RAIRO. RECHERCHE OPÉRATIONNELLE

FOUAD FAYEZ BOCTOR
RAYMOND TRÉMOLIÈRES

Optimisation de la production d'acier en coulée continue : approches pour un grand problème combinatoire

RAIRO. Recherche opérationnelle, tome 15, n° 4 (1981), p. 317-333

<http://www.numdam.org/item?id=RO_1981__15_4_317_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 15, n° 4, novembre 1981, p. 317 à 333)

OPTIMISATION DE LA PRODUCTION D'ACIER EN COULÉE CONTINUE : APPROCHES POUR UN GRAND PROBLÈME COMBINATOIRE (*)

par Fouad Fayez BOCTOR et Raymond TRÉMOLIÈRES (1)

Résumé. – *Ce premier rapport tente d'apporter un certain nombre d'éléments pour la résolution d'un très grand problème à propos de la production d'acier. Toute tentative de prise en compte de la totalité des composantes de ce problème dans une même modélisation conduisant à une impasse, nous proposons ici de tenir compte de la structure « décentralisable » d'un tel problème.*

Ce rapport permet d'illustrer la nécessité, surtout à propos des problèmes concrets, de ne pas séparer les phases de modélisation des considérations concernant la résolution elle-même.

Mots clés : Programmation mathématique, programmation en nombres entiers, décomposition, combinatoire, gestion de la production, acier.

Abstract. – *In this paper a large scale integer programming problem arising in the area of steel production is considered. Two decentralized models are given according to different ways of handling the costs.*

This paper shows why it is necessary to avoid the temptation of considering that modelization and resolution can be viewed separatly.

Keywords: Mathematical programming, integer programming, decomposition, combinatorial methods, production management, steel mill.

REMERCIEMENTS

Nous avons une dette de reconnaissance envers notre ami Bernard Martinet auquel nous devons de nous avoir signalé ce problème. Nous remercions les ingénieurs de la Solmer qui nous ont facilité la tâche dans ce travail.

I. INTRODUCTION

Les tentatives d'application de la programmation mathématique dans le secteur de l'acier sont illustrées par un certain nombre d'études dans les domaines suivants :

– des modèles économiques interindustriels : *cf.* par exemple Tsao-Day [20];

– des modèles d'optimisation globale de la production dans les acieries : *cf.* Phaff [14], Redwine-Wismer [16];

– des modèles d'optimisation des mélanges : *cf.* Driebeek [6] (p. 34, 71 et 92), Fabian [8];

(*) Reçu mai 1980.

(1) Institut d'Administration des Entreprises, 29, avenue Robert-Schuman, 13617 Aix-en-Provence.

R.A.I.R.O. Recherche opérationnelle/Operations Research, 0399-0559/1981/317/$ 5.00

– le problème des « recuits de bandes d'acier doux bobinées » : *cf.* Mathieu-Paing [13]. Ce problème, de type « knaspack », vise essentiellement à emmagasiner en piles des bobines de différents formats. La résolution est faite par programmation linéaire et par une exploration lexicographique de sous-problèmes de « knapsack » uni-contrainte par une méthode lexicographique;

– enfin, le célèbre problème de découpe des rubans, présenté pour la première fois par Eisemann [7], problème auquel Gilmore et Gomory ont attaché une méthode de résolution très souvent citée, et que de nombreux auteurs continuent à étudier : *cf.* Haessler [11], Haessler-Vonderembse [12], Ahluwalia-Saxena [1], Sundaram [18], Caruso-Kokat [4], Phillipson-Ravindran [15].

A notre connaissance, le problème que nous abordons ici, bien que typiquement du secteur de l'acier, semble absent de la littérature. Il concerne l'ordonnancement de la production d'acier par coulée continue en vue de satisfaire un certain carnet de commandes au moindre coût, ceci dans une technologie qui impose des règlages à chaque changement de largeurs.

Précisons d'abord ce que l'on appelle une commande. Il s'agit de la donnée :

– d'un code-acier (un certain type d'acier);

– d'une largeur;

– d'un délai ou date de livraison.

Les clients demandent, en effet, des aciers d'une qualité donnée, aciers devant être généralement livrée en « rouleaux » – ou bobinés ([1]) –, sortes de grands rubans qui, déroulés, forment des tôles d'acier d'une certaine largeur.

Au sortir du four, l'acier, contenu dans des « poches » d'environ 300 t, est déversé dans une sorte d'entonnoir à deux sorties que l'on appelle un distributeur. Chaque poche contient un certain type d'acier. Un même code-acier sera donc produit par multiple de 300 t, avec passage à d'autres codes entre 2 poches. Les 2 « goulottes » du distributeur sont réglées chacune en fonction des largeurs à produire dans un même code-acier.

Les largeurs possibles sont de 85 à 155 cm par pas de 5 cm. On appelle « réglage » un couple de largeurs installé sur un même distributeur.

