N° d'ordre : 606

THÈSE

présentée à

l'UNIVERSITÉ DES SCIENCES ET TECHNIQUES
de LILLE I

pour obtenir le titre de

DOCTEUR DE SPÉCIALITÉ

(Traitement de l'Information)

par

Denise WERTHEIMER

LE PROBLEME D'IMPLANTATION A TROIS NIVEAUX :

UNE METHODE DE RESOLUTION EXACTE

THÈSE soutenue le 6 Septembre 1976
devant la commission d'examen

Monsieur	P. BACCHUS,	Président
Madame	M. GUIGNARD_SPIELBERG,	Rapporteur
Monsieur	P. HUARD,	Examinateur
Monsieur	J.P. STEEN,	Examinateur
Monsieur	K. SPIELBERG,	Invité

UNIVERSITE DES SCIENCES
ET TECHNIQUES DE LILLE

01.02.1976

DOYENS HONORAIRES de l'Ancienne Faculté des Sciences

MM. R. DEFRETIN, H. LEFEBVRE, M. PARREAU.

PROFESSEURS HONORAIRES des Anciennes Facultés de Droit et Sciences Economiques, des Sciences et des Lettres

M. ARNOULT, Mme BEAUJEU, MM. BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, CORSIN, DEHEUVELS, DEHORS, DION, FAUVEL, FLEURY, P. GERMAIN, HEIM DE BALSAC, HOCQUETTE, KAMPE DE FERIET, KOUGANOFF, LAMOTTE, LASSERRE, LELONG, Mme LELONG, MM. LHOMME, LIEBAERT, MARTINOT-LAGARDE, MAZET, MICHEL, NORMANT, PEREZ, ROIG, ROSEAU, ROUBINE, ROUELLE, SAVART, WATERLOT, WIEMAN, ZAMANSKI.

PRESIDENTS HONORAIRES DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

MM. R. DEFRETIN, M. PARREAU.

PRESIDENT DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

M. J. LOMBARD.

PROFESSEURS TITULAIRES

M.	BACCHUS Pierre	Astronomie
M.	BEAUFILS Jean-Pierre	Chimie Physique
M.	BECART Maurice	Physique Atomique et Moléculaire
M.	BILLARD Jean	Physique du Solide
M.	BIAYS Pierre	Géographie
M.	BONNEMAN Pierre	Chimie Appliquée
M.	BONNOT Ernest	Biologie Végétale
M.	BONTE Antoine	Géologie Appliquée
M.	BOUGHON Pierre	Algèbre
M.	BOURIQUET Robert	Biologie Végétale
M.	CELET Paul	Géologie Générale
M.	CONSTANT Eugène	Electronique
M.	DECUYPER Marcel	Géométrie
M.	DELATTRE Charles	Géologie Générale
M.	DELHAYE Michel	Chimie Physique
M.	DERCOURT Michel	Géologie Générale
M.	DURCHON Maurice	Biologie Expérimentale
M.	FAURE Robert	Mécanique
M.	FOURET René	Physique du Solide
M.	GABILLARD Robert	Electronique
M.	GLACET Charles	Chimie Organique
M.	GONTIER Gérard	Mécanique
M.	GRUSON Laurent	Algèbre
M.	GUILLAUME Jean	Microbiologie
M.	HEUBEL Joseph	Chimie Minérale
M.	LABLACHE-COMBIER Alain	Chimie Organique
M.	LANSRAUX Guy	Physique Atomique et Moléculaire
M.	LAVEINE Jean-Pierre	Paléontologie
M.	LEBRUN André	Electronique
M.	LEHMANN Daniel	Géométrie

Mme	LENOBLE Jacqueline	Physique Atomique et Moléculaire
M.	LINDER Robert	Biologie et Physiologie Végétales
M.	LOMBARD Jacques	Sociologie
M.	LOUCHEUX Claude	Chimie Physique
M.	LUCQUIN Michel	Chimie Physique
M.	MAILLET Pierre	Sciences Economiques
M.	MONTARIOL Frédéric	Chimie Appliquée
M.	MONTREUIL Jean	Biochimie
M.	PARREAU Michel	Analyse
M.	POUZET Pierre	Analyse Numérique
M.	PROUVOST Jean	Minéralogie
M.	SALMER Georges	Electronique
M.	SCHILTZ René	Physique Atomique et Moléculaire
Mme	SCHWARTZ Marie-Hélène	Géométrie
M.	SEGUIER Guy	Electrotechnique
M.	TILLIEU Jacques	Physique Théorique
M.	TRIDOT Gabriel	Chimie Appliquée
M.	VIDAL Pierre	Automatique
M.	VIVIER Emile	Biologie Cellulaire
M.	WERTHEIMER Raymond	Physique Atomique et Moléculaire
M.	ZEYTOUNIAN Radyadour	Mécanique

PROFESSEURS SANS CHAIRE

M.	BELLET Jean	Physique Atomique et Moléculaire
M.	BODARD Marcel	Biologie Végétale
M.	BOILLET Pierre	Physique Atomique et Moléculaire
M.	BOILLY Bénoni	Biologie Animale
M.	BRIDOUX Michel	Chimie Physique
M.	CAPURON Alfred	Biologie Animale
M.	CORTOIS Jean	Physique Nucléaire et Corpusculaire
M.	DEBOURSE Jean-Pierre	Gestion des entreprises
M.	DEPREZ Gilbert	Physique Théorique
M.	DEVRAINNE Pierre	Chimie Minérale
M.	GOUDMAND Pierre	Chimie Physique
M.	GUILBAULT Pierre	Physiologie Animale
M.	LACOSTE Louis	Biologie Végétale
Mme	LEHMANN Josiane	Analyse
M.	LENTACKER Firmin	Géographie
M.	LOUAGE Francis	Electronique
Mle	MARQUET Simone	Probabilités
M.	MIGEON Michel	Chimie Physique
M.	MONTEL Marc	Physique du Solide
M.	PANET Marius	Electrotechnique
M.	RACZY Ladislas	Electronique
M.	ROUSSEAU Jean-Paul	Physiologie Animale
M.	SLIWA Henri	Chimie Organique

MAITRES DE CONFERENCES (et chargés d'Enseignement)

M.	ADAM Michel	Sciences Economiques
M.	ANTOINE Philippe	Analyse
M.	BART André	Biologie Animale
M.	BEGUIN Paul	Mécanique
M.	BKOUCHE Rudolphe	Algèbre
M.	BONNELLE Jean-Pierre	Chimie
M.	BONNEMAIN Jean-Louis	Biologie Végétale
M.	BOSCQ Denis	Probabilités
M.	BREZINSKI Claude	Analyse Numérique
M.	BRUYELLE Pierre	Géographie

.../...

M. CARREZ Christian	Informatique
M. CORDONNIER Vincent	Informatique
M. COQUERY Jean-Marie	Psycho-Physiologie
Mle DACHARRY Monique	Géographie
M. DEBENEST Jean	Sciences Economiques
M. DEBRABANT Pierre	Géologie Appliquée
M. DE PARIS Jean-Claude	Mathématiques
M. DHAINAUT André	Biologie Animale
M. DELAUNAY Jean-Claude	Sciences Economiques
M. DERIEUX Jean-Claude	Microbiologie
M. DOUKHAN Jean-Claude	Physique du Solide
M. DUBOIS Henri	Physique
M. DYMENT Arthur	Mécanique
M. ESCAIG Bertrand	Physique du Solide
Me EVRARD Micheline	Chimie Appliquée
M. FONTAINE Jacques-Marie	Electronique
M. FOURNET Bernard	Biochimie
M. FORELICH Daniel	Chimie Physique
M. GAMBLIN André	Géographie
M. GOBLOT Rémi	Algèbre
M. GOSSELIN Gabriel	Sociologie
M. GRANELLE Jean-Jacques	Sciences Economiques
M. GUILLAUME Henri	Sciences Economiques
M. HECTOR Joseph	Géométrie
M. HERMAN Maurice	Physique Spatiale
M. JOURNEL Gérard	Physique Atomique et Moléculaire
Mle KOSMAN Yvette	Géométrie
M. KREMBEL Jean	Biochimie
M. LAURENT François	Automatique
Mle LEGRAND Denise	Algèbre
Mle LEGRAND Solange	Algèbre
M. LEROY Jean-Marie	Chimie Appliquée
M. LEROY Yves	Electronique
M. LHENAFF René	Géographie
M. LOCQUENEUX Robert	Physique Théorique
M. LOUCHET Pierre	Sciences de l'Education
M. MACKE Bruno	Physique
M. MAHIEU Jean-Marie	Physique Atomique et Moléculaire
Me N'GUYEN VAN CHI Régine	Géographie
M. MAIZIERES Christian	Automatique
M. MALAUSSENA Jean-Louis	Sciences Economiques
M. MESSELYN Jean	Physique Atomique et Moléculaire
M. MONTUELLE Bernard	Biologique Appliquée
M. NICOLE Jacques	Chimie Appliquée
M. PAQUET Jacques	Géologie Générale
M. PARSY Fernand	Mécanique
M. PECQUE Marcel	Chimie Physique
M. PERROT Pierre	Chimie Appliquée
M. PERTUZON Emile	Physiologie Animale
M. PONSOLLE Louis	Chimie Physique
M. POVY Lucien	Automatique
M. RICHARD Alain	Biologie
M. ROGALSKI Marc	Analyse
M. ROY Jean-Claude	Psycho-Physiologie
M. SIMON Michel	Sociologie
M. SOMME Jean	Géographie
Mle SPIK Geneviève	Biochimie
M. STANKIEWICZ François	Sciences Economiques
M. STEEN Jean-Pierre	Informatique

.../...

M.	THERY Pierre	Electronique
M.	TOULOTTE Jean-Marc	Automatique
M.	TREANTON Jean-René	Sociologie
M.	VANDORPE Bernard	Chimie Minérale
M.	VILLETTE Michel	Mécanique
M.	WALLART Francis	Chimie
M.	WERNIER Georges	Informatique
M.	WATERLOT Michel	Géologie Générale
Mme	ZINN-JUSTIN Nicole	Algèbre

Madame GUIGNARD-SPIELBERG m'a proposé le sujet de ce travail. Qu'elle veuille trouver ici l'expression de ma profonde gratitude pour ses conseils, ses encouragements et l'intérêt qu'elle m'a toujours témoigné.

Je remercie Monsieur le Professeur BACCHUS pour l'honneur qu'il me fait de présider le jury de cette thèse.

Que Monsieur le Professeur HUARD, Monsieur le Docteur SPIELBERG et Monsieur STEEN trouvent ici l'expression de mes remerciements pour avoir bien voulu accepter de faire partie du jury.

Que soient également remerciés les membres du Groupe de Recherche Opérationnelle du Laboratoire de Calcul et les membres du Centre Interuniversitaire du Traitement de l'Information pour les échanges amicaux qui ont jalonné ce travail.

Je tiens à remercier Madame et Monsieur DEBOCK qui ont assuré la réalisation matérielle de ce rapport et Mademoiselle DESQUIENS qui avec gentillesse et efficacité l'a dactylographié.

