâches ?

– quels sont les « meilleurs » débuts effectifs de chargement t_i, $i = 1, \ldots, N$?

– quelles sont les tâches que doivent effectuer chacun des camions en se basant :

* sur la règle FIFO (first in, first out)
* sur la règle LIFO (last in, first out)?

– quels sont les « battements » des camions avant chaque tâche, c'est-à-dire quelle est la durée de l'inaction en Centrale de chaque camion avant qu'il se présente au chargement pour une nouvelle tâche ?

III. DÉTERMINATION DES DATES DE CHARGEMENT ET CONSTITUTION DU PLANNING DES TÂCHES

Nous avons choisi de résoudre le problème en nous basant sur l'hypothèse que les tâches étaient *toutes également prioritaires* (ce qui est souvent le cas). Ceci justifie l'énoncé suivant :

> *Hypothèse fondamentale* (H) : les tâches sont renumérotées dans l'ordre croissant des t_i^-, plus précisément pour $i = 1, \ldots, N-1$
>
> $$t_i^- < t_{i+1}^-, \qquad \text{ou bien} \quad t_i^- = t_{i+1}^- \qquad \text{et} \qquad t_i^+ \leqslant t_{i+1}^+.$$
>
> D'autre part les dates effectives de chargement devront respecter les contraintes
>
> $$t_i + c_i \leqslant t_{i+1} \quad , \quad i = 1, \ldots, N-1.$$

Compte tenu de cette hypothèse et étant donnés

$$M \quad ; \quad d_k^- \quad , \quad d_k^+ \quad , \quad k = 1, \ldots, M,$$

nous allons donner une procédure permettant de définir les dates effectives de chargement t_i, $i = 1, \ldots, N$.

Les t_i seront déterminés en deux temps : dans un premier temps nous définirons des dates t_i^* telles que pour tout autre ordonnancement réalisable t_i' on ait, pour $i = 1, \ldots, N$

$$\begin{cases} t_i^* \leqslant t_i' \\ \max\,[t_i^* - t_i^-, 0] \leqslant \max\,[t_i' - t_i^-, 0]. \end{cases}$$

Dans un deuxième temps nous déterminerons à partir des t_i^* des dates t_i « réalistes » : nous entendons par là que si l'on a

$$t_i^* = t_i^- - a_i, \qquad \text{avec} \quad a_i > 0$$

et s'il est possible de programmer la tâche i à un instant plus rapproché de t_i^- (par valeurs inférieures), ceci est bien sûr préférable.

En effet que l'on « démarre » la tâche i à $t_i^* < t_i^-$ alors qu'il est possible de la programmer à t_i avec $t_i^* < t_i \leqslant t_i^-$ ne change rien à la date de retour qui restera t_i^+ ; comme la qualité du béton risque de se dégrader si l'on attend trop longtemps avant le déchargement, on a donc tout intérêt à « repousser » au maximum les « t_i^* » vers les t_i^- (pour les i tels que $t_i^* < t_i^-$); ce faisant il faut naturellement s'assurer que l'on ne « retarde » pas des tâches postérieures à 1.

Le processus se décompose en les deux algorithmes qui suivent :

III.1. Algorithme 1 : détermination des t_i^*

On pose au départ

$$\left|\begin{array}{l} t_0^* = -\infty \quad , \quad c_0 = 0 \\ s_{1,k} = d_k^- \quad , \quad k = 1, \ldots, M \\ K_1 = \{ k \mid t_1^+ + \max [s_{1,k} - t_1^-, 0] \leqslant d_k^+ \} \\ k_1 = \text{indice de } K_1 \text{ tel que } \quad s_{1,k_1} = \min (s_{1,k} ; k \in K_1) \\ p_1 = s_{1,k_1}. \end{array}\right. \tag{0}$$

Pour $i = 1, \ldots, N$ les vecteurs $s_i = (s_{i,k}, k = 1, \ldots, M)$ nous servirons à repérer les disponibilités des camions au fur et à mesure qu'ils seront affectés à de nouvelles tâches : « s_i » donne donc les disponibilités des M camions en supposant qu'il reste à programmer les tâches i, $i - 1$, $\ldots$, N.

Les ensembles K_i, $i = 1, \ldots, N$ définissent les camions qui peuvent effectuer la tâche i, c'est-à-dire ceux qui sont disponibles au moins jusqu'à la fin de la tâche i.

L'indice k_1 donne pour la tâche 1 le numéro du camion disponible depuis le plus long temps. Pour les indices k_i qui suivront ceci sera moins restrictif mais on pourra toujours considérer que k_i correspond à l'un des camions disponibles depuis le temps le plus long.

Les nombres p_i, $i = 1, \ldots, N$ définissent, pour chaque tâche i, le premier instant où un camion est disponible pour la réaliser.