Dans les procédés classiques de la coulée continue ces largeurs ne peuvent être modifiées qu'après interruption du processus, – ce qui demande que le distributeur ainsi que la poche correspondante soient entièrement vidés. Ces interruptions impliquent d'ailleurs un changement de distributeur, et elles sont d'environ 1 heure chaque fois qu'on veut changer les largeurs. La production doit, de toute façon, être interrompue toutes les 5 poches, ou même moins, un

([1]) « coil » en anglais.

même distributeur ne pouvant, du fait de sa détérioration, supporter une coulée en continu de plus de 5 poches successives. Dans la mesure du possible, il convient de tenter de faire correspondre les changements de distributeur avec les modifications nécessaires de réglage. Pour l'une ou l'autre de ces interruptions, la coulée continue étant interrompue, il convient de « réarmer la coulée » pour reprendre la production.

Le chiffre de 5 poches est ici donné pour indication. La fréquence de changement du distributeur peut dépendre, en effet, du code-acier. Un changement de distributeur, sans modification de largeurs, implique une perte de temps d'environ 1/4 d'heure.

Deux poches contenant des aciers à codes différents peuvent être vidées l'une après l'autre à condition de conserver les mêmes largeurs. Ceci peut se faire, – et c'est même recommandé – sans interrompre la coulée continue et implique un « ralentissement » d'environ 1/4 d'heure.

Dans cette aciérie, l'acier peut aussi être coulé en « discontinu » suivant un autre processus qu'on appelle la « voie lingot ». Ce deuxième processus est considéré comme plus coûteux que la « voie continue ».

Nous allons donner 2 modélisations du problème selon les « coûts » retenus. Il y a globalement 3 approches :

– minimiser les «chutes », c'est-à-dire la production en trop due au fait qu'un même code-acier ne peut être produit que par multiple de 300 t; ceci étant l'objectif principal on peut alors tenir compte du coût des interruptions pour en diminuer le nombre;

– minimiser les « interruptions », c'est-à-dire satisfaire la demande avec un minimum de règlages; cet objectif étant atteint, chercher à minimiser les « chutes » ou les « manques » (demandes non satisfaites par la coulée continue mais reportées sur la voie lingot);

– résolution complète du problème en minimisant un coût total faisant intervenir le coût des interruptions, celui des chutes et celui des manques.

Dans une première partie (II), nous modéliserons le premier problème dans lequel l'objectif fondamental est la minimisation des chutes; nous prendrons alors comme objectif secondaire celui de la minimisation du nombre d'interruptions.

Dans une seconde partie (III), nous renversons les priorités sur les objectifs. Dans une troisième partie (IV), nous abordons le problème en toute généralité en cherchant à minimiser un coût global intégrant les interruptions, les chutes et les manques.

II. LA MINIMISATION DES CHUTES

Nous basant sur les considérations précédentes, nous allons modéliser, dans un premier temps, le problème dans lequel l'objectif fondamental est celui de la minimisation des chutes. Nous y annexerons ensuite la prise en compte des interruptions pour tenter d'en minimiser le nombre.

1. Modélisation et considérations sur la résolution du problème central

Plutôt que d'envisager dès l'abord une modélisation globale de l'ensemble du problème, nous avons choisi de retenir comme premier objectif celui de résoudre le problème minimal sous-jacent à l'ensemble des divers critères de coûts, ou de temps que l'on peut attacher à un tel système de production.

Nous allons voir qu'il est possible, au moins dans un premier temps, de formuler le problème de l'adéquation de la production d'acier dans un code donné avec les largeurs existant dans le carnet de commandes.

Ceci conduit à formuler l'équation de base du problème.

Les données sont :

m, nombre de largeurs à satisfaire dans un certain code-acier; D_i, demande (en tonnes) pour une largeur donnée i, $i=1, \ldots, m$; les demandes sont supposées à égalité d'urgence dans un délai d'approximativement 4 jours; $d_i = D_i/300$; L_i, largeur i, $i=1, \ldots, m$.

Les inconnues sont :

x_{ij}, nombre de poches à vider pour un règlage ij;

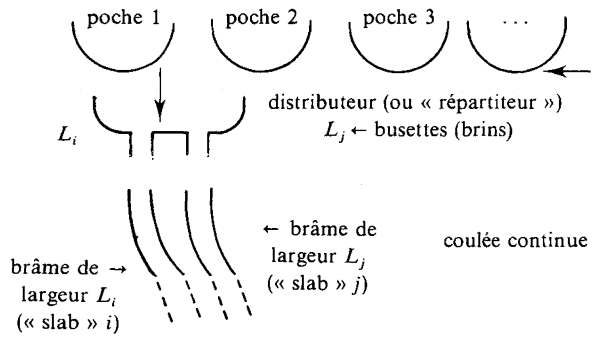
L'équation de base est, pour une demande i :

$$\sum_{\substack{k \\ (k<i)}} \frac{L_i}{L_k+L_i} x_{ki} + x_{ii} + \sum_{\substack{j \\ (i<j)}} \frac{L_i}{L_i+L_j} x_{ij} = D_i/300.$$