TABLE DES MATIÈRES

ERRATA

page	ligne	
10	-8	lire "et peuvent être louées ..."
17	-7	lire "Rappels sur les procédures d'exploration..."
25	-7	lire "on procède à une partition de J_L :"
36	8	lire [4]
50	-5	lire "prendre $\pi(y) = \delta s$
50	-7	lire "prendre $\pi(y) = \delta q$
64	3	lire "$X_1 \cap (J_1 \cup J_L)$
64	5 et 6	lire "$X_2 \cap (J_1 \cup J_L)$
73	6, 7 et 9	lire "$X_2 \cap (J_1 \cup J_{L1})$
73	8	lire "$X_1 \cap (J_1 \cup J_{L1})$
75	1	lire "min $cx + \sum_{i \in X_1} f_i y_i' + \sum_{j \in X_2} f_j y_j'$
82	6	lire "... $\leq Z^* - ZE_n$"
84	-2	lire "$\bar{c}_{ij} = \pi(i) + c_{ij} - \pi(j) \geq 0 \quad \forall (i,j) \in U,\ i \in X'$ "
85	2	lire "$(j,k) \in U$ ..."
86	14	lire " $c_{j_1 j_2} = \pi(j_2) - \pi(j_1)$
87	-2	lire "$\forall (i, j_1) \in U$" : $c - \bar{c}_{ij_1} > o$
91	5	lire "$s_{i_1 i_2} = +\infty$
115	9	lire "0. 0. 0. 0.09 0.52 0.35"

INTRODUCTION

La "State Enumeration Method" due à M. GUIGNARD et à K. SPIELBERG [17] est une méthode par séparation et évaluation séquentielle qui permet de résoudre les programmes linéaires en variables mixtes ; un tel programme s'écrit :

$$
\begin{array}{l}
\text{minimiser } cx + fy \\
\text{sous les conditions} \\
\qquad Ax + By \leq b \\
\qquad x_i \geq 0 \qquad\qquad \forall i \in I \\
\qquad y_j \in \{0,1\} \qquad \forall j \in J
\end{array}
$$

Nous proposons une adaptation de la méthode à une classe particulière de ces programmes : celle constituée par les programmes linéaires en variables mixtes, définis en relation avec des graphes.

Un programme linéaire en variables mixtes est dit *défini en relation avec un graphe* ou *défini sur ce graphe* si :

- les variables continues x_i sont en correspondance biunivoque avec les arcs du graphe sur lequel circule un flot ; ces variables continues sont alors les flux des arcs correspondants

- les variables booléennes y_j sont en correspondance avec les sommets du graphe ; elles traduisent l'existence ou la non-existence des sommets correspondants.

Nous restreignons notre étude aux programmes linéaires en variables mixtes définies sur des *graphes tripartis*.

Dans le chapitre I nous exposons le modèle mathématique auquel nous nous intéressons et nous formulons le problème correspondant.
Un bref tour d'horizon des méthodes de résolution des programmes en variables mixtes et une description rapide de la méthode de M. GUIGNARD et K. SPIELBERG constituent l'essentiel du second chapitre.

Lorsque les variables booléennes sont toutes fixées le programme à résoudre correspond au problème bien connu du "flot à coût minimal". Nous utilisons l'algorithme "out-of-kilter" de FULKERSON pour résoudre les sous-problèmes d'optimisation. Pour en raccourcir les temps de résolution nous proposons un algorithme de Recherche d'une Solution Initiale (algorithme R.S.I.).
Des rappels sur l'algorithme "out-of-kilter", la présentation de l'algorithme R.S.I. et des résultats numériques sont donnés dans le chapitre III.

Dans le paragraphe A du chapitre IV nous déterminons un *programme auxiliaire* permettant de donner un minorant de la fonction économique du sous-problème initial considéré ; ce programme nous permet également de déterminer un "état" c'est-à-dire de prévoir l'existence ou la non-existence des sommets non encore fixés.

Nous proposons un *renforcement* de ce programme, renforcement qui améliore le premier minorant et qui augmente la validité des décisions prises dans la détermination de l'état.

Dans le paragraphe B nous calculons des *évaluations* permettant de rendre plus efficaces des contraintes de BENDERS de type 1.
En vue de raccourcir l'exploration de l'arborescence, des tests sont proposés au paragraphe C tandis que l'exposé d'une procédure heuristique termine le chapitre.

Ce travail se termine par l'énoncé de la méthode, un exemple d'application et des résultats numériques.

NOTATIONS ET DÉFINITIONS

1 $|I|$

nombre d'éléments de l'ensemble I

2) $G(X,U)$

graphe dont X est l'ensemble des sommets et U l'ensemble des arcs

3) (x,y)

représente un arc de G(X,U) d'origine x et d'extrémité y : y est le *successeur* de x et x est le *prédécesseur* de y

4) *graphe connexe*

un graphe G(X,U) est connexe si : $\forall x$, $\forall y (x \neq y)$ il existe au moins une chaîne de x à y

5) *sous-graphe*

G(X',U') est un sous-graphe de G(X,U) si :

- $X' \subset X$
- $U' = \{(x,y) \in U : x \text{ et } y \in X'\}$

6) *réseau*

un graphe G(X,U), sans boucle, connexe est un réseau :

- s'il existe au moins un $x_o \in X$ tel que x_o n'ait pas de prédécesseur
- s'il existe au moins un $x_n \in X$ tel que x_n n'ait pas de successeur

7) *réseau réduit*

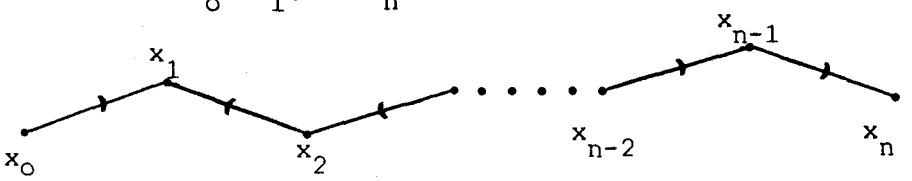
si x_o et x_n sont uniques le réseau est dit réseau réduit

8) *réseau réduit fermé*

si l'on introduit un arc de retour entre x_n et x_o, le réseau est dit réseau réduit fermé

9) *arc-avant et arc-arrière d'une chaîne*

soit une chaîne $x_o, x_1, \ldots, x_n$



l'arc (x_o, x_1) est un arc-avant de la chaîne

l'arc (x_1, x_2) est un arc-arrière de la chaîne

10) *flot compatible*

ϕ est un flot compatible si $\forall \phi_i : b_i \leq \phi_i \leq c_i$ où b_i (resp. c_i) est la limite inférieure (resp. supérieure) du flux ϕ_i pouvant circuler sur l'arc correspondant

11) *arborescence*

un graphe G(X,U) est une arborescence de racine x_o si et seulement si :

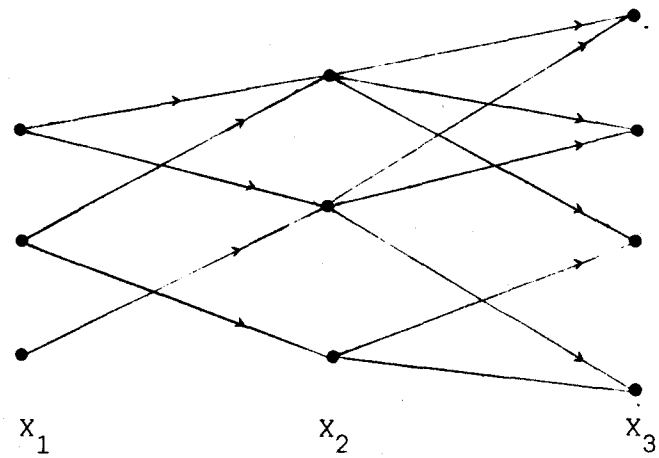
- il n'existe qu'un sommet $x_o \in X$ tel que x_o n'ait pas de prédécesseur
- $\forall x_i \in X$, $x_i \neq x_o$, x_i n'ait qu'un seul prédécesseur
- le graphe ne contient aucun circuit

CHAPITRE I

Nous allons décrire dans ce chapitre le *modèle mathématique* auquel nous nous intéressons ; nous préciserons ensuite les notations qui seront les nôtres jusqu'à la fin de cette rédaction, ce qui nous permettra de donner la *formulation* du problème étudié. Nous terminerons par des *exemples concrets* répondant à notre modèle.

A . MODÈLE MATHÉMATIQUE

1 - Le graphe



Ce graphe G(X,U) est un graphe *tri-parti* ; c'est-à-dire que l'ensemble X est composé de la partition (X_1,X_2,X_3), les arcs étant de la forme :

$$(x,y) \in U \text{ avec } x \in X_i \text{ et } y \in X_{i+1},\ i = 1 \text{ ou } 2$$

Nous appellerons :

X_1 l'ensemble des entrées ou des *origines*

X_2 l'ensemble des *transits*

X_3 l'ensemble des sorties ou des *destinations*.

Des variables booléennes sont attachées aux *origines* et aux *transits* ; elles traduisent l'*existence* ou la *non-existence* des sommets correspondants.

2 - Circulation dans le graphe

Le long de ce graphe, dans le sens des arcs, circule un flot qui est "produit" par les origines et "consommé" par les destinations, en passant par les transits.

Cette circulation est soumise à

- des *contraintes de conservation* de flot en chaque transit

- des *contraintes de capacité*
  le flux $\phi(x,y)$ circulant sur chaque arc est tel que

$$i(x,y) \leq \phi(x,y) \leq s(x,y)$$

où $i(x,y)$ est la quantité minimum pouvant circuler sur l'arc, tandis que $s(x,y)$ en est la quantité maximum ou *capacité*.

Les *flux* des différents arcs seront les *variables continues* du problème.

3 - Fonction économique

Il faut distinguer :

- les coûts *fixes*
  ils sont liés à l'existence ou à la non-existence de certains sommets.

- les coûts *variables*

  * coût de *transport* : c'est le coût engendré par un flux circulant sur un arc ; on peut supposer, en première approximation, que ce coût est linéaire, ce qui n'est pas toujours exact dans la réalité. Il peut arriver que la fonction c représentant ce coût soit :

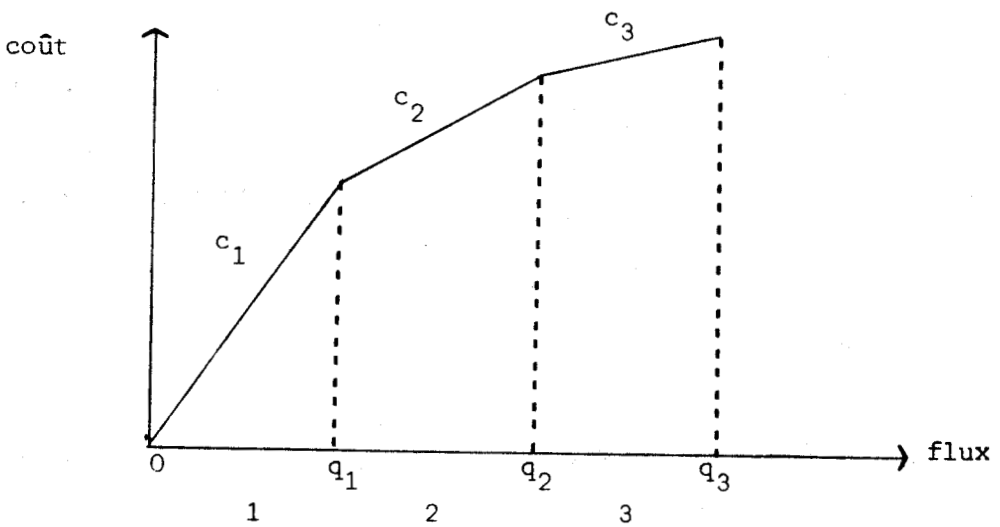
    - linéaire par morceaux

    - monotone non décroissante

- et telle que $c_i \geq c_{i+1}$ $i=0,1,2,\ldots,p$
  où c_i représente la pente de la portion de la fonction c comprise entre les points d'abscisses q_{i-1} et q_i et où n représente le nombre de tranches. (fig. 1).