On opère alors itérativement de $i = 1$ à N comme suit :

$$\boxed{t_i^* = \max [t_{i-1}^* + c_{i-1}, p_i]} \, . \tag{1}$$

puis, avant de revenir à cette formule, en incrémentant i de une unité, on définit :

$$\left|\begin{array}{ll} s_{i+1,k} & = \max [s_{i,k}, t_i^* + c_i] \quad , \quad \forall k \neq k_i \\ s_{i+1,k_i} & = t_i^+ + \max [t_i^* - t_i^-, 0] \\ K_{i+1} & = \{ k \mid t_{i+1}^+ + \max [s_{i+1,k} - t_{i+1}^-, 0] \leqslant d_k^+ \} \\ k_{i+1} & = \text{indice de } K_{i+1} \quad \text{tel que} \quad s_{i+1,k_{i+1}} = \min (s_{i+1,k} ; k \in K_{i+1}) \\ p_{i+1} & = s_{i+1,k_{i+1}}. \end{array}\right. \tag{2}$$

Ce premier algorithme est du type programmation dynamique *directe*.

On notera que si K_i est vide la tâche i est impossible à programmer. Dans ce cas on élimine la tâche de la liste et on passe aux tâches suivantes.

Nous expliciterons l'algorithme sur l'exemple de la figure 2.

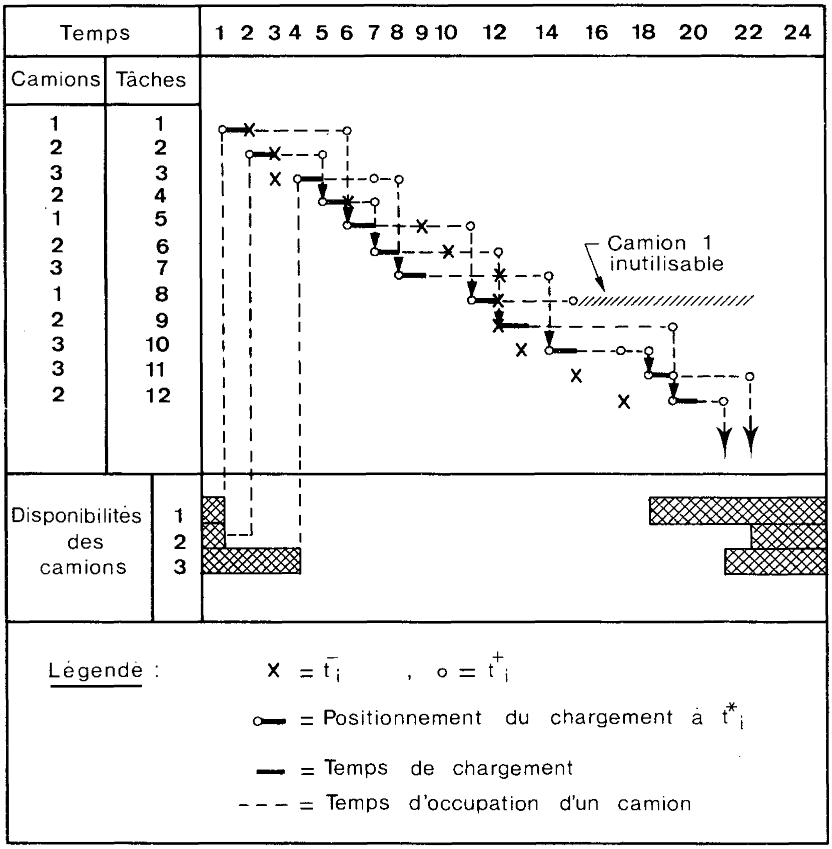

Figure 2.
Algorithme 1 pour un problème à $N = 12$ tâches et $M = 3$ camions.

Remarques sur la règle « First In, First Out »

On remarquera que dans l'algorithme précédent le choix de k_{i+1} est moins restrictif que dans le cas où l'on applique rigoureusement la règle du « premier arrivé, premier servi ». L'application stricto sensu de cette règle conduit à remplacer les formules (2) par les formules suivantes

$$\left|\begin{array}{ll} s_{i+1,k} & = s_{i,k} \quad , \quad \forall k \neq k_i \\ s_{i+1,k_i} & = t_i^+ + \max\,[t_i^* - t_i^-, 0] \\ K_{i+1} & = \{\, k \mid t_{i+1}^+ + \max\,[s_{i+1,k} - t_{i+1}^-, t_i^* + c_i - t_{i+1}^-, 0] \leqslant d_k^+ \,\} \\ k_{i+1} & = \text{indice de } K_{i+1} \text{ tel que } \quad s_{i+1,k_{i+1}} = \min\,(s_{i+1,k};\, k \in K_{i+1}) \\ p_{i+1} & = s_{i+1,k_{i+1}}. \end{array}\right. \qquad (3)$$

Si $\hat{K}^*_{i+1}$ est le sous-ensemble des k_{i+1} que l'on peut choisir dans (2), et si $\hat{K}_{i+1}$ est le sous-ensemble des k_{i+1} que l'on peut choisir dans (3) on a :

$$\hat{K}_{i+1} \subseteq \hat{K}^*_{i+1}$$

et $\hat{K}_{i+1} \neq \emptyset$ dès que $\hat{K}^*_{i+1} \neq \emptyset$: autrement dit si l'on démontre l'optimalité de l'algorithme (0) (1) (2), on démontrera par la même celle de (0) (1) (3).