Cette équation est basée sur la considération que sur les 2 « brins » les vitesses d'écoulement sont les mêmes, de même que les épaisseurs; en sorte que 300 t vidées avec un règlage ij donneront :

$$\frac{L_i}{L_i+L_j} \times 300 \text{ t en largeur } L_i,$$

$$\frac{L_j}{L_i+L_j} \times 300 \text{ t en largeur } L_j.$$



Dans l'équation, les x_{ij} sont des nombres entiers, $0, 1, 2, \ldots$ Sauf exception, l'équation ne peut donc être satisfaite à l'égalité.

Dans une première approche, nous imposerons que les demandes d_i soient satisfaites par valeurs supérieures et nous tentons de minimiser les chutes.

Ceci mène à la formulation mixte :

$$(\mathrm{PM}) \begin{cases} \underline{\min}\ \lambda_1+\lambda_2+\ldots\lambda_m, \\ \sum\limits_{(k<i)}^{k} a_{ik}x_{ki}+x_{ii}+\sum\limits_{(i<j)}^{j} a_{ij}x_{ij}-\lambda_i=d_i, \qquad i=1,\ldots,m, \\ \lambda_i \geqq 0 \quad \text{(variable réelle « continue »)}, \qquad i=1,\ldots,m, \\ x_{ij}\geqq 0 \text{ entier}, \qquad i=1,\ldots,m;\ \ j=1,\ldots,m, \end{cases}$$

où les a_{ij} définissent les coefficients de la matrice des contraintes avec

$$a_{ij}=L_i/(L_i+L_j)$$

et où les variables sont les x_{ij}, avec $i \leqq j$.

Le problème apparaît à ce stade comme rentrant dans le cadre de la programmation linéaire mixte.

En remarquant que :

$$a_{ij}+a_{ji}=1$$

et en sommant toutes les équations, on obtient :

$$\sum_{i=1}^{m}\lambda_i=\sum_{i=1}^{m}\sum_{j=i}^{m}x_{ij}-\sum_{i=1}^{m}d_i.$$

Comme $\sum d_i$ est constant, le problème se formule en fait comme un programme en nombres entiers purs de la catégorie des « Multiconstraints knapsack Problem » ([2]). Le *problème central* du modèle est le suivant :

$$(P)\qquad \begin{cases} \underline{\min} \sum_{i=1}^{m} \sum_{j=i}^{m} x_{ij}, \\ \sum_{\binom{k}{k<i}} a_{ik} x_{ki} + x_{ii} + \sum_{\binom{j}{i<j}} a_{ij} x_{ij} \geqq d_i, \qquad i=1,\ldots,m, \\ x_{ij} \geqq 0 \text{ entier}, \qquad i=1,\ldots,m; \qquad j=1,\ldots,m. \end{cases}$$

Autrement dit, le problème de minimiser la chute totale ($\sum \lambda_i$) est équivalent à celui de minimiser le nombre de poches ($\sum x_{ij}$) nécessaires pour satisfaire les demandes d'un même code-acier. Pour permettre de mieux saisir la structure de ce problème, nous en donnons une présentation dans le cas de 3 largeurs fictives avec les demandes correspondantes (dans un même code-acier) :

$$\begin{aligned} L_1 &= 1, \quad d_1 = 2{,}2 \quad (D_1 = 660 \text{ t}), \\ L_2 &= 2, \quad d_2 = 1{,}7 \quad (D_2 = 510 \text{ t}), \\ L_3 &= 3, \quad d_3 = 5{,}4 \quad (D_3 = 1\,620 \text{ t}). \end{aligned}$$

Le problème s'écrit :

$$\begin{cases} \underline{\min}\ X_{11} + X_{22} + X_{33} + X_{12} + X_{13} + X_{23} \\ X_{11} + \frac{1}{3} X_{12} + \frac{1}{4} X_{13} \geqq 2{,}2 \\ X_{22} + \frac{2}{3} X_{12} + \frac{2}{5} X_{23} \geqq 1{,}7 \\ X_{33} + \frac{3}{4} X_{13} + \frac{3}{5} X_{23} \geqq 5{,}4 \\ x_{ij} \text{ entier} \geqq 0, \qquad i \leqq j, \quad i=1,2,3, \quad j=1,2,3. \end{cases}$$

On remarque que, dans la matrice des contraintes, il n'y a au plus que deux coefficients non nuls par colonne et que la somme des coefficients d'une même colonne est égale à 1.

([2]) *Voir* par exemple W. Shih [17] pour une revue bibliographique récente.