Economiquement cela correspond à ce que l'on appelle un *tarif dégressif*.

* coûts *divers* proportionnels aux flux



La *fonction économique du problème* sera la somme :

- des frais fixes

- des frais variables.

4 - Objectif

Il est de minimiser la fonction économique compte-tenu des contraintes de flot et des contraintes de capacité.

Remarque 1

Nous avons *supposé* tout au long de notre travail que la quantité minimum pouvant circuler sur chaque arc est nulle. Nous signalons que quelques petites modifications permettraient de s'affranchir de cette hypothèse.

Remarque 2

L'introduction de variables booléennes attachées aux arcs permet de prendre en compte la non-linéarité des fonctions c.

Soit $\phi(x,y)$ le flux circulant sur l'arc (x,y) ; supposons que le graphe de la fonction c ait la forme de la fig. 1, n étant le nombre de tranches.

Posons :

$$(1) \qquad \phi(x,y) = \phi_1 + \phi_2 + \ldots + \phi_n$$

$$(2) \qquad 0 \leq \phi_i \leq q_i - q_{i-1} \qquad i=1,\ldots,n$$

où chaque nouvelle variable ϕ_i représente la fraction de flux prise dans la tranche i.

On désire :

$$(3) \qquad \phi_\ell \neq 0 \Longrightarrow \phi_i = q_i - q_{i-1} \qquad \forall i=1,\ldots,\ell-1$$

Pour que ceci soit réalisé on introduit des variables booléennes y_i :

$$(4) \qquad \sum_{i=1}^{i=n} y_i = 1$$

les conditions (1) et (2) deviennent :

$$(1') \qquad \phi(x,y) = \phi_1 + \phi_2 + \ldots + \phi_n + \sum_{i=1}^{n} q_{i-1}\, y_i$$

$$(2') \qquad 0 \leq \phi_i \leq (q_i - q_{i-1})\, y_i$$

La relation (3) est alors vérifiée ; en effet :

$$(4) \qquad \Longrightarrow \exists \ell \text{ unique} : 1 \leq \ell \leq n : y_\ell = 1$$

$$(2') \Longrightarrow \begin{cases} \phi_i = 0 \quad \forall i \neq \ell \\ 0 \leq \phi_\ell \leq q_\ell - q_{\ell-1} \end{cases}$$

$$(3') \Longrightarrow \quad \phi(x,y) = q_{\ell-1} + \phi_\ell$$

Remarque 3

Nous avons *supposé*, dans la suite de notre travail, que les coûts de transport étaient *linéaires* ;

- en effet lorsque l'on compare, à l'optimum, le pourcentage des frais fixes pris par rapport à la somme totale, au pourcentage des frais de transport pris également par rapport à la somme totale, on peut considérer cette hypothèse comme valable.

- supposons que l'on introduise des variables booléennes attachées aux arcs et que l'on décide de résoudre le problème par une méthode de séparation et d'évaluation ; le choix d'une variable à séparer devra porter simultanément sur ces variables et sur celles introduites par l'existence ou la non-existence de certains sommets.
  Dans la plupart des cas concrets correspondants au modèle indiqué, pour que la décision soit prise de séparer sur une variable d'arc il faudrait que les coûts dégressifs soient très fortement dégressifs. Ce qui n'est généralement pas le cas dans la réalité.

En première approximation il semble donc valable de retenir l'hypothèse ci-dessus et ensuite, éventuellement d'affiner la solution optimale en tenant compte des tarifs dégressifs.

B . FORMULATION DU PROBLÈME

Notations

$m = |X_1|$: nombre d'origines

$p = |X_2|$: nombre de transits

$n = |X_3|$: nombre de destinations

i désigne une origine

j désigne un transit

k désigne une destination

$I(j) = \{i \in X_1 : (i,j) \in U\}$

$K(j) = \{k \in X_3 : (j,k) \in U\}$

$J(i) = \{j \in X_2 : (i,j) \in U\}$

$J(k) = \{j \in X_2 : (j,k) \in U\}$

Pour une période donnée

$\underline{P}_i$: quantité minimum de flux produit par l'origine i

$\bar{P}_i$: quantité maximum de flux produit par l'origine i

$\underline{D}_k$: quantité minimum absorbée par la destination k

$\bar{D}_k$: quantité maximum absorbée par la destination k

$\underline{Q}_j$: quantité minimum de flux pouvant passer par le transit j

$\bar{Q}_j$: quantité maximum de flux pouvant passer par le transit j

x_{ij} : flux circulant sur l'arc (i,j)

c_{ij} : coût unitaire comprenant un coût unitaire de production et un coût unitaire de circulation sur l'arc (i,j)

s_{ij} : capacité de l'arc (i,j)

x_{jk} : flux circulant sur l'arc (j,k)

c_{jk} : coût unitaire comprenant un coût unitaire de passage au transit j et un coût unitaire de circulation sur l'arc (j,k)

s_{jk} : capacité de l'arc (j,k)

y_i : variable booléenne prenant la valeur 1 si l'origine i existe et la valeur zéro sinon

y_j : variable booléenne prenant la valeur 1 si le transit j existe et la valeur zéro sinon

f_i : coût fixe dû à l'existence de l'origine i

f_j : coût fixe dû à l'existence du transit j.

Formulation du problème

$$\left[\begin{array}{ll}
 & \min \sum_{i \in X_1} (f_i + \sum_{j \in J(i)} c_{ij} x_{ij}) + \sum_{j \in X_2} (f_j + \sum_{k \in K(j)} c_{jk} x_{jk}) \\
(1) & y_i \underline{P}_i \leq \sum_{j \in J(i)} x_{ij} \leq y_i \bar{P}_i \qquad \forall i \in X_1 \\
(2) & \underline{D}_k \leq \sum_{j \in J(k)} x_{jk} \leq \bar{D}_k \qquad \forall k \in X_3 \\
(3) & \sum_{i \in I(j)} x_{ij} - \sum_{k \in K(j)} x_{jk} = 0 \qquad \forall j \in X_2 \\
(4) & y_j \underline{Q}_j \leq \sum_{i \in I(j)} x_{ij} \leq y_i \bar{Q}_j \qquad \forall j \in X_2 \\
(5) & \begin{cases} 0 \leq x_{ij} \leq s_{ij} & \forall i \in X_1,\ j \in J(i) \\ 0 \leq x_{jk} \leq s_{jk} & \forall k \in X_3,\ j \in J(k) \end{cases} \\
(6) & y_\ell \in \{0,1\} \qquad \forall \ell = i \in X_1 \text{ et } \forall \ell = j \in X_2
\end{array}\right.$$

- l'ensemble des contraintes (1) traduit le fait que si $y_i=0$ aucune quantité de "produit" ne sort de l'origine i, alors que si $y_i=1$ il peut en sortir au minimum $\underline{P}_i$ et au maximum $\bar{P}_i$

- l'ensemble des contraintes (2) signifie que toutes les destinations doivent recevoir au minimum $\underline{D}_k$ et au maximum $\bar{D}_k$

- l'ensemble des contraintes (3) indique que toute la quantité de "produit" entrant dans le transit j en sort intégralement

- l'ensemble des contraintes (4) montre qu'un transit j ne peut être utilisé que si au moins une certaine quantité $\underline{Q}_j$ y transite, cette quantité ne pouvant dépasser la capacité $\bar{Q}_j$ de ce transit

- les contraintes (5) sont des contraintes de bornes sur les variables x_{ij} et x_{jk}

- les contraintes (6) sont des conditions d'intégrité sur les variables y_ℓ.

Le programme linéaire P est un programme linéaire en variables mixtes pouvant se mettre sous la forme

$$\text{PLM} = \begin{cases} \min\ cx + fy \\ Ax + By \leqslant b \\ x \geqslant 0 \\ y \text{ variable booléenne} \end{cases}$$

La matrice A a une structure particulière : elle n'a que six éléments non nuls par colonne ; le nombre de ces éléments non nuls pouvant être ramené à cinq si l'on fait rentrer l'ensemble des contraintes (5) dans des contraintes de bornes, c'est-à-dire si l'on met le programme P sous la forme

$$\text{PLMB} = \begin{cases} \min\ cx + fy \\ Ax + By \leqslant b \\ 0 \leqslant x \leqslant \alpha \\ y \text{ variable booléenne} \end{cases}$$

C . EXEMPLES CONCRETS

1 - Localisation d'usines et d'entrepôts

- une entreprise déjà en place sur le marché envisage l'ouverture de nouvelles usines et de nouveaux entrepôts

- une entreprise désire s'implanter sur le marché.

Dans les deux cas le problème est le suivant :

"pendant une période déterminée des usines fournissant des marchandises doivent satisfaire la demande d'un nombre donné de clients. Le circuit d'approvisionnement nécessite l'implantation d'entrepôts. A chacune des usines et à chacun des entrepôts est attaché un prix fixe d'ouverture (investissements-gestion) indépendant de la quantité produite ou de la quantité qui transite.
Parmi des sites possibles quels doivent être ceux retenus pour les usines et ceux retenus pour les entrepôts pour que le coût total de ce problème de distribution, compte-tenu des coûts fixes dûs à l'ouverture des usines et des entrepôts, soit minimum ?"

On peut donner une *interprétation économique de certaines notations* :

$I(j)$: ensemble des usines pouvant desservir l'entrepôt j

$K(j)$: ensemble des clients pouvant être servis à partir de l'entrepôt j

$J(i)$: ensemble des entrepôts pouvant être desservis à partir de l'usine i

$J(k)$: ensemble des entrepôts pouvant servir le client k.

$\underline{P}_i$: production minimum de l'usine i

$\bar{P}_i$: production maximum de l'usine i

$\underline{D}_k$: demande minimum du client k

$\bar{D}_k$: demande maximum du client k

$\bar{Q}_j$: capacité de stockage de l'entrepôt j

x_{ij} : quantité de marchandises fournies par l'usine i à l'entrepôt j

c_{ij} : coût unitaire comprenant le coût unitaire de production de l'usine i et le coût unitaire de transport de l'usine i à l'entrepôt j

s_{ij} : capacité de transport des moyens utilisés entre l'usine i et l'entrepôt j

x_{jk} : quantité de marchandises fournies par l'entrepôt j au client k

c_{jk} : coût unitaire comprenant le prix unitaire de déchargement -stockage- chargement au transit j et le coût unitaire de transport de l'entrepôt j au client k

s_{jk} : capacité de transport des moyens utilisés entre l'entrepôt j et le client k

f_i : coût fixe dû à l'ouverture de l'usine i

f_j : coût fixe dû à l'ouverture de l'entrepôt j

Remarque

C'est cet exemple qui nous fera souvent appeler par la suite, dans le modèle mathématique proposé,

les origines : des *usines*
les transits : des *entrepôts*
les destinations : des *clients*

2 - Implantation d'un réseau téléinformatique

Dans un réseau multipoint un ou plusieurs ordinateurs centraux sont li reliés chacun à un certain nombre de concentrateurs gérant eux-mêmes des terminaux variés.