III.2. Algorithme 2 : détermination des t_i

Dans la figure 2 on remarque que la tâche 7 est chargée à $t = 8$ ce qui est inutile puisqu'il n'y a pas de chargement jusqu'à $t = 10$ compris. L'algorithme qui suit va nous permettre d'obtenir la configuration réaliste de la figure 3 où l'on ne charge pas les tâches inutilement trop tôt. On détermine ainsi les dates effectives de chargement t_i.

On opère comme suit :

On pose

$$t_{N+1} = +\infty$$

puis, pour $i = N$ à 1, on définit

$$\boxed{t_i = \max\left[t_i^*, \min\left(t_{i+1} - c_i, t_i^-\right)\right]}\ .$$

A l'inverse de l'algorithme précédent, il s'agit ici d'un procédé du type programmation dynamique « *inverse* ».

Remarques sur la règle « Last In, First Out »

Une fois les t_i déterminés nous pouvons soit garder les k_i donnés par l'algorithme 1, ce qui correspond à une règle du genre « premier arrivé, premier servi », soit les recalculer complètement pour satisfaire une règle du type « dernier arrivé, premier servi », et ceci *sans changer* les t_i. Les t_i, $i = 1, \dots, N$ étant ainsi déterminés par l'algorithme (2) on définit au départ

$$s_{1,k} = d_k^- \quad , \quad k = 1, \dots, M$$

puis itérativement de $i = 1$ à N :

$$\left|\begin{array}{ll} K_i & = \{ k \mid t_i^+ + \max\left[t_i - t_i^-, 0\right] \leqslant d_k^+ \} \\ k_i & = \text{indice de } K_i \text{ tel que } s_{i,k_i} = \max\left(s_{i,k}; k \in K_i, s_{i,k} \leqslant t_i\right) \\ s_{i+1,k} & = s_{i,k} \quad , \quad \forall k \neq k_i \\ s_{i+1,k_i} & = t_i^+ + \max\left(t_i - t_i^-, 0\right). \end{array}\right.$$

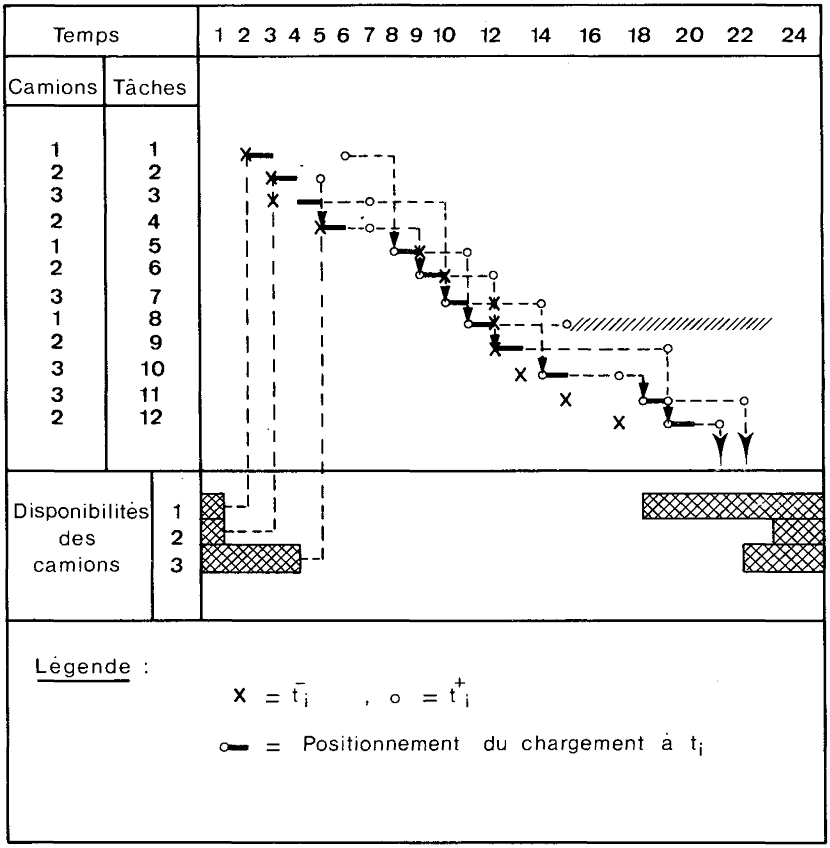

Figure 3.
Algorithme 2 appliqué aux résultats de la figure 2 (FIFO).