On peut aussi remarquer que ce problème possède une solution réalisable évidente, — mais généralement non optimale —, à savoir ([3]) :

$$x_{11}=[d_1]^+, \qquad x_{22}=[d_2]^+, \qquad x_{33}=[d_3]^+.$$

Les solutions du problème en nombre entier telles que :

$$\sum_{i<j} x_{ij} \leqq \sum_{i} [d_i]^+ - 1,$$

sont indiquées ci-après ([4]) :

TABLEAU I

N°	x_1	x_2	x_3	x_{12}	x_{13}	x_{23}
1.....	0	0	0	2	7	1
2.....	0	0	1	3	6	0
3.....	0	0	2	3	5	0
4.....	1	0	0	1	4	4
5.....	1	0	0	1	5	3
6.....	1	0	1	1	4	3
7.....	1	0	2	2	3	2
8.....	1	0	2	2	4	1
9.....	1	0	3	2	3	1
10.....	1	0	4	3	2	0
11.....	1	0	5	3	1	0
12.....	1	1	0	0	5	3
13.....	1	1	0	0	6	2
14.....	1	1	1	0	5	2
15.....	1	1	2	1	4	1

N°	x_1	x_2	x_3	x_{12}	x_{13}	x_{23}
16.....	1	2	1	0	6	0
17.....	1	2	2	0	5	0
18.....	2	0	1	0	2	5
19.....	2	0	2	0	1	5
20.....	2	0	3	1	0	4
21.....	2	0	3	1	1	3
22.....	2	0	4	1	0	3
23.....	2	0	5	2	0	1
24.....	2	1	2	0	3	2
25.....	2	1	3	0	1	3
26.....	2	1	3	0	2	2
27.....	2	1	4	0	1	2
28.....	2	1	5	1	0	1
29.....	2	2	4	0	2	0
30.....	2	2	5	0	1	0

Toutes ces solutions sont telles que $\sum_{i \leqq j} x_{ij} = 10$, soit 10 poches de 300 t. Il est à remarquer que l'optimum est très « plat », et qu'il est obtenu par un grand nombre de solutions équivalentes.

On constate aussi que la résolution de ce problème par *programmation linéaire continue* (on suppose les « x » continus) n'est, — sauf cas exceptionnel —, d'aucun intérêt, et conduit à une solution fort « éloignée » de la solution en nombres entiers. Une résolution du problème en continu ne fera que donner pour solution optimale la solution :

$$x_{ii}=d_i, \qquad i=1,\ldots,m,$$

([3]) $[d]^+$ = plus petit entier supérieur ou égal à d.

([4]) Solutions obtenues par la méthode lexicographique de Tremolières [19].

soit $x_{11}=2{,}2$, $x_{22}=1{,}7$, $x_{33}=5{,}4$, $x_{12}=x_{13}=x_{23}=0$, d'où en arrondissant : $x_{11}=3$, $x_{22}=2$, $x_{33}=6$, ce qui nécessite 11 poches et une perte de 300 t par rapport à l'optimum. Il va de soi aussi que les indications fournies par les variables duales n'offrent aucun intérêt pour guider le choix vers des variables entières.

Il est d'autre part facile de vérifier que le gain maximal que l'on peut réaliser à partir de la solution $\sum [d_i]^+ \times 300$ est de :

$$(\sum [d_i]^+ - [\sum d_i]^-) \times 300.$$

Considérations sur la dimension du problème P

Pour m largeurs, le problème P possède $m(m+1)/2$ variables :

- pour 5 largeurs : 15 variables entières, 5 contraintes;
- pour 10 largeurs : 55 variables entières, 10 contraintes;
- pour 15 largeurs : 120 variables entières, 15 contraintes.

Il est de plus aisé de fixer une borne supérieure à chaque variable en fonction des valeurs du second nombre :

$$x_{ij} \leqq [d_i]^+,$$

$$x_{ij} \leqq \max [[a_{ij} d_i]^+, [a_{ji} d_j]^+].$$

Le problème (P) peut donc être transformé en un problème bivalent dont la dimension sera d'autant plus grande que les d_i seront plus importants. On rappelle que :

$$x \text{ entier} \in [0, p] \Leftrightarrow x = \sum_{i=1}^{s} 2^i \delta_i,$$

avec : δ_i, variable bivalente (0,1) de rang i attachée à x_i; s, plus grand entier tel que $2^s \leqq p$.

A titre d'exercice nous précisons pour l'exemple ci-dessus le nombre de variables bivalentes que l'on peut attacher à chaque variable entière :

$$x_{11} \to 2; \qquad x_{22} \to 2; \qquad x_{33} \to 3,$$

$$x_{12} \to 3; \qquad x_{13} \to 4; \qquad x_{23} \to 4,$$

soit un total de 18 variables bivalentes.

A supposer que les demandes et les largeurs soient telles que chaque variable entière puisse être transformée en 3 variables bivalentes, un problème à 15 largeurs mènerait à un problème à 360 variables bivalentes. (Ces chiffres correspondent à un seul code-acier : pour les problèmes réels, où l'on peut raisonnablement se fixer environ 20 codes différents et 15 largeurs et peut calculer que la combinatoire est en fait de 3 120 variables entières et de 720 contraintes (*voir* [4]).)