Les lignes de transmission ont une capacité limite et peuvent être bornées à des tarifs proportionnels à leur longueur et parfois dégressifs.

Le coût d'exécution d'un programme par un ordinateur central dépend d'une part de cet ordinateur et d'autre part de la nature du travail (scientifique ou de gestion, par exemple). En outre l'unité centrale peut être saturée si des travaux trop importants ou trop nombreux lui sont confiés. Le coût du logiciel de gestion des transmissions varie suivant le type d'ordinateur, de concentrateur (cablé ou programmé) et de terminal.

En fonction :

- des travaux à réaliser
- de la charge du réseau
- des unités de traitement

il peut être intéressant de déterminer :

- quelles sont les liaisons à établir
- par quelles unités centrales les différents travaux doivent être exécutés
- par quel chemin faire transiter les informations.

On peut alors considérer que les ordinateurs centraux sont des "usines" produisant à un certain coût les résultats de travaux, que les concentrateurs sont des intermédiaires qui ventilent les résultats vers des "clients" terminaux.

Chaque participant du réseau demande donc une certaine "quantité" d'un type de travail (il reste à imaginer l'unité de mesure) dont le coût est commun. En général tout le travail demandé par un terminal sera effectué par le même ordinateur mais on peut aussi envisager qu'il soit partagé entre plusieurs participants.

CHAPITRE II

A . MÉTHODE DE BENDERS [5]

Considérons le programme mixte

$$P_1 \begin{cases} \min cx + fy \\ Ax + By = b \\ x \geq 0 \\ y_k = \{0,1\} \quad k = 1,2,\ldots,n \end{cases}$$

J.F. BENDERS a montré que la résolution d'un programme linéaire mixte pouvait être ramenée à la résolution alternée de deux programmes associés :

- l'un est un programme linéaire dont les variables sont astreintes à être entières
- l'autre est un programme linéaire dont les variables sont continues.

Notons que l'approche proposée par J.F. BENDERS peut s'appliquer à des cas plus généraux que celui présenté ci-dessus.

Pour un y fixé, $y=\bar{y}$, le programme P_1 devient un programme linéaire en variables continues x

$$P(\bar{y}) \begin{cases} \min cx + f\bar{y} \\ Ax = b - B\bar{y} \\ x \geq 0 \end{cases}$$

Considérons le dual $D(\bar{y})$ du programme $P(\bar{y})$

$$D(\bar{y}) \begin{cases} \max u(b - B\bar{y}) \\ uA \leq c \\ (u) \end{cases}$$

L'ensemble des solutions réalisables de $\underline{D(\bar{y})}$ *ne dépend pas de* $\bar{y}$.

Soit $E = \{u : uA \leq 0\}$

$L = \{u_\ell\}$ l'ensemble des points extrêmes de E

$K = \{u_k\}$ l'ensemble des rayons extrémaux du cône polyédrique convexe défini par :

$CP = \{u : uA \leq 0\}$

les rayons extrémaux sont parallèles aux arêtes "infinies" de E et chaque rayon u_k n'est défini qu'à un scalaire près.

On montre l'équivalence du programme initial P_1 et du programme

$$P_2 \begin{cases} \min\limits_{y} (fy + \min\limits_{x} cx) \\ Ax = b - By \\ x \geq 0 \\ y_j = \{0,1\} \quad j = \quad j = 1,2,\ldots,n \end{cases}$$

lui-même équivalent au programme

$$P_3 \begin{cases} \min\limits_{y} [fy + \max\limits_{u} (u(b - By))] \\ uA \leq 0 \\ (u) \end{cases}$$

Résolution du programme $D(\bar{y})$

Le théorème fondamental de la dualité permet de dire que si :

- $E = \emptyset$ alors $P(\bar{y})$ n'a pas de solution optimale finie et P_2 non plus
- $E \neq \emptyset$ et si $D(\bar{y})$ a une solution optimale finie alors le maximum est atteint en un point u_ℓ de L
- $E \neq \emptyset$ et si $D(\bar{y})$ n'a pas de solution à distance finie alors il en est de même pour $P(\bar{y})$ et pour P_2.

Or ce cas se rencontre si, pour $y=\bar{y}$ fixé, il existe un rayon u_k tel que $u_k(b - By) > 0$.
Réciproquement si pour $y=\bar{y}$ fixé : $u_k(b - By) \leq 0 \quad \forall u_k$ alors $u(b - B\bar{y})$ ne peut pas croître vers $+\infty$.

Donc $D(\bar{y})$ aura une solution à distance finie si et seulement si

- $E \neq \emptyset$

- $u_k(b - B\bar{y}) \leq 0 \quad \forall u_k \in K$.

Si ceci est vérifié alors l'optimum de $D(\bar{y})$ est atteint en un point extrême de E.

On peut formuler le problème sous la forme :

$$P_4 \begin{cases} \min\limits_{y} [fy + \max\limits_{u \in L} (b - By)] \\ u_k(b - By) \leq 0 & \forall u_k \in K \\ y_k \in \{0,1\} & k = 0,1,2,\ldots,n \end{cases}$$

puis sous la forme :

$$P_5 \begin{cases} \min\limits_{y} z \\ z \geq fy + \max\limits_{u_1 \in L} [u_1(b - By)] \\ u_k(b - By) \leq 0 & \forall u_k \in K \\ y_k \in \{0,1\} & k = 1,2,\ldots,n \end{cases}$$

On démontre l'*équivalence* des programmes P_1 et P_5. [12]

Dans le programme P_5 ne figurent que les inconnues y et z.
Si nous connaissions à priori :

- L ensemble des points extrêmes

- K ensemble des rayons extrémaux

de l'ensemble E = {u : uA ≤ c} nous pourrions construire toutes les contraintes du programme P_5 et résoudre ce programme dont la solution y* nous permettrait de trouver x* solution optimale de P(y*) ; la solution (x*, y*) serait alors la solution optimale de P_1.

L'algorithme de BENDERS permet d'engendrer, au fur et à mesure, des points extrêmes et des rayons extrêmes sans qu'il soit nécessaire de les engendrer tous pour trouver la solution optimale de P_1.

Le programme P_5 est équivalent au programme P_6 où z *joue le rôle d'une variable d'écart.*

$$P_6 \quad \begin{cases} \min\limits_{y} z & \\ (1) \quad (u_\ell B - f)\, y + z \geq u_\ell b & \forall u_\ell \in L \\ (2) \quad u_k B y \geq u_k b & \forall u_k \in K \\ y_j \in \{0,1\} & j=1,\ldots,n \end{cases}$$

Description d'une itération

Supposons que nous ayons déjà engendré :

- $L_1 \subset L$

- $K_1 \subset K$

Soit P_6' le programme P_6 restreint aux sous-ensembles L_1 et K_1 ; les contraintes de ce programme s'écrivent :

$$(1') \quad (u B - f)\, y + z \geq u_\ell b \qquad \forall u_\ell \in L_1$$

$$(2') \quad u_k B y \geq u_k b \qquad \forall u_k \in K_1$$

Supposons que le programme P_6' :

- n'ait aucune solution à distance finie ; alors P_1 n'en a pas non plus et l'algorithme est terminé

- ait une solution réalisable $\bar{y}$ et que $\bar{z}'$ soit la valeur correspondante de sa fonction économique. On résoud alors le problème $D(\bar{y})$:

  - l'optimum de $D(\bar{y})$ peut ne pas être fini puisque les contraintes (2') de P_6' ne contiennent pas toutes les contraintes (2) de P_6. Il existe alors un u_{k_0} qui appartient à K et qui n'appartient pas à K_1 ; on ajoute cette contrainte au programme P_6' et l'on itère le processus

  - l'optimum de $D(\bar{y})$ a une solution finie $\bar{u}$; on en déduit $\bar{x}$ donc $c\bar{x} = \bar{u}(b - B\bar{y})$ et $\underline{z} = c\bar{x} + f\bar{y}$

    Remarque

    $\underline{z}$ est une *borne inférieure* de $z^* = cx^* + fy^*$ où (x^*, y^*) est la solution optimale du programme P_1 tandis que $\bar{z}'$ en est une *borne supérieure*

    - si $\underline{z} = \bar{z}'$
      nous avons obtenu la solution optimale du programme P_1

    - si $\underline{z} > z'$
      cela prouve que l'inéquation

      $$z + (\bar{u}B - f)\,\bar{y} \geq \bar{u}b$$

      n'est pas vérifiée pour $z=\bar{z}$ et $y=\bar{y}$ et que, par conséquent, elle ne fait pas partie des contraintes antérieures : on l'ajoute et on itère le processus.

On montre que la suite $(\bar{z}')$ obtenue en cours d'algorithme est une suite monotone non croissante.

Les différentes contraintes engendrées sont dites *contraintes de BENDERS.*

- *Contraintes du premier type :*

$$(u_\ell B - f)\, y + z \geq u_\ell b \qquad \forall u_\ell \in L$$

- *Contraintes du second type :*

$$u_k(By - b) \geq 0 \qquad\qquad u_k \in K$$

Remarque

Si les calculs deviennent trop longs on peut les arrêter puisqu'à chaque itération on obtient une borne inférieure et une borne supérieure de z^*, ce qui permet une évaluation de l'écart entre la solution trouvée et la solution optimale.

On peut commencer l'algorithme en prenant :

$$y_k = 0 \qquad\qquad k = 1,2,\ldots,n$$

La procédure de BENDERS *nécessite* à chaque itération *la résolution d'un programme en nombres entiers* : P'_6. Pour éviter la résolution d'un tel programme, C.E. LEMKE et K. SPIELBERG [24] ont proposé d'utiliser une procédure d'optimisation par séparation sur les composantes du vecteur y.

B . RAPPELS SUR LES PROCÉDURES D'EXPLOITATION PAR SÉPARATION ET ÉVALUATION

Dans le cas où le problème étudié présente un caractère combinatoire ou discret il n'est plus possible d'utiliser un algorithme séquentiel classique, donnant une suite de solutions convergeant vers la solution optimale suivant une certaine métrique comme c'est le cas, par exemple, lorsque l'on traite un problème de programmation linéaire ou non linéaire en variables continues.

On utilise alors les méthodes dites *méthodes d'optimisation par séparation* ou *procédures arborescentes d'optimisation*. Ces méthodes consistent à examiner des sous-ensembles de plus en plus restreints de l'ensemble des solutions jusqu'à ce que l'on ait mis en évidence l'existence d'une solution optimale ou que l'on ait obtenu la certitude qu'il n'en existait pas.