III.3. Convergence

Dans cette section nous établissons l'optimalité du processus.

Avant de montrer ceci par deux théorèmes, nous devons préciser quelques définitions.

Définitions et notations

1. Un *ordonnancement* $\Omega(N, M)$, de N tâches avec M camions est un ensemble de couples (t_i, k_i), $i = 1, \ldots, N$ où t_i est la date de chargement effective de la tâche i et k_i est le numéro du camion affecté à cette tâche.

2. Nous dirons qu'un ordonnancement est *réalisable* s'il vérifie

a) l'hypothèse fondamentale (H).

b) aucune tâche n'est éliminée, i.e. $K_i \neq \emptyset$, $\forall i = 1, \ldots, N$.

c) les camions ne sont pas utilisés en dehors de leurs fourchettes de disponibilité $[d_k^-, d_k^+]$.

d) un camion n'est programmé que sur une seule tâche en même temps.

3. Pour un ordonnancement réalisable quelconque $\Omega(N, M)$ on appelle *profil des disponibilités* pour la tâche i, le vecteur $s_i = (s_{i,k}, k = 1, \ldots, N)$ (donné par l'algorithme 1) des disponibilités des camions en supposant que toutes les tâches qui précèdent i sont effectuées.

4. On appelle *profil monotone* des disponibilités pour la tâche i, un vecteur $\pi_i = (\pi_{i,k}, k = 1, \ldots, M)$ dont les composantes sont celles de s_i classées en ordre croissant : on note $\tau(h)$ le numéro du camion classé au rang h.

5. On note $\Omega^*(N, M)$ un ordonnancement quelconque obtenu par l'algorithme 1 [(0) (1) (2)] et pour faciliter ce qui suit, nous noterons par une astérisque tous les éléments p_i, $s_{i,k}$, K_i, k_i, π_i attachés à $\Omega^*(N, M)$. Nous supposerons en outre que dans $\Omega^*(N, M)$, toutes les tâches sont effectuées.

Les résultats que nous allons établir concerneront les algorithmes 1 et 2 indépendamment de leurs deux variantes; il n'est cependant pas difficile de voir qu'ils les concernent aussi toutes les deux.

THÉORÈME 1 : *Soient π_i^* et π_i' des profils correspondants respectivement à $\Omega^*(N, M)$ et à un autre ordonnancement réalisable quelconque $\Omega'(N, M)$. Alors $\pi_i^* \leqslant \pi_i'$, $\forall i = 1, \ldots, N$; plus précisément*

$$\pi_{i,h}^* \leqslant \pi_{i,h}' \quad , \quad \forall h = 1, \ldots, M$$

et de plus

$$t_i^* \leqslant t_i' \quad , \quad \forall i = 1, \ldots, N.$$

Démonstration

Nous allons démontrer par récurrence que nous avons pour $i = 1, \ldots, N - 1$

$$(P_i) \quad \begin{cases} t_{i-1}^* \leqslant t_{i-1}' \\ \pi_{i,h}^* \leqslant \pi_{i,h}' \quad , \quad h = 1, \ldots, M. \end{cases}$$

Tout d'abord la propriété est vraie pour $i = 1$ puisque l'on a trivialement $t_0^* \leqslant t_0'$ en prenant par définition $t_0^* = t_0' = -\infty$; d'autre part $\pi_{1,h}^* = \pi_{1,h}'$, $h = 1, \ldots, M$.

Supposant la propriété P_i vraie pour i quelconque fixé nous allons voir que P_{i+1} est alors vraie.

Par construction on a

$$K_i^* = \{ k \mid t_i^+ + \max [s_{i,k}^* - t_i^-, 0] \leqslant d_k^+ \}$$
$$K_i' = \{ k \mid t_i^+ + \max [s_{i,k}' - t_i^-, 0] \leqslant d_k^+ \}.$$

Appelons $\tau_i^*(h)$ et $\tau_i'(h)$ les correspondances qui donnent respectivement pour Ω^* et Ω' les indices k^* et k' se trouvant au rang h dans l'ordre croissant des composantes de s_i^* et de s_i'.

On a alors par hypothèse

$$s_{i,\tau_i^*(h)}^* \leqslant s_{i,\tau_i'(h)}' \quad , \quad h = 1, \ldots, M \tag{4}$$

d'où

$$K_i' \subseteq K_i^*.$$

Or k_i^* est choisi dans K_i^* tel que $s_{i,k_i^*}^* = s_{i,\tau_i^*(1)}^* = p_i^*$; on a alors d'après (4):

$$s_{i,k_i}^* \leqslant s_{i,k}' \quad , \quad \forall k = 1, \ldots, M.$$

Soit k_i' le camion retenu dans Ω'. On aura

$$t_i^* = \max [t_{i-1}^* + c_{i-1}, s_{i,k_i^*}^*]$$
$$t_i' \geqslant \max [t_{i-1}' + c_{i-1}, s_{i,k_i'}'].$$