Un des problèmes posés par la formulation en variable bivalentes, et qui ne semble pas avoir été constaté dans la littérature est que ce genre de formulation mène à examiner une combinatoire plus importante qu'en résolvant le problème directement en nombres entiers.

En effet, supposant par exemple que l'on ait implicitement $x \leqq 4$, on écrira :

$$x = \delta_1 + 2\delta_2 + 4\delta_3,$$

laissant ainsi la possibilité de générer des solutions inutiles $x = 5, 6$ ou 7. On peut bien sûr éviter ceci mais au prix de l'insertion de la contrainte $x \leqq 4$ sous forme explicite ([5]).

Pour cette raison on donne dans [19] les moyens de réaliser directement des explorations combinatoires en résolvant directement ces problèmes en nombres entiers ([6]).

Variantes du problème

Lorsque les « chutes » sur certaines largeurs ne présentent aucun ennui (demande future non nulle pour ces chutes, avec un délai suffisamment court pour que les coûts de stockage soient négligeables) le problème PM se formule :

$$\underline{\min} \sum_I \lambda_i / \text{sous les contraintes de PM},$$

avec I = sous-ensemble de largeurs pour lesquelles il est important de minimiser les chutes.

Il est facile de voir que le problème se ramène encore à un problème en nombre entier. Par exemple, dans notre exemple à 3 largeurs où seules les chutes de la largeur 1 comptent, la fonction économique s'écrira :

$$\underline{\min}\ x_{11} + \frac{1}{3}x_{12} + \frac{1}{4}x_{13}.$$

On peut aussi facilement tenir compte de « coûts » c_i attachés aux chutes relativement aux demandes d_i. Le problème se formule encore sous la forme d'un

([5]) Ce qui implique, pour le problème P à m largeurs de rajouter $m(m+1)/2$ contraintes aux m contraintes initiales.

([6]) Il est intéressant de noter que mis à part les formalismes du type Gomory où l'on utilise la méthode du simplexe, la quasi-totalité des publications se réfèrent à des formalismes en variables 0-1.

problème purement en nombre entier. Par exemple, pour des coûts $c_1 \geqq 0$, $c_2 = 0$, $c_3 \geqq 0$, l'exemple à 3 largeurs aura pour critère :

$$\underline{\min}\ c_1 x_{11} + c_3 x_{33} + \frac{1}{3} c_1 x_{12} + \left(\frac{1}{4} c_1 + \frac{3}{4} c_3\right) x_{13} + \frac{3}{5} c_3 x_{23}.$$

REMARQUES : Le problème central (P) peut se rapprocher d'autres types de problèmes :

– « the set covering problem » : min $cx/E\,x > b$, $x = 0, 1$, $E =$ matrice de 0 ou 1;

– « the matching problem » : max $cx/A\,x < b$, $x = 0, 1$, $A =$ matrice de 0 ou 1.

Certains modèles donnés ont été étudiés; par exemple le « matching problem » en contraintes d'égalité avec x entier positif (*cf.* Balinski-Spielberg [2], p. 22, 230 et 234; Garfinkel-Nemhauser [10], p. 77 et 288).

A ce stade, ayant formulé le problème central de la minimisation des chutes, nous allons maintenant tenir compte des réglages.

Généralisation au cas de la prise en compte des temps de réglage en sous-objectif.

D'après ce qui précède, nous pouvons expliciter toutes les solutions optimales pour un même code-acier (*cf.* tableau I). En faisant de même pour un ensemble de p code-aciers, on obtient des ensembles :

$$X^1, X^2, \ldots, X^p,$$

où X^i est l'ensemble de toutes les solutions optimales pour le code-acier i.

Les chutes étant minimisées, on peut chercher une combinaison :

$$(x^1, x^2, \ldots, x^p),$$

avec $x^i \in X^i$, telle que le vecteur :

$$x = x^1 + x^2 + \ldots + x^p,$$

ait le maximum de composantes nulles, ce qui correspond à un minimum de réglages pour satisfaire les demandes.

Le problème étant vu sous l'angle :

– critère prioritaire : minimiser les chutes;

– critère secondaire : minimiser le nombre de réglages,

la solution x que nous venons de définir résout le problème.

La programmation des coulées peut se faire alors comme suit :

– étant donné i indice d'une composante non nulle de x, programmer à la suite :

$$x_i^1, x_i^2, \ldots, x_i^p;$$

– modifier alors le réglage i pour faire de même avec une nouvelle composante non nulle de x.

A titre d'exemple, soient les demandes suivantes pour 3 codes-acier.