Le premier algorithme de procédure par séparation semble être celui de LANG et DOIG [23] en 1960. Mais ce n'est qu'en 1963 que LITTLE, MURTY, SWEENY et KAREL ont donné l'algorithme célèbre sur le voyageur de commerce [25].

En 1964 BERTHIER et ROY ont formalisé la méthode en introduisant les *principes de séparation et d'évaluation* [7]

- le premier principe permet de construire l'arborescence par séparation : il repose sur des tests.

- le deuxième principe permet d'attacher une fonction d'évaluation par défaut (dans un problème de minimisation) de la fonction économique sur chacun des sous-ensembles construits par séparation.

Nous allons donner une description rapide :

- de la *Procédure* par *Séparation* et *Evaluation Progressive* (P.S.E.P)

- de la *Procédure* par *Séparation* et *Evaluation Séquentielle* (P.S.E.S)

I Procédure par Séparation et Evaluation Progressive (P.S.E.P)

Pour rester dans le cadre de notre problème considérons le programme

$$P \left| \begin{array}{l} \min z = cx + fy \\ Ax + By \leqslant b \\ x \geqslant 0 \\ y_j = \{0,1\}, \qquad j \in J = \{1,2,\ldots,n\} \end{array} \right.$$

qui est un programme linéaire en variables mixtes.

Soit E l'ensemble des solutions (x,y) réalisables de P. En un noeud courant γ de l'arborescence il y a des composantes de y fixées et d'autres non encore fixées : nous les dirons *libres*. Il existe donc une partition de l'ensemble J en

$$J_1 = \{j \in J : y_j = 1\}$$
$$J_0 = \{j \in J : y_j = 0\}$$
$$J_L = \{j \in \{J - J_1 \cup J_0\}\}$$

A une itération courante du déroulement de l'algorithme un noeud de l'arborescence peut se trouver dans l'un des trois états suivants :

- **état terminal**
  toutes les variables y_j sont fixées et l'on a alors obtenu une solution (x,y) du problème P ;
  le noeud ne peut plus être séparé

- **état séparé**
  le noeud est séparé mais non terminal ; il y a encore des variables y_j libres

- **état pendant**
  le noeud n'a pas encore été séparé.

Principe de séparation progressive

Ce principe revient à construire une suite de partitions $\bar{E}$ de E, ensemble des solutions réalisables de P, chacune d'elle se déduisant de la précédente par dichotomie d'un de ses éléments.

A une itération quelconque nous considérons un noeud pendant γ de l'arborescence ; en ce noeud la partition $(J_1^\gamma, J_0^\gamma, J_L^\gamma)$ est connue. La séparation va s'effectuer sur une variable y_{j^*} dont l'indice $j^* \in J_L$ est judicieusement choisi.

On obtient alors deux nouveaux noeuds γ_1 et γ_2 et deux nouvelles partitions de J :

$$\text{en } \gamma_1 : \begin{cases} J_1^{\gamma_1} = J_1^{\gamma} \cup \{j^*\} \\ J_0^{\gamma_1} = J_0^{\gamma} \\ J_L^{\gamma_1} = J_L^{\gamma} - \{j^*\} \end{cases}$$

$$\text{en } \gamma_2 : \begin{cases} J_1^{\gamma_2} = J_1^{\gamma} \\ J_0^{\gamma_2} = J_0^{\gamma} \cup \{j^*\} \\ J_L^{\gamma_2} = J_L^{\gamma} - \{j^*\} \end{cases}$$

Principe d'évaluation progressive

En chacun des nouveaux noeuds créés *nous calculons une évaluation par défaut du problème* P correspondant à la partition de J au noeud considéré. Pour simplifier appelons γ l'un ou l'autre des deux noeuds et J^{γ} la partition correspondante de J.

Nous devons donner une évaluation par défaut de la fonction économique du programme

$$P^{\gamma} \begin{cases} \min z = cx + fy & \\ Ax + By \leq b & \\ x \geq 0 & \\ y_j = 1 & \forall j \in J_1^{\gamma} \\ y_j = 0 & \forall j \in J_0^{\gamma} \\ y_j = \{0,1\} & \forall j \in J_L^{\gamma} \end{cases}$$

Si le programme P^{γ} est irréalisable nous prendrons $\underline{z}^{\gamma} = +\infty$.

Méthode

Soit z^* la valeur optimale du problème P cherchée. Si l'on ne connaît aucune solution réalisable prendre $z^* = +\infty$.

A une itération courante

1 - Chercher parmi tous les sommets non terminaux celui dont l'évaluation par défaut $\underline{z}$ est minimale

si $\underline{z} > z^*$ aller en 2

sinon opérer une séparation et procéder aux deux évaluations correspondantes ; un nouveau noeud crée peut être :

- **terminal :**
  éventuellement réajuster z^* et reprendre en 1

- **pendant :**
  reprendre en 1.

2 - Le problème est terminé.

Procédure par Séparation et Evaluation Séquentielle (P.S.E.S.)

Un noeud de l'arborescence est dit *fermé* :

- s'il est terminal

- ou si, à partir de lui, on ne peut plus développer d'arc, c'est-à-dire que l'on a la certitude qu'il n'existe aucun noeud succédant à ce noeud pouvant conduire à une solution réalisable meilleure que celle déjà obtenue.

Dans cette variante le sommet sélectionné n'est plus celui d'évaluation minimale comme dans la méthode précédente mais le dernier créé et non fermé.

L'application de cette procédure revient donc à décrire un chemin depuis la racine de l'arborescence jusqu'à ce que l'on rencontre un sommet terminal ou fermé : alors on remonte le chemin en sens inverse jusqu'à rencontrer un sommet non fermé ; on repart de ce dernier sommet pour décrire un nouveau chemin et ceci jusqu'à ce que l'on revienne à la racine de l'arborescence, la procédure est alors terminée.

L'arborescence a une structure un peu diffférente de celle décrite dans la méthode P.S.E.P. et elle porte souvent dans la langue anglo-saxone le nom de "*single-branch*" par opposition à la première appelée "*multi-branch*" [3].

C'est en 1965 que E. BALAS [1] a proposé le premier une procédure de séparation et d'évaluation séquentielle pour résoudre des programmes linéaires en variables bi-valentes. Des extensions et des améliorations ont été apportées par F. GLOVER [13], C. LEMKE et K. SPIELBERG [24], A.M. GEOFFRION [11] et E. BALAS [2] notamment.

Remarque

La méthode P.S.E.P. présente l'inconvénient de conduire lentement à une solution réalisable et nécessite un grand stockage d'informations qui exige souvent d'avoir recours à une mémoire externe à l'ordinateur, d'où une augmentation du temps de calcul.

La méthode P.S.E.S. présente l'avantage d'obtenir généralement plus rapidement une solution réalisable mais si la règle de choix utilisée pour les séparations est peu appropriée au problème à résoudre, elle conduit alors à de nombreux calculs inutiles avant de donner une solution voisine de l'optimum.

Il est recommandé de commencer les deux méthodes avec une solution réalisable obtenue à l'aide d'une procédure heuristique.

On peut encore proposer *une troisième méthode mixte* qui utilise tour à tour les deux méthodes précédentes :

- si l'on connaît une solution réalisable on commence par appliquer la procédure P.S.E.P pendant un certain temps, par exemple celui de résoudre un nombre fixé de sous-problèmes ou jusqu'à ce que le stockage des informations devienne trop important. On applique alors la procédure P.S.E.S au dernier sous-problème courant de la procédure précédente et ceci

pendant un certain temps ; on sélectionne alors un sous-problème courant de la liste des problèmes à résoudre par cette procédure et l'on réapplique la procédure P.S.E.P. et ainsi de suite

- si l'on ne connaît pas de solution réalisable on commence par appliquer la procédure P.S.E.S. et l'on procède comme ci-dessus.

Dans les procédures décrites ci-dessus, une séparation bien choisie d'une part et une "bonne" évaluation d'autre part permettent de raccourcir l'exploration de l'arborescence, soit en donnant rapidement une "bonne" solution réalisable, soit en abandonnant l'examen de certaines branches de l'arborescence.

Les notions de *relaxation*, *minoration*, *pénalité* et *test* permettent de calculer un minorant de la fonction économique, de décider de la variable sur laquelle doit s'opérer la séparation et d'accélérer l'exploration de l'arborescence.

Considérons un programme mathématique :

$$P \begin{cases} \min f(x) \\ x \in D \end{cases}$$

et les deux programmes :

$$P' \begin{cases} \min f(x) \\ x \in D' \end{cases} \qquad \text{où} \quad D' \supset D$$

$$P'' \begin{cases} \min g(x) \\ x \in D \end{cases} \qquad \text{où} \quad g(x) \leqslant f(x) \quad \forall x \in D$$

Nous dirons que le programme P' est une *relaxation* de P et que le programme P" en est une *minoration* sur D.

Les *relaxations* et *les minorations* permettent de calculer des minorants de la fonction économique.

Les *pénalités* peuvent se diviser en deux catégories :

- celles attachées à la fonction économique ;
  elles permettent également de construire un minorant de cette fonction économique

- celles attachées aux contraintes.

Les *tests* indiquent la façon de se déplacer dans l'arborescence ; ils portent sur les minorants et sur les pénalités.

Pour une bibliographie plus complète et pour une étude théorique sur les procédures par séparation nous signalons la thèse de P. HANSEN "Programmes mathématiques en variables 0-1". [18]

Il consacre le premier chapitre à ce problème ; l'auteur y répond notamment aux questions suivantes :

- quand un ensemble de solutions doit-il être séparé ?

- comment les séparations doivent-elles être faites ?

- comment s'enchaînent les séparations ?

- quelles sont les formes des tests utilisés ?

- quels sont les tests à sélectionner parmi un ensemble de tests possibles et dans quel ordre doivent-ils être appliqués ?

C . LA "STATE ENUMERATION METHOD" [17]

Cette méthode dont les principes avaient été tracés dans [16] est une procédure par séparation et évaluation séquentielle (P.S.E.S.).

Elle repose :

1) sur la *détermination d'un état* en chaque noeud créé de l'arborescence ; c'est une partition des indices des variables libres en ce noeud.

Soit γ le noeud considéré et les ensembles :

$$J_1^\gamma = \{j \in J : y_j = 1\}$$
$$J_0^\gamma = \{j \in J : y_j = 0\}$$
$$J_L^\gamma = \{j \in J : y_j = \{0,1\}\}$$

les deux premiers ensembles sont constitués par les variables fixées et le dernier par les variables libres. (J est l'ensemble des indices des variables booléennes).

On procède à une partition de J_ℓ :

$$J_{L1} : \{j \in J : y_j = 1\}$$
$$J_{L2} : \{j \in J : y_j = 0\}$$
$$J_{L3} : \{j \in J : y_j = \{0,1\}$$

Les affectations à zéro ou à un des variables libres ne sont que temporaires ; par exemple un y_j, $j \in J$, dont la valeur est fixée à un pourra valoir zéro en un noeud successeur du noeud γ.

2) sur des *inégalités engendrées en chaque noeud.* Elles sont déduites de la résolution de *problèmes auxiliaires* remplaçant le problème initial.

Cette méthode prend en compte la nature du problème ; l'exploitation des inégalités permettant de raccourcir la recherche, la manière dont celles-ci sont engendrées, c'est-à-dire le choix de l'état, est essentielle.