Par hypothèse de récurrence on a $t_{i-1}^* \leqslant t_{i-1}'$, d'où

$$t_i^* \leqslant t_i'. \tag{5}$$

En ce qui concerne les profils nous aurons

$$\begin{aligned}
s_{i+1,k}^* &= \max [s_{i,k}^*, t_i^* + c_i] \quad , \quad \forall k \neq k_i^* \qquad \text{avec} \quad k_i^* \in K_i^* \\
s_{i+1,k_i}^* &= t_i^* + \max [t_i^* - t_i^-, 0] \\
s_{i+1,k}' &= \max [s_{i,k}', t_i' + c_i] \quad , \quad \forall k \neq k_i' \qquad \text{avec} \quad k_i' \in K_i' \\
s_{i+1,k_i'}' &= t_i^+ + \max [t_i' - t_i^-, 0];
\end{aligned}$$

ou encore, en définissant h_i^* et h_i' tels que

$$k_i^* = \tau_i^*(h_i^*) \quad , \quad k_i' = \tau_i'(h_i'),$$

on peut écrire

$$\begin{cases} s_{i+1,\tau_i^*(h)}^* = \max [s_{i,\tau_i^*(h)}^* \ , \ t_i^* + c_i] \quad , \quad \forall h \neq h_i^* \\ s_{i+1,\tau_i^*(h_i^*)}^* = t_i^+ + \max [t_i^* - t_i^-, 0] \end{cases}$$

$$\begin{cases} s_{i+1,\tau_i'(h)}' = \max [s_{i,\tau_i'(h)}', t_i' + c_i] \quad , \quad \forall h \neq h_i' \\ s_{i+1,\tau_i'(h_i')}' = t_i^+ + \max [t_i' - t_i^-, 0]. \end{cases}$$

En utilisant l'hypothèse de récurrence il vient

$$\begin{cases} s_{i+1,\tau_i^*(h)}^* \leqslant s_{i+1,\tau_i'(h)}' \quad , \quad \forall h \neq h_i^* \qquad \text{et} \qquad h_i' \\ s_{i+1,\tau_i^*(h_i^*)}^* \leqslant s_{i+1,\tau_i'(h_i')}'. \end{cases}$$

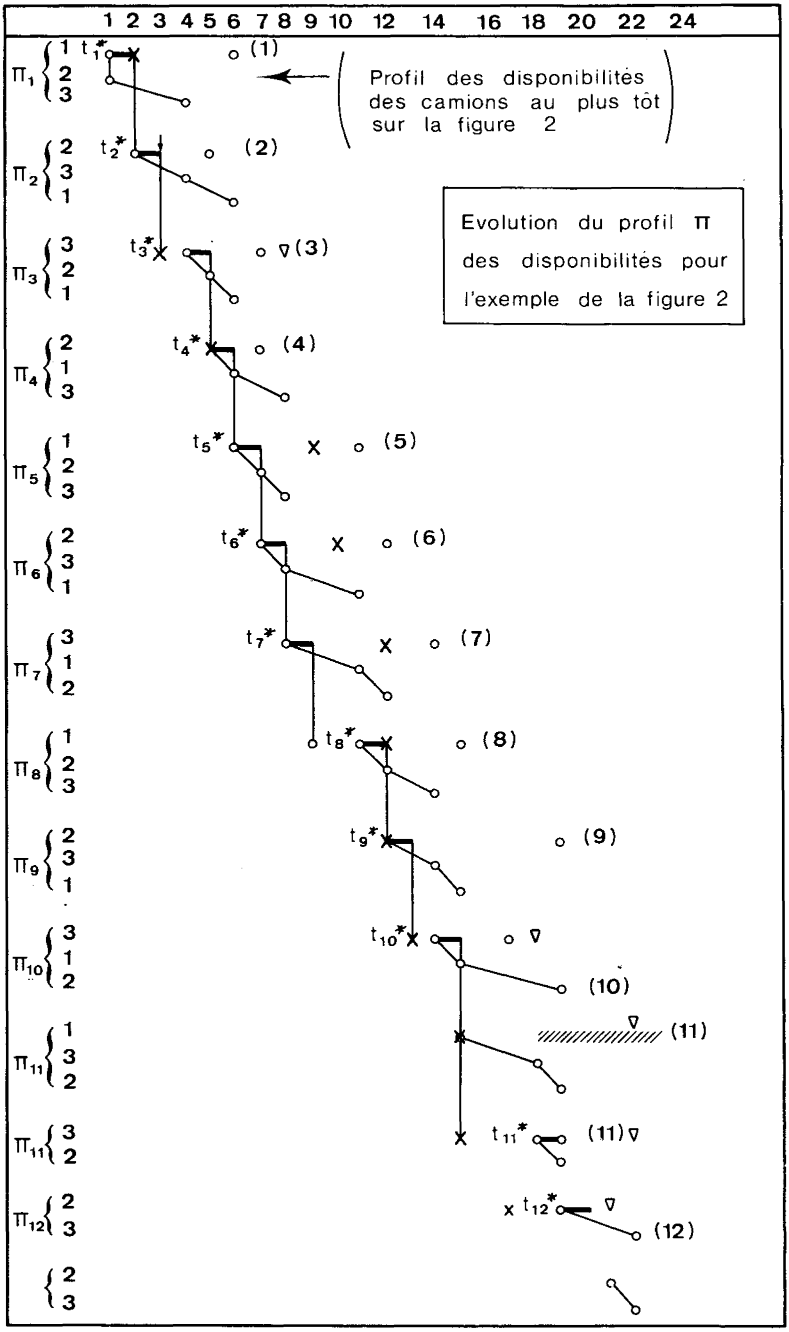
1 2 3 4 5 6 7 8 9 10 12 14 16 18 20 22 24
Profil des disponibilités des camions au plus tôt sur la figure 2
Evolution du profil π des disponibilités pour l'exemple de la figure 2
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)