TABLEAU II

Demandes pour 3 codes

	Codes		
Largeurs	1	2	3
1 : 90.	1,1	2,4	1
2 : 120.	2,5	1,2	0,4
3 : 140.	3,3	0,3	2

Les solutions optimales pour ces 3 codes figurent dans le tableau ci-après :

TABLEAU III

Solutions optimales pour 3 codes

Réglages	Code 1	Code 2 (7 solutions)	Code 3 (7 solutions)
11.	0 0	1 2 2 2 2 3 3	0 0 1 1 1 1 1
22.	0 2	0 0 0 1 1 1 1	0 0 0 0 0 0 1
33.	1 2	0 0 0 0 0 1 0	0 1 1 1 2 2 2
12.	2 1	3 1 2 1 1 1 0	0 1 0 0 0 1 0
13.	1 2	1 0 0 0 1 0 0	3 2 0 1 0 0 0
23.	3 0	0 2 1 1 0 0 1	1 0 2 1 1 0 0

Un exemple de combinaisons impliquant un minimum de réglages est donné dans le tableau suivant :

TABLEAU IV

Exemple de solution minimisant les chutes en priorité et le nombre de réglages en critère secondaire

Réglages	c_1	c_2	c_3	Réglages à retenir
11.	0	1	0	×
22.	0	0	0	
33.	1	0	0	×
12.	2	3	0	×
13.	1	1	3	×
23.	3	0	1	×

III. LA MINIMISATION DES RÉGLAGES

Dans cette 3e partie, nous mettons en critère prioritaire la minimisation du nombre d'interruptions dues à la nécessité de changer de réglages de largeurs.

Pour poser le problème, nous considérons 3 largeurs. Il y a 6 réglages possibles :

$$11, 22, 33, 12, 13, 23.$$

Pour 4 largeurs on a :

$$11, 22, 33, 44, 12, 13, 14, 23, 24, 34.$$

Pour n largeurs, il est facile de vérifier qu'il y a :

$$n(n+1)/2 \text{ réglages possibles.}$$

Pour $n=15$, on a donc 120 réglages possibles.

Supposons que 3 largeurs soient demandées. Les demandes peuvent *a priori* être satisfaites par les combinaisons suivantes de réglages :

$$11, 23,$$
$$12, 13,$$
$$12, 23,$$
$$12, 33,$$
$$13, 22,$$
$$13, 23.$$

Pour chacune de ces combinaisons, on vérifie bien que les 3 largeurs sont présentes. Ces combinaisons permettent donc *a priori* de satisfaire les demandes avec 2 réglages seulement.

Il y a cependant la possibilité de satisfaire ces demandes avec un plus grand nombres de réglages, par exemple la suite

$$(11, 13, 23, 33)$$

qui implique 4 réglages.

Dans [3] nous donnons un algorithme lexicographique pour la génération de toutes les familles exhaustives de combinaisons p à p de n objets.

A titre d'illustration, nous donnons ci-après (tableau V) toutes les familles de combinaisons avec répétition de 3 objets pris 2 à 2, telles que chaque objet soit

représenté dans la famille au moins une fois, chaque combinaison ne figurant qu'une fois dans la famille.

TABLEAU V

	11	23		11	12	13	22	
	12	13		11	12	13	23	
	12	23		11	12	13	33	
	12	33		11	12	22	23	
	13	22		11	12	22	33	
	13	23		11	12	23	33	
11	12	13		11	13	22	23	
11	12	23		11	13	22	33	
11	13	33		11	13	23	33	
11	13	22						
11	13	23		11	22	23	33	
				12	13	22	23	
11	22	23		12	13	22	33	
11	22	33		12	13	23	33	
11	23	33		12	22	23	33	
				13	22	23	33	
12	13	22						
12	13	23						
12	13	33	11	12	13	22	23	
12	22	23	11	12	13	22	33	
12	22	33	11	12	13	23	33	
12	23	33	11	12	22	23	33	
			11	13	22	23	33	
13	22	23	12	13	22	23	33	
13	22	33						
13	23	33	11	12	13	22	23	33

Pour 3 objets, il y a donc 35 familles différentes.

Dans le cas où le critère prioritaire est celui des interruptions, la solution est à rechercher dans les familles comportant le minimum de combinaison, ici les 6 premières familles ([7]).

Parmi ces 6 familles, on retient celle permettant de minimiser le critère secondaire de minimisation des chutes.

Le calcul est ici immédiat. Prenons par exemple la famille (11, 23). Pour chaque code-acier k tel que l'on ait des demandes dans l'une au moins de ces largeurs (si une largeur n'est pas demandée poser que la demande

([7]) Une méthode de résolution heuristique appliquée à la résolution de problèmes réels est donnée dans Boctor-Trémolières [4].

correspondante est nulle), on a à résoudre un problème du type :

$$P^k \quad \begin{cases} \underline{\min}\ x_{11}+ox_{22}+ox_{33}+ox_{12}+ox_{23}+x_{23}, \\ x_{11} \geqq d_1^k, \\ \dfrac{L_2}{L_2+L_3}x_{23} \geqq d_2^k, \\ \dfrac{L_3}{L_2+L_3}x_{23} \geqq d_3^k, \\ x_{11},\ x_{23} \text{ entiers} \geqq 0, \end{cases}$$

dont la solution est évidente :

$$x_{11}^k=[d_1^k]^+, \qquad x_{23}^k=(L_2+L_3)\max\{[d_2^k/L_2]^+, [d_3^k/L_3]^+\}$$

et l'on rappelle que les écarts par rapport aux seconds membres définissent les chutes.