Nous allons préciser cette méthode.

Considérons le programme :

$$P \left[\begin{array}{ll} \min \delta\xi + \gamma\eta & \\ D\xi + C\eta \leqslant \beta & \\ \xi \geqslant 0 & \eta_j = \{0,1\},\ j \in J = \{1,2,\ldots,n\} \end{array} \right.$$

dans lequel on suppose que ni l'ensemble des variables continues, ni l'ensemble des variables booléennes ne sont vides.

Un noeud γ de l'arborescence est dit de *niveau* n si n composantes y_j ont été fixées par séparation, on dit aussi par *choix*.

Une *itération* de l'algorithme comprend :

- soit un "pas en avant" permettant de passer du noeud γ de niveau n à un niveau γ^+ de noeud n^+ : ceci se fait en fixant une variable η_{k*} à un ou à zéro ; ce choix est fait en fonction de l'état au noeud γ.

- soit un "pas en arrière" permettant de revenir au noeud γ^- de niveau n^- en fixant η_{k_*} à la valeur complémentaire de celle qu'il avait au noeud γ.

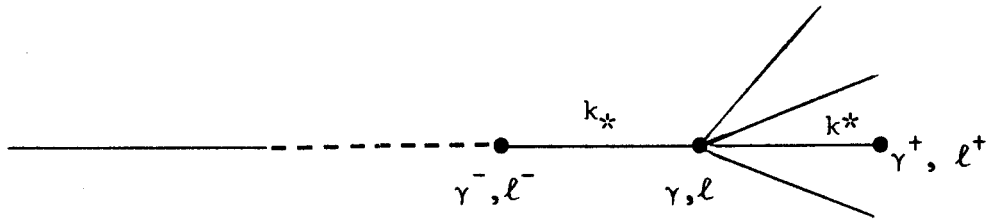


Fig. 1.

Sur la figure 1, les différentes branches, autres que celle de k*, partant du noeud γ, représentent tous les indices des variables libres.

Au noeud γ de niveau n considérons les ensembles :

$$J_1 = \{j \in J : \eta_j = 1\} \qquad \text{variables fixées}$$
$$J_0 = \{j \in J : \eta_j = 0\}$$
$$J_L = \{j \in J : \eta_j \in \{0,1\}\} \qquad \text{variables libres}$$

Remarque

Parmi les variables fixées il y en a n *fixées par choix* et éventuellement d'autres fixées *par tests*.

Soit P_n le programme initial P dans lequel toutes les variables fixées

$$\eta_j : j \in J_1 \cup J_0$$

ont été remplacées par leurs valeurs.

$$P_n \begin{cases} \min \delta\xi + \gamma\eta \\ D\xi + C\eta \leq \beta \\ \xi \geq 0 \qquad \eta_j \in \{0,1\},\ j \in J_L \end{cases}$$

Soit PA_n un programme auxiliaire obtenu par *relaxation* du programme P_n, la relaxation portant sur toutes les variables libres η_j, $j \in J_L$.

$$PA_n \begin{cases} \min \delta\xi + \gamma\eta \\ D\xi + C\eta \leq \beta \\ \xi \geq 0, 0 \leq \eta_j \leq 1 \qquad \forall j \in J_L \end{cases}$$

A) Détermination de l'état au noeud γ de niveau n

Il y a plusieurs moyens pour l'obtenir :

1 - le déduire de celui du noeud γ^- de niveau n^- par le transfert de k_* de J_n vers J_1 ou vers J_0 suivant la valeur de η_{k_*}

2 - le déduire de la solution du problème PA_n par une *procédure d'arrondi* par exemple.
Soit τ un nombre tel que :

$$0 < \tau < 1$$

on forme :

$J_{L1} = \{j : \eta_j > \tau\}$
$J_{L2} = \{j : \eta_j < \tau\}$
$J_{L3} = \emptyset$ ou est formé des variables qui montrent une tendance à vouloir rester fractionnaires au noeud γ et aux prédécesseurs du noeud γ.

On peut alors imposer :

$$n_j = 1 \qquad \forall j \in J_{L1}$$
$$n_j = 0 \qquad \forall j \in J_{L2}$$
$$0 \leq n_j \leq 1 \qquad \forall j \in J_{L3}$$

La difficulté est de choisir convenablement le paramètre τ.

3 - le déduire de la résolution du programme PA_n résolu par des méthodes telle que celle de GOMORY [14] ou par la Méthode des Centres de P. HUARD [21].

B) Détermination des inégalités

Considérons le programme PE_n qui se déduit du programme P_n en fixant les variables libres à leur valeur déterminée par l'état :

$$n_j = 1 \qquad \forall n_j \in J_{L1}$$
$$n_j = 0 \qquad \forall n_j \in J_{L2}$$
$$0 \leq n_j \leq 1 \qquad \forall n_j \in J_{L3}$$

La résolution de ce programme peut conduire à des inégalités

- du type introduit par *BENDERS*

(2) $b.n \leq p$

(1) $a.n \leq Z^* + p$

où Z^* est la meilleure valeur connue de la fonction économique ; où les coefficients a, b, p sont obtenus à partir des coûts réduits du tableau final du programme PE_n, l'inégalité (2) étant obtenue dans le cas où le problème est non réalisable, l'inégalité (1) étant obtenue dans le cas d'une solution réalisable.

- du type introduit par *GOMORY-JOHNSON* [15].

Ces inégalités sont donc engendrées au cours de l'algorithme et l'ordre dans lequel elles le sont dépend de l'état correspondant au noeud considéré, d'où l'importance de la détermination de l'état en ce noeud. C'est la liberté de ce choix qui donne toute sa "souplesse" à l'algorithme.

C) Exploitation des inégalités

Il peut y avoir une inégalité ou un ensemble d'inégalités :

- si une inégalité est *irréalisable* elle implique un "pas en arrière".

  ex : $3\eta_1 - 2\eta_2 - 5\eta_3 \leq -12$

- *une inégalité peut permettre de fixer des variables libres ;* on procède de la manière suivante :

  - changer tous les termes négatifs en termes positifs au moyen du changement de variable :

    $\bar{\eta}_j = 1 - \eta_j$

- fixer une variable à zéro si son coefficient excède le membre de droite

  ex : $-5\eta_1 + 3\eta_2 - 4\eta_3 + 2\eta_4 \leq -7$ [17]

  $5\bar{\eta}_1 + 3\eta_2 + 4\bar{\eta}_3 + 2\eta_4 \leq 2$

  $\bar{\eta}_1 = \bar{\eta}_3 = \eta_2 = 0 \Longrightarrow \eta_2 = 0,\ \eta_1 = 1,\ \eta_3 = 1$

- *les inégalités permettent de déterminer l'indice* k* *de la variable* η_{k^*} sur laquelle s'opèrera la séparation. Cette variable η_{k^*} doit être choisie parmi toutes les variables libres ; il apparaît avantageux de limiter l'ensemble de ces variables à un ensemble plus restreint. C.E. LEMKE et K. SPIELBERG [24] ont proposé un moyen pour construire cet ensemble restreint qu'ils ont appelé *"ensemble de variables préférées"*. Ils exploitent les inégalités par *réduction complète*, une par une, et déterminent pour chacune un ensemble de variables préférées $\pi(i)$. On choisit l'ensemble $\pi(i^*)$ dont le nombre d'éléments est minimal sur tous les $\pi(i)$. C'est dans cet ensemble $\pi(i^*)$ que l'on choisit, à l'aide éventuellement d'un autre critère, l'indice k* déterminant la variable η_{k^*} sur laquelle s'opèrera la nouvelle séparation.

Réduire une inégalité par "*réduction complète*" c'est faire une combinaison de cette inégalité avec les contraintes :

$$0 \leqslant \eta_k \leqslant 1$$

Supposons que l'inégalité s'écrive :

$$a_1\eta_1 + \ldots + a_p\eta_p \leqslant -\gamma \qquad \gamma > 0$$

Ce procédé consiste à éliminer les variables η_k dont les coefficients a_k sont négatifs dans l'ordre des a_k décroissants tant que les a_k restent plus grands que le membre de droite.

La détermination de l'ensemble $\pi(i)$ peut se faire en tenant compte ou non de l'état ; il peut même être un moyen de le déterminer.

Exemple :

$$(1) \quad -5\eta_1 + 3\eta_2 - 4\eta_3 + 2\eta_4 + 7\eta_5 \leqslant -1 \quad [17]$$

On rend tous les coefficients négatifs au moyen du changement de variable : $\bar{\eta}_i = 1 - \eta_i$

$$(1) \quad -5\eta_1 - 3\bar{\eta}_2 - 4\eta_3 - 2\bar{\eta}_4 - 7\bar{\eta}_5 \leqslant -13$$

on élimine η_4 $\qquad 0 \leqslant \eta_4 \leqslant 1 \Longrightarrow 0 \geqslant -\eta_4 \leqslant -1$

$$(1) \quad -5\eta_1 - 3\bar{\eta}_2 - 4\eta_3 - 7\bar{\eta}_5 \leqslant -11$$

éliminons $\bar{\eta}_2$

$$(1) \quad -5\eta_1 - 4\eta_3 - 7\bar{\eta}_5 \leqslant -8$$

éliminons $\bar{\eta}_3$

$$(1) \quad -5\eta_1 - 7\bar{\eta}_5 \leqslant -4$$

Pour que l'inégalité (1) soit réalisable il faut que $\eta_1 + \bar{\eta}5 \geq 1$ c'est-à-dire que $\eta_1 = 1$ ou que $\eta_5 = 0$. On en déduit que l'ensemble de variables préférées relatif à l'inégalité (1) est celui constitué des indices 1 et 5.

ALGORITHME

1 - au niveau $\bar{n}$ on connaît : l'état $E_{\bar{n}}$, k_*, Z^*, une solution du problème $PE_{\bar{n}}$ et l'inégalité associée ou un ensemble d'inégalités

2 - déterminer l'état E_n à partir du programme PA_n

3 - résoudre le programme PE_n, en retenir la solution éventuellement, construire et garder les inégalités associées

4 - exploiter les inégalités
si aucun "pas en avant" n'est possible aller en 6
si une solution réalisable est obtenue, réajuster éventuellement Z^* et la retenir, aller en 6

5 - sélectionner k^* de l'ensemble des variables préférées ou à défaut de l'ensemble des variables libres
faire un "pas en avant" sur k^*
remplacer $\bar{n}$ par n, $E_{\bar{n}}$ par E_n, k_* par k^* etc.
revenir en 1

6 - remplacer n par $\bar{n}$, $\bar{n}$ par $\bar{\bar{n}}$; si $\bar{n} = 0$ aller en 7 sinon fixer k^* à la valeur complémentaire de celle qu'il avait au niveau n, repartir en 1

7 - fin.

En résumé l'algorithme nécessite :

- la résolution de programmes auxiliaires PA_n
- la résolution de programmes d'état PE_n
- la construction de conditions du type BENDERS ou du type GOMORY-JOHNSON
- éventuellement la connaissance d'une valeur approchée de Z^* ; en effet celle-ci permet de rendre plus rapidement efficace l'exploitation des inégalités de BENDERS du type 1.