Figure 4.

Si $h_i^* = h_i'$ le résultat est établi. Sinon, en supposant $h_i^* \neq h_i'$, il reste à montrer que

$$s^*_{i+1,\tau_i^*(h_i')} \leqslant s'_{i+1,\tau_i'(h_i^*)}.$$

Nous avons

$$s^*_{i+1,\tau_i^*(h_i')} = \max\left[s^*_{i,\tau_i^*(h_i')}, t_i^* + c_i\right]$$
$$s'_{i+1,\tau_i'(h_i^*)} = \max\left[s'_{i,\tau_i'(h_i^*)}, t_i' + c_i\right].$$

Or

$$s'_{i+1,\tau_i'(h_i^*)} \geqslant t_i' + c_i$$
$$t_i' + c_i \geqslant s'_{i,k_i'} + c_i = s'_{i,\tau_i'(h_i')} + c_i \geqslant s^*_{i,\tau_i^*(h_i')} + c_i \geqslant s^*_{i,\tau_i^*(h_i')}$$
$$t_i' + c_i \geqslant t_i^* + c_i$$

d'où

$$s'_{i+1,\tau_i'(h_i^*)} \geqslant \max\left[s^*_{i,\tau_i^*(h_i')}, t_i^* + c_i\right] = s^*_{i+1,\tau_i^*(h_i')}.$$

La propriété P_1 étant vraie, le théorème est donc démontré.

COROLLAIRE. *Soit $\Omega^*(N, M)$ un ordonnancement obtenu par l'algorithme 1 et $\Omega'(N, M)$ un ordonnancement réalisable quelconque alors*

$$\max\left[t_i^* - t_i^-, 0\right] \leqslant \max\left[t_i' - t_i^-, 0\right] \quad , \quad \forall i = 1, \ldots, N$$

Ce résultat est immédiat puisque $t_i^* \leqslant t_i', \forall i$. L'algorithme définit donc un ordonnancement tel que chacune des tâches est programmée soit en temps voulu, soit avec un *retard minimal.*

Nous illustrons l'évolution du profil monotone dans la figure 4 ci-contre.

THÉORÈME 2. *Soit $\Omega(N, M)$ l'ordonnancement obtenu par l'algorithme 2. On a alors*

$$\max\left[t_i - t_i^-, 0\right] = \max\left[t_i^* - t_i^-, 0\right]. \tag{6}$$

De plus pour tout autre ordonnancement $\Omega'(N, M)$ réalisable et « satisfaisant », et nous entendons par là que

$$\max\left[t_i' - t_i^-, 0\right] = \max\left[t_i^* - t_i^-, 0\right], \tag{7}$$

on a

$$\min\left[t_i' - t_i^-, 0\right] \leqslant \min\left[t_i - t_i^-, 0\right]; \tag{8}$$

ce qui traduit le fait que les camions ne peuvent être chargés qu'inutilement trop tôt dans Ω'.

Démonstration

Nous démontrons d'abord (6). On a

$$t_i - t_i^- = \max\left[t_i^* - t_i^-, \min\left(t_{i+1} - (t_i^- + c_i), 0\right)\right]$$

Si $\quad t_{i+1} \geqslant t_i^- + c_i \quad$ alors

$$t_i - t_i^- = \max\left[t_i^* - t_i^-, 0\right]$$

d'où (6).

Si $t_{i+1} < t_i^- + c_i$ alors

$$t_i - t_i^- = \max [t_i^* - t_i^-, t_{i+1} - (t_i^- + c_i)].$$

Dans le cas où $t_i^* - t_i^- \leqslant 0$ on a $t_i - t_i^- \leqslant 0$ d'où (6).
Dans le cas où $t_i^* - t_i^- > 0$ on a $t_i - t_i^- = t_i^* - t_i^-$ d'où (6).