De façon générale, étant donné une famille f d'interruptions :

$$f=[(i_1\ i_2), (i_3\ i_4), \ldots, (i_{r-1}\ i_r)]$$

et :

$$x_{i_1 i_2}^k, x_{i_3 i_4}^k, \ldots, x_{i_{r-1} i_r}^k,$$

une solution minimale de l'un des problèmes P^k correspondants ($k=, \ldots, p$), la chute totale pour une largeur i et pour tous les codes-aciers k est donnée par :

$$c_{\text{chute}}(L_i, f)=\sum_{k=1}^{p}[z_i^k-d_i^k]^+,$$

avec :

$$z_i^k=\sum_{\substack{s=1\\(i_s=i)}}^{r}\Big(\sum_{\substack{t=1\\(i_t<i)}}^{r}a_{i_s i_t}x_{i_t i_s}^k+x_{i_s i_s}^k+\sum_{\substack{t=1\\(i<i_t)}}^{r}a_{i_s i_t}x_{i_s i_t}^k\Big).$$

La chute totale pour toutes les largeurs L_i, $i=, \ldots, m$ est :

$$c_{\text{chute}}(f)=\sum_{i=1}^{m}c^{\text{chute}}(L_i).$$

A chaque famille f de réglages, on peut donc associer un « coût » $c_{\text{chute}}(f)$.

Si $\mathscr{F}_{\min}$ est l'ensemble de toutes les familles minimales d'interruption la solution correspondant au cas :

- critère prioritaire : minimiser les interruptions;
- critère secondaire : minimiser les chutes;

est donnée par la famille $f \in \mathscr{F}_{\min}$ telle que :

$$c_{\text{chute}}(f) \leqq c_{\text{chute}}(f'), \qquad \forall\ f' \in \mathscr{F}_{\min}.$$

Dans le cas où l'on peut associer un coût $c^k_{\text{chute}}(L_i)$ aux chutes de largeur L_i dans un code k, et un coût $c^k_{\text{manque}}(L_i)$ aux manques de largeur L_i dans un code k, la solution recherchée sera la famille minimale $f \in \mathscr{F}_{\min}$ telle que :

$$c(f) = c_{\text{chute}}(f) + c_{\text{manque}}(f),$$

est minimal : posant :

$$c_{\text{chute}}(f, x) = \sum_{k=1}^{p} c^k_{\text{chute}}(L_i)\,[z^k_i - d^k_i]^+,$$

$$c_{\text{manque}}(f, x) = \sum_{k=1}^{p} c^k_{\text{manque}}(L_i)\,[d^k_i - z^k_i]^+,$$

on définit $c(f)$ par :

$$c(f) = \min_{x \text{ entier} \geqq 0} [c_{\text{chute}}(f, x) + c_{\text{manque}}(f, x)],$$

ce qui correspond à un problème de minimisation en nombres entiers d'une fonction linéaire par morceaux. Le problème est séparable dans le cas où les coûts sont indépendants des réglages.

IV. PROBLÈME GÉNÉRAL

Dans le cas général, tous les critères sont ramenés à une unité financière : on définit :

$$c_{\text{inter}} = \text{coût d'une interruption pour réglage},$$

$$c^k_{\text{chute}}(L_i) = \text{coût de 300 t de chutes en largeur } i,$$

$$c^k_{\text{manque}}(L_i) = \text{coût de 300 t de manques en largeur } i.$$

A titre d'exemple, une première vision du problème donnait :

$$c_{\text{chute}} = 10 \text{ F}, \qquad c_{\text{manque}} = 50 \text{ F}, \qquad c_{\text{inter}} = 15\,000 \text{ F}.$$

([8]) On pourra remarquer que l'énumération des familles peut se faire implicitement.

Le problème peut se résoudre alors comme suit :

– Énumérer toutes les familles d'interruptions (cette énumération peut se faire implicitement); soit :

$$f = \{(i_1, i_2)\ldots(i_{r-1}\, i_r)\}$$

une famille d'interruptions.

– Pour chaque famille et pour l'ensemble des codes-aciers k, déterminer le nombre de poches $x^k_{i_s i_{s+1}}$ pour chaque réglage (avec $x^k_{i_s i_{s+1}} \geqq 1$) induisant un coût total c_{cm} des chutes et des manques minimal.

– Associer à la famille f un coût total :

$$c_T = (r-1)\, c_{inter} + c_{cm}.$$

Le problème général est alors résolu.