CHAPITRE III

Reprenons le problème auquel nous nous intéressons et dont la formulation P_1 a été donnée au chapitre I, page 7 .

Lorsque toutes les *variables booléennes sont fixées* ce programme se présente sous la forme :

$$
P_f \left\{
\begin{array}{lll}
 & \min \sum_{i \in X_1} \sum_{j \in J(i)} c_{ij} x_{ij} + \sum_{j \in X_2} \sum_{k \in K(j)} c_{jk} x_{jk} & \\
(1) & \underline{P}_i \leq \sum_{j \in J(i)} x_{ij} \leq \bar{P}_i & \forall i \in X_1 : y_i = 1 \\
(2) & \underline{D}_k \leq \sum_{j \in J(k)} x_{jk} \leq \bar{D}_k & \forall k \in X_3 \\
(3) & \sum_{i \in I(j)} x_{ij} - \sum_{k \in K(j)} x_{jk} = 0 & \forall j \in X_2 : y_j = 1 \\
(4) & \underline{Q}_j \leq \sum_{i \in I(j)} x_{ij} \leq \bar{Q}_j & \forall j \in X_2 : y_j = 1 \\
(5) & \begin{cases} 0 \leq x_{ij} \leq s_{ij} \\ 0 \leq x_{jk} \leq s_{jk} \end{cases} & \begin{array}{l} \forall i \in X_1, \forall j \in J(i) \\ \forall k \in X_3, \forall j \in J(k) \end{array}
\end{array}
\right.
$$

Sous cette forme, ce problème n'est autre que celui de la *recherche sur un graphe d'un flot compatible à coût minimal.*

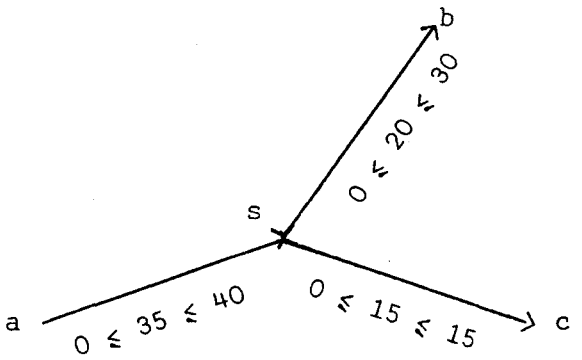
Pour résoudre ce programme nous utilisons l'algorithme *"out-of-kilter"* de FULKERSON.

C'est une méthode primale-duale qui présente l'*avantage* de pouvoir commencer avec une *solution réalisable* ou *non-réalisable* ; ceci est intéressant car l'on peut enchaîner les résolutions successives des différents programmes d'optimisation, chacun commençant avec la solution optimale du précédent, que cette solution soit réalisable ou non.

Cet algorithme présente un *autre avantage* qui lui donne une grande souplesse d'application : si la solution initiale est non seulement irréalisable mais ne présente pas en un sommet du graphe (ou en plusieurs) la propriété

de conservation du flot, l'algorithme donne néanmoins la solution optimale et, au sommet considéré, il y a la même non conservation du flot. On pourra donc *facilement fermer une origine ou un transit* en utilisant cette propriété.

Par exemple supposons que l'on ait extrait d'un graphe ce sous-graphe :



sur chaque arc figurent trois nombres :

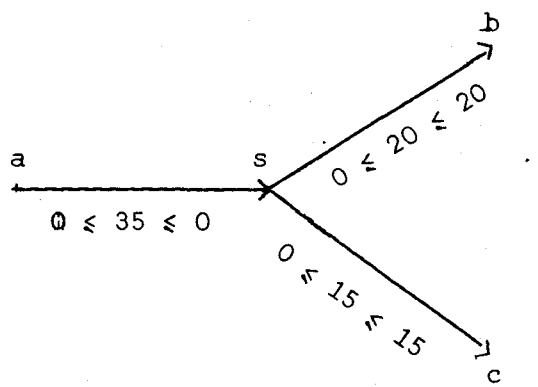
- le premier et le troisième représentent successivement les quantités minimum et maximum (capacité) pouvant circuler sur l'arc.

- celui du milieu désigne le flux circulant sur l'arc.

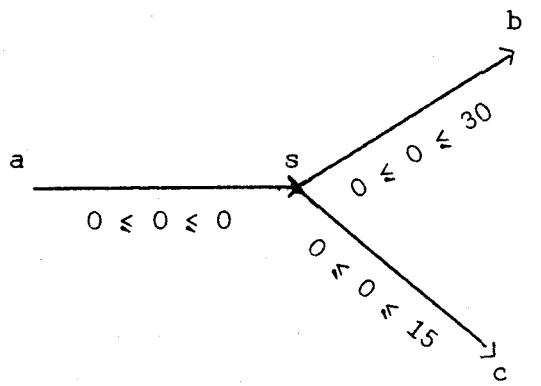
Supposons que nous désirons *fermer le sommet* s ; nous avons alors deux possibilités :

- nous pouvons mettre la capacité de l'arc (a,s) à zéro : la solution initiale n'est plus réalisable.

état initial

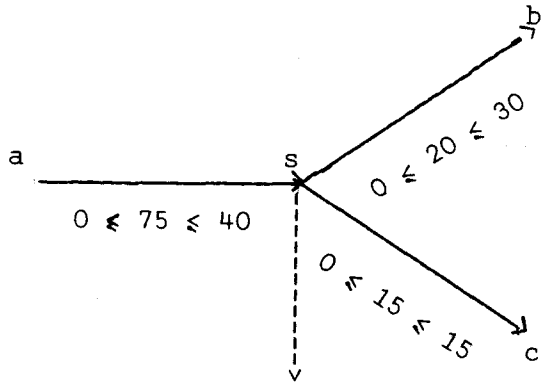


état final



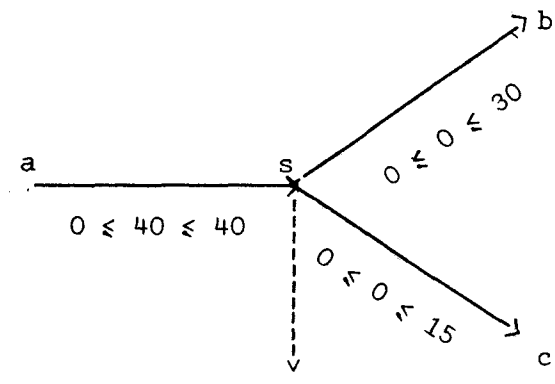
- nous pouvons ajouter au flux de l'arc (a,s) la capacité de l'arc : la solution n'est plus réalisable et il n'y a plus conservation du flot en s

état initial



non conservation : 40

état final



non conservation : 40

Cet algorithme présente un *inconvénient* : son déroulement est *lent* : en effet, en autres choses, il faut effectuer de nombreux tests. Des améliorations ont été proposées [25].

Compte-tenu de notre modèle mathématique nous avons préféré rechercher une *solution initiale* la plus proche possible de la solution optimale. C'est l'algorithme correspondant que nous présentons dans le paragraphe B de ce chapitre ; dans ce paragraphe figurent également les résultats numériques obtenus.

Le paragraphe A est consacré à des rappels sur l'algorithme "out-of-kilter", tandis que le paragraphe C montre un des avantages que présente l'algorithme que nous proposons, dans la résolution de sous-programmes successifs d'un même programme.

A . ALGORITHME "OUT-OF-KILTER"

L'algorithme "out-of-kilter" de FULKERSON [10] permet de résoudre le problème général de flot à coût minimal.

Considérons un *réseau réduit fermé* dont X est l'ensemble des sommets et U l'ensemble des arcs.

A chaque arc (x,y) associons les nombres entiers :

- s(x,y) : capacité de l'arc
- i(x,y) : quantité minimum pouvant circuler sur l'arc
- $\phi(x,y)$: flux
- c(x,y) : coût unitaire de circulation sur l'arc.

Dans ce réseau, dans le sens des arcs, circule un flot.

Cette circulation est soumise à des contraintes :

- de conservation de flot en chaque sommet
- de capacité :

$$i(x,y) \leq \phi(x,y) \leq s(x,y) \qquad \forall (x,y) \in U$$

Le problème posé est le suivant :

"trouver un flot ϕ compatible avec le graphe et tel que le coût de circulation de ce flot soit minimal".

Formulation du problème :

$$F \quad (1) \quad \left[\begin{array}{ll} \min \sum_{(x,y)\in U} c(x,y)\,\phi(x,y) & \\ \sum_{y:(y,x)\in U} \phi(y,x) - \sum_{y:(x,y)\in U} \phi(x,y) = 0 & \forall x \in X \\ i(x,y) \leq \phi(x,y) \leq s(x,y) & \forall (x,y) \in U \end{array} \right.$$

Il se présente sous la forme d'un *programme linéaire en variables bornées*, dont la matrice des contraintes, autres que celles de bornes, a une structure particulière : chaque colonne n'a que deux termes non nuls.

Attachons :

aux contraintes (1) : les variables duales $\pi(x)$
aux contraintes $i(x,y) \leq \phi(x,y)$: les variables duales $w(x,y)$
aux contraintes $\phi(x,y) \leq s(x,y)$: les variables duales $v(x,y)$

Un flot ϕ sera optimal s'il vérifie les *conditions de Kuhn et Tucker*, c'est-à-dire si :

$$\left[\begin{array}{l} \forall x \in X \quad \exists \pi(x) \\ \forall (x,y) \in U \ \exists v(x,y) \geq 0,\ w(x,y) \geq 0 : \\ \pi(x) - \pi(y) + c(x,y) = w(x,y) - v(x,y) \\ w(x,y)\,[\phi(x,y) - i(x,y)] = 0 \\ v(x,y)\,[\phi(x,y) - s(x,y)] = 0 \end{array}\right.$$

Posons :

$$\bar{c}(x,y) = \pi(x) - \pi(y) + c(x,y).$$

Un flot ϕ sera optimal si et seulement si il est possible de trouver, pour tout arc (x,y), des variables $\pi(x)$ et $\pi(y)$ telles que l'une des trois conditions (mutuellement exclusives) soient vérifiées :

$$\left|\begin{array}{l} \bar{c}(x,y) > 0 \text{ et } \phi(x,y) = i(x,y) \\ \bar{c}(x,y) = 0 \text{ et } i(x,y) \leq \phi(x,y) \leq s(x,y) \\ \bar{c}(x,y) < 0 \text{ et } \phi(x,y) = s(x,y) \end{array}\right.$$

Les variables duales $\pi(x)$, $\forall x \in X$, sont appelées *variables nodales*. Les variables $v(x,y)$ et $w(x,y)$ sont appelées les *variables d'arcs*.

Pour un flot ϕ donné et des variables nodales données un arc (x,y) peut se trouver dans l'un des états suivants :

état α	$\bar{c}(x,y) > o$ et $\phi(x,y) = i(x,y)$
état α_1	$\bar{c}(x,y) > o$ et $\phi(x,y) < i(x,y)$
état α_2	$\bar{c}(x,y) > o$ et $(x,y) > i(x,y)$
état β	$\bar{c}(x,y) = o$ et $i(x,y) \leq \phi(x,y) \leq s(x,y)$
état β_1	$\bar{c}(x,y) = o$ et $\phi(x,y) < i(x,y)$
état β_2	$\bar{c}(x,y) = o$ et $\phi(x,y) > s(x,y)$
état γ	$\bar{c}(x,y) < o$ et $\phi(x,y) = s(x,y)$
état γ_1	$\bar{c}(x,y) < o$ et $\phi(x,y) < s(x,y)$
état γ_2	$\bar{c}(x,y) < o$ et $\phi(x,y) > s(x,y)$

Si l'arc (x,y) se trouve dans l'un des états α, β, γ, il est dit *conforme* ou *"in-kilter"*, sinon il est dit *non conforme* ou *"out-of-kilter"*.