Montrons (8) par récurrence inverse: pour $i = N$ on a, pour $\Omega(N, M)$,

$$t_N = \max [t_N^*, t_N^-]. \tag{9}$$

Considérant $\Omega'(N, M)$ on a deux cas. Soit $t_N^* > t_N^-$, alors $t'_N = t_N^*$ d'après (7) et $t_N = t_N^*$ d'après (9). On a donc (8).

Soit $t_N^* \leqslant t_N^-$, alors $t_N = t_N^-$ d'après (9) d'où $\min [t_N - t_N^-, 0] = 0$. Or d'après (7) on a alors $t'_N \leqslant t_N^-$ d'où (8).

Supposons maintenant la propriété (8) vraie pour $i + 1$ et montrons qu'elle le reste pour i.

On a

$$t_i = \max [t_i^*, \min (t_{i+1} - c_i, t_i^-)]. \tag{10}$$

Supposons $t_i^* > t_i^-$ alors

$$t_i = \max [t_i^*, \min (t_{i+1} - c_i, t_i^-)] = t_i^*$$

et $t'_i = t_i^*$ d'après (7), d'où (8) vérifié en égalité.

Supposons $t_i^* \leqslant t_i^-$ et examinons deux cas.

Soit $t_{i+1} - c_i \leqslant t_i^-$: d'après (10) on a alors $t_i \geqslant t_{i+1} - c_i$. Or par hypothèse $t'_{i+1} \leqslant t_{i+1}$ d'où $t'_i \leqslant t_{i+1} - c_i$ par définition: il vient $t'_i \leqslant t_i$ d'où (8).

Soit $t_{i+1} - c_i > t_i^-$ alors $t_i = \max (t_i^*, t_i^-) = t_i^-$. D'après (7) on a $t'_i \leqslant t_i^-$ d'où (8).

La propriété (8) étant vraie de $i = N$ est alors vraie pour $i = N - 1, \ldots, 1$.

IV. APPLICATION

Nous donnons ici un exemple de planning obtenu par ordinateur pour une centrale de la région d'Aix-Marseille le jeudi 2 octobre 1975.

Il y avait 22 commandes qui se décomposaient en 52 voyages ou tâches élémentaires.

Le problème a été résolu comme suit : Nous avons fait un premier passage sur ordinateur avec la règle LIFO pour un parc limite de 20 camions. L'application de cette règle conduit au résultat que 14 camions suffisent pour assurer toutes les tâches sans aucun retard. On a alors réappliqué le programme en réduisant le parc pour voir quels genres de retard cela pouvait entraîner.

Numéro de la commande	Numéro des tâches	t_j^-	t_j^+	t_j	Retour	Battement	Numéro du camion	
1	1	7h05	8h10	6h55	8h10	1h55	1	
12	1	7h05	8h20	7h00	8h20	2h00	2	
13	1	7h05	8h35	7h05	8h35	2h05	3	
9	1	7h15	8h20	7h15	8h20	2h15	4	
1	2	7h25	8h30	7h25	8h30	2h25	5	
4	1	7h35	8h40	7h30	8h40	2h30	6	
12	2	7h35	8h50	7h35	8h50	2h35	7	
1	3	7h45	8h50	7h45	8h50	2h45	8	
4	2	7h55	9h00	7h55	9h00	2h55	9	
1	4	8h05	9h10	8h00	9h10	3h00	10	
14	1	8h05	9h35	8h05	9h35	3h05	11	
4	3	8h15	9h20	8h15	9h20	0h05	1	
1	5	8h25	9h30	8h20	9h30	0h00	2	
5	1	8h25	10h10	8h25	10h10	0h05	4	
4	4	8h35	9h40	8h35	9h40	0h05	5	
1	6	8h45	9h50	8h45	9h50	0h10	3	
4	5	8h55	10h00	8h50	10h00	0h10	6	
1	7	9h05	10h10	8h55	10h10	0h05	7	
7	1	9h05	10h10	9h00	10h10	0h10	8	
17	1	9h05	10h20	9h05	10h20	0h05	9	
10	1	9h15	10h20	9h15	10h20	0h05	10	
1	8	9h25	10h30	9h25	10h30	0h05	1	
7	2	9h25	10h30	9h30	10h35	0h00	2	
21	1	9h25	11h00	9h35	11h10	0h00	11	
1	9	9h45	10h50	9h45	10h50	0h05	5	
21	2	9h55	11h30	9h55	11h30	0h05	3	
1	10	10h05	11h10	10h05	11h10	0h05	6	
22	1	10h15	11h10	10h15	11h10	0h05	4	
1	11	10h25	11h30	10h20	11h30	0h10	7	
21	3	10h25	12h00	10h25	12h00	0h15	8	
21	4	10h55	12h30	10h55	12h30	0h35	9	
20	1	13h05	14h10	12h55	14h10	2h35	10	
6	1	13h05	14h20	13h00	14h20	2h30	1	
8	1	13h05	14h30	13h05	14h30	2h30	2	
20	2	13h25	14h30	13h15	14h30	2h25	5	
18	1	13h25	14h40	13h20	14h40	2h10	4	
19	1	13h25	14h50	13h25	14h50	2h15	6	
2	1	13h35	14h40	13h30	14h40	2h20	11	
6	2	13h35	14h50	13h35	14h50	2h05	3	
20	3	13h45	14h50	13h40	14h50	2h10	7	
19	2	13h45	15h10	13h45	15h10	1h45	8	
2	2	13h55	15h00	13h55	15h00	1h25	9	
20	4	14h05	15h10	14h05	15h10	1h35	12	
6	3	14h05	15h20	14h10	15h25	0h00	10	
19	3	14h05	15h30	14h20	15h45	0h00	1	*
2	3	14h15	15h20	14h30	15h35	0h00	2	*
19	4	14h25	15h50	14h35	16h00	0h05	5	
15	1	14h55	16h10	14h50	16h10	0h10	4	
16	1	14h55	16h20	14h55	16h20	0h15	11	
11	1	15h05	16h35	15h05	16h35	0h15	3	
16	2	15h15	16h40	15h15	16h40	0h25	6	
3	1	15h35	16h50	15h35	16h50	0h45	7	