V. COMMENTAIRES

Il est clair que le problème est d'autant plus difficile à résoudre qu'on tient compte d'un plus grand nombre de particularités. La résolution est facilitée :

– si les coûts sont indépendants des codes;

– s'il y a un grand écart entre les coûts des manques et des chutes et celui des interruptions. Dans le cas où les premiers sont prépondérants, le modèle I (chap. II) est à retenir; dans le cas inverse, le modèle II (chap. III) est à choisir.

Il ne semble pas que le modèle général (chap. IV) puisse être résolu avec la technologie actuelle.

On remarquera que les modèles I et II reviennent à considérer des décompositions particulières du modèle général : ils peuvent être vus d'une certaine façon comme définissant des heuristiques pour la résolution du problème général.

En tous les cas, le problème présenté ici montre l'intérêt d'une approche « décomposée » des grands problèmes en nombres entiers.

BIBLIOGRAPHIE

1. K. G. AHLUWALIA et U. SAXENA, *Development of an Optimal Core Steel Slitting and Inventory Policy*, A.I.L.E. Trans., vol. 10, n° 4, Décembre 1978, p. 399-408.
2. M. L. BALINSKI et K. SPIELBERG, *Methods for Integer Programming : Algebraic, Combinatorial, and Enumerative*, chap. 7, in Progress in Op. Res., ARONOVSKY, éd., vol. III, Willey, N.Y., 1969.

3. F. F. BOCTOR et R. TRÉMOLIÈRES, *Combinaisons, permutations, arrangements et générations lexicographiques des familles exhaustives de combinaisons p à p*, Note I.A.E.-Aix, janvier 1980.
4. F. F. BOCTOR et R. TRÉMOLIÈRES, *Le problème de la coulée continue : approches optimales et heuristiques*, Note I.A.E.-Aix, mars 1980.
5. F. A. CARUSO et J. J. KOKAT, *Coil Slitting-a Shop Floor Solution*, Ind. Eng., novembre 1973, p. 18-23.
6. N. J. DRIEBEEK, *Applied Linear Programming*, Addison-Wesley, 1969.
7. K. EISEMANN, *The Trim Problem*, Man, Sc., vol. 3, 1957, p. 279-284.
8. T. FABIAN, *Blast Furnace Production Planning. A Linear Programming Example*, Man. Sc., vol. 14, n° 2, octobre 1967 p. 1-27; *A Linear Programming Model of Integrated Iron and Steel Production*, Man, Sc., vol. 4, n° 4, juillet 1958, p. 415-449.
9. P. C. GILMORE et R. E. GOMORY, *A Linear Programming Approach to the Cutting, Stock Problem*, Part I : Op. Res., vol. 9, 1961, p. 849-859; Part II : Op. Res., vol. 11, 1963, p. 863-888.
10. R. GARFINKEL et G. NEMHAUSER, *Integer Programming*, Wiley, 1972.
11. R. W. HAESSLER, *A Procedure for Solving the 1 . 5 Dimensional Coil Slitting Problem*, A.I.I.E. Trans., vol. 10, n° 1, mars 1978, p. 70-75.
12. R. W. HAESSLER et M. VONDEREMBSE, *A Procedure for Solving the Master slab Cutting Stock Problem in the Steel Industry*, A.I.I.E. Trans., vol. 11, n° 2, juin 1979.
13. J. C. MATHIEU et PAING, *Programmation d'opérations dans une uside sidérurgique à l'aide de la programmation linéaire et de la méthode du knapsack*, 6e Congrès A.F.I.R.O., mai 1967.
14. M. V. PHAFF, *Production Planning at Hoogovens Steel Using Linear Programming*, unpublished, Operationnel Research Department, Hoogovens steel BV, IJmuiden.
15. R. H. PHILLIPSON et A. RAVINDRAN, *Application of Mathematical Programming to metal cutting*, Math. Progr. (Study 11, Engineering, Optimization), octobre 1979.
16. C. N. REDWINE et D. A. WISMER, *A Mixed Integer Programming Model for Scheduling Orders in a Steel Mill*, J. Optim. Th. and Applic., vol. 14, n° 3, 1974, p. 305-319.
17. W. SHIH, *A Branch and Bond Method for the Multiconstraint* 0-1 *Knapsack Problem*, J. Opl. Res. Soc., vol. 30, n° 4, avril 1979, p. 369-378.
18. R. M. SUNDARAM, *An Application of Goal Programming Technique in Metal Cutting*, Int. J. of Prod. Res., vol. 16, n° 5, septembre 1978, p. 375-382.
19. R. TRÉMOLIÈRES, *Algorithmes lexicographiques pour la résolution directe de problèmes en nombres entiers à coefficients positifs*, Note I.A.E.-Aix, janvier 1980.
20. C. S. TSAO et R. H. DAY, *A Process Analysis Model of the U.S. Steel Industry*, Man. Sc., vol. 17, n° 10, juin 1971, p. 588 et suivantes.