Avec 13 camions la somme des retards est de 15 minutes, ce qui est absolument négligeable puisque l'on tolère un retard de 10 à 15 minutes par tâches.

Après examen du planning nous avons alors essayé 11 camions le matin et 12 l'après-midi. On trouve un retard total de 1 h avec deux tâches seulement retardées de 15 minutes et ce l'après-midi. La décision fut alors prise de fixer 11 camions le matin et 12 l'après-midi.

Le planning correspondant, avec la règle FIFO, est listé ci-avant.

V. COMMENTAIRES

L'hypothèse fondamentale qui sous-tend le modèle ne se retrouve pas d'habitude dans les procédures d'ordonnancement systématiques. Nous allons voir sur un exemple la raison de son adoption dans cette étude.

Considérons en effet la figure 5 où nous prenons 1 camion pour 2 tâches « *a* » et « *b* ». Si le problème est de minimiser les retards on doit effectuer (paradoxalement pourrait-on dire) la tâche « *b* » en premier, ce qui entraîne un retard de 50 pour la tâche « *a* »; alors qu'en programmant « *a* » la première, on a un retard de 60 pour la tâche « *b* ».

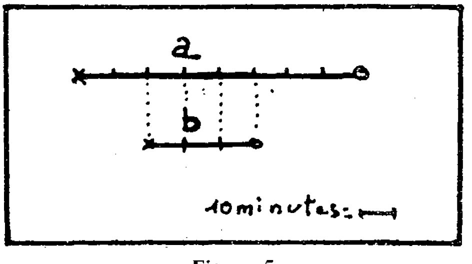

Figure 5.

Il semblerait donc que notre hypothèse fondamentale soit « théoriquement » trop contraignante (elle programme « *a* » d'abord) : en fait la réalité empêche de considérer que l'on puisse programmer des tâches avec un trop grand retard, l'exemple en question n'est donc pas, « pratiquement », acceptable puisque les retards excèdent la longueur d'une tâche, laquelle est toujours supérieure à 20 minutes. Sur un tel exemple il faut en fait « deux » camions.

Nous tenons à préciser que cette étude constitue une approche réaliste du « vrai » problème. La réalité est cependant plus compliquée et fera l'objet de publications ultérieures dans lesquelles nous aborderons en particulier le problème de la gestion coordonnée d'un ensemble de Centrales.

Nous distinguerons en outre des priorités différentes selon les tâches et nous aborderons les problèmes d'équilibrage des tâches et des affectations des camions entre les Centrales (cf. (3) (4) (5)).

REMERCIEMENTS

Nous tenons à remercier Monsieur J. A. Bartoli pour l'aide précieuse qu'il nous a apportée dans la phase de test « in situ » qui s'est échelonnée sur environ 1 mois.

Nous tenons bien sûr à remercier tout le personnel et la direction de la Société pour laquelle cette étude a été réalisée.

RÉFÉRENCES

1. J. AYANIAN, *Contribution à l'organisation comptable et au contrôle de gestion dans l'industrie du béton prêt à l'emploi*, Mémoire d'Expertise Comptable, avril 1974.
2. J. COLMIN, *Gestion en temps réel d'un parc de camions*, Document IBM consacré à la Recherche Opérationnelle, octobre 1969.
3. R. TRÉMOLIÈRES, J. C. AUBERT, *A decision model for the regulation of orders in the manufacture of concrete*, IFAC Workshop, Toulouse, september, 1977.
4. R. TRÉMOLIÈRES, *Un modèle interactif de décision et d'ordonnancement avec priorités*, Congrès AFCET, novembre 1977.
5. R. TRÉMOLIÈRES, *Scheduling of one unit of a single resource*, Note I.A.E. Aix, avril 1977.