A chaque arc (x,y) on associe un *entier* k *positif ou nul* et défini de la manière suivante :

états α ou β ou γ	$k = o$
états α_1 ou β_1	$k = i(x,y) - \phi(x,y)$
états β_2 ou γ_2	$k = \phi(x,y) - s(x,y)$
états α_2	$k = \bar{c}(x,y) \quad [\phi(x,y) - i(x,y)]$
états γ_1	$k = \bar{c}(x,y) \quad [\phi(x,y) - s(x,y)]$

Le problème est le suivant :

on cherche un flot ϕ et des variables nodales π qui rendent tous les arcs conformes. D.R. FULKERSON utilise alternativement deux techniques :

1) on tente à l'aide d'une méthode de marquage de changer de flux de l'arc pour l'amener à être conforme.

2) si l'on ne peut ni augmenter, ni diminuer le flux de cet arc on tente de le rendre conforme en changeant les variables nodales.

Notons que ces deux techniques ont été conçues pour qu'aucun arc ne soit détérioré au cours de l'algorithme, c'est-à-dire que *l'indice de conformité de chaque arc varie d'une manière monotone non croissante.*

PROCEDURE DE MARQUAGE

Supposons qu'un sommet x soit déjà marqué : $[\ell^{+-}, \varepsilon(\ell)]$

Procédure A

Si $(x,y) \in U$, le sommet y pourra être marqué à partir du sommet x et recevra alors la marque $[x^{+}, \varepsilon(x)]$ que s'il se trouve dans l'état :

$$\alpha_1 : \text{alors } \varepsilon(x) = \min [\varepsilon(\ell), i(x,y) - \phi(x,y)]$$

$$\gamma_1 \text{ ou } \beta_1 \text{ ou } \beta(\phi \neq s) : \text{alors } \varepsilon(x) = \min [\varepsilon(\ell), i(y,x) - \phi(y,x)]$$

Procédure B

Si $(y,x) \in U$, le sommet y pourra être marqué à partir du sommet x et recevra alors la marque $[x^{-}, \varepsilon(x)]$ que s'il se trouve dans l'état :

$$\gamma_2 : \text{alors } \varepsilon(x) = \min [\varepsilon(\ell), \phi(y,x) - s(y,x)]$$

$$\alpha_2, \beta_2, \beta(\phi \neq i) : \text{alors } \varepsilon(x) = \min [\varepsilon(\ell), i(y,x) - \phi(y,x)]$$

ALGORITHME

On part d'une solution primale duale quelconque. *On examine les arcs les uns après les autres dans un ordre quelconque.*

PAS 1 Si l'arc (x,y) est conforme passer à l'arc suivant ; si tous les arcs sont conformes le flot est optimal et l'algorithme est terminé. Si l'arc (x,y) est non conforme aller au PAS 2.

PAS 2 Appliquer la procédure de marquage en commençant :

- par le sommet x si l'arc (x,y) est dans l'un des trois états α_2, β_2, γ_2.

- par le sommet y sinon.

Si l'on peut marquer le sommet y à partir du sommet x ou le sommet x à partir du sommet y aller au PAS 3, sinon aller au PAS 4.

PAS 3 On dit qu'il y a *passage*. Sur la chaîne existante modifier le flot de manière suivante :

- si l'arc (x,y) est dans l'un des trois états α_1, β_1, γ_1 ajouter $\varepsilon(x)$ au flux de chaque arc-avant de la chaîne et soustraire $\varepsilon(x)$ au flux de chaque arc-arrière de cette chaîne. Ajouter $\varepsilon(x)$ au flux de l'arc (x,y).

- sinon ajouter $\varepsilon(y)$ au flux de chaque arc-avant de la chaîne et soustraire $\varepsilon(y)$ au flux de chaque arc-arrière. Soustraire $\varepsilon(y)$ au flux de l'arc (x,y).

Si l'arc (x,y) est conforme aller au PAS 1, sinon effacer les marques et aller au PAS 2.

PAS 4 On dit alors qu'il y a *non-passage*. Soit Z l'ensemble des sommets marqués et $\bar{Z}$ celui des sommets non marqués.

Considérer les deux sous-ensembles d'arcs :

$$U_1=\{(x,y) \mid x \in Z,\ y \in \bar{Z} : \bar{c}(x,y) > 0 \text{ et } f(x,y) \leq s(x,y)\}$$

$$U_2=\{(x,y) \mid x \in \bar{Z},\ y \in Z : \bar{c}(x,y) < 0 \text{ et } f(x,y) \geq i(x,y)\}$$

Poser $\delta = \min \left| \min_{U_1} \bar{c}(x,y),\ \min_{U_2} (-\bar{c}(x,y)) \right|$

si $U_1 \cup U_2 \neq \emptyset$, sinon poser $\delta = \infty$

- si $\delta = \infty$ le problème n'admet pas de solution compatible

- sinon changer les variables nodales en ajoutant δ à toutes les variables nodales des sommets non marqués.

Effacer les marques.

Si l'arc (x,y) est conforme aller au PAS 1, sinon aller au PAS 2.

On montre que l'algorithme converge en un nombre fini d'itérations.

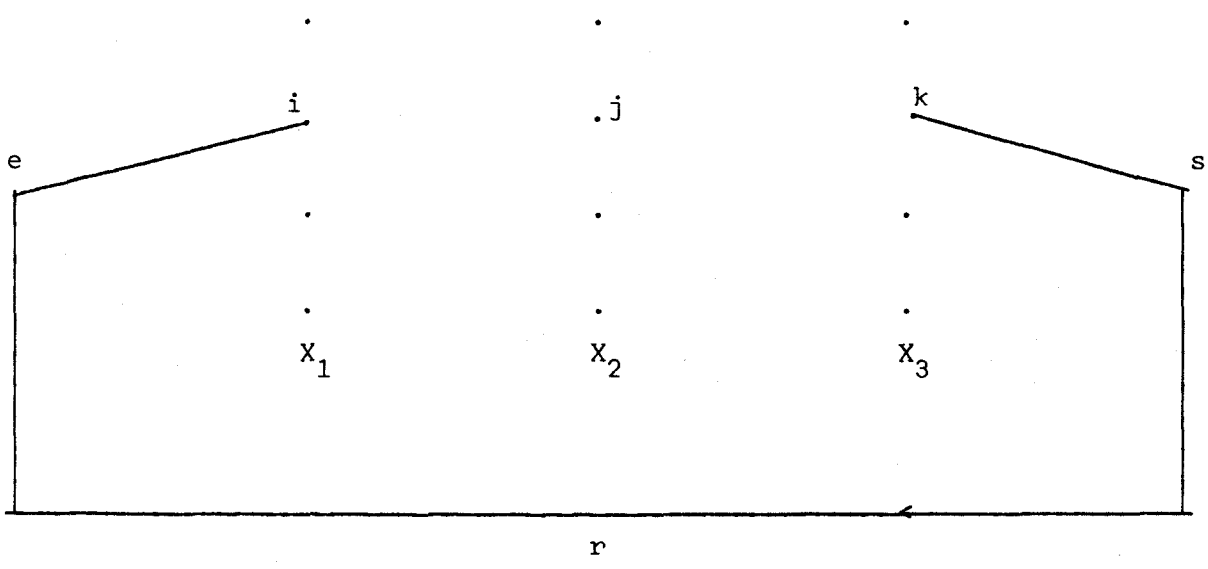
Remarques

- cet algorithme s'applique à un multigraphe.

- les quantités $\bar{c}(x,y) = \pi(x) - \pi(y) + c(x,y)$ sont des *coûts réduits*.
  $\forall (x,y) \in U$: $\bar{c}(x,y)$ traduit la variation de la fonction économique consécutive à l'augmentation ou à la diminution d'une unité du flux passant sur cet arc.

B . RECHERCHE D'UNE SOLUTION INITIALE.

Pour appliquer l'algorithme "out-of-kilter" au problème formulé en tête de ce chapitre, nous devons transformer le graphe G(X,U) de notre modèle mathématique en un *réseau réduit fermé* et tel que toutes les contraintes du programme P_f soient prises en considération.



Contrainte (1) : il suffit de créer les arcs (e,i) $\forall i \in X_1$, e *étant le sommet-entrée* ; on pose :

$$s_{ei} = \bar{P}_i$$
$$i_{ei} = \underline{P}_i$$
$$c_{ei} = 0$$

Contrainte (2) : il suffit de créer les arcs (k,s) $\forall k \in X_3$, s *étant le sommet-sortie* ; on pose :

$$s_{ks} = \bar{D}_k$$
$$i_{ks} = \underline{D}_k$$
$$c_{ks} = 0$$

Contrainte (4) : il faut remplacer chaque transit j par deux sommets j' et j" et l'on pose :

$$s_{j'j"} = \bar{Q}_j$$
$$i_{j'j"} = \underline{Q}_j$$
$$c_{j'j"} = 0$$

On ajoute un arc de retour r = (s,e) et l'on pose :

$$s_{se} = \min \left[\sum_{k \in X_1} \bar{P}_i, \sum_{k \in X_3} \bar{D}_k \right]$$

$$i_{se} = \max \left[\sum_{i \in X_1} \underline{P}_i, \sum_{k \in X_3} \underline{D}_k \right]$$

$$c_{se} = 0.$$

On conserve les contraintes (3) auxquelles on ajoute les contraintes de conservation de flot aux sommets e, s, i ($\forall i \in X_1$) et k ($\forall k \in X_3$).

Le programme P_f peut alors se mettre sous la forme F du paragraphe précédent et l'on peut lui appliquer l'algorithme "out-of-kilter".

Pour simplifier la présentation de *l'algorithme de Recherche d'une Solution Initiale* (R. S. I.) que nous proposons, nous allons un peu simplifier le problème P_f. Cela n'enlève aucune généralité à l'algorithme ; il suffira de quelques modifications de cet algorithme pour le transposer ensuite au problème P_f.

Considérons le programme A déduit du programme P_f.

$$
A \quad \left\{ \begin{array}{ll}
\min \sum_{i \in X_1} \sum_{j \in J(i)} c_{ij} x_{ij} + \sum_{j \in X_2} \sum_{k \in K(j)} c_{jk} x_{jk} & \\
(1') \quad \sum_{j} x_{ij} \leq \bar{P}_i & \forall i \in X_1,\ j \in J(i) \\
(2') \quad \sum_{k} x_{jk} \geq \bar{D}_k & \forall k \in X_3,\ j \in J(k) \\
(3') \quad \sum_{i} x_{ij} - \sum_{k} x_{jk} = 0 & \forall j,\ i \in I(j) \text{ et } k \in K(j) \\
(5') \quad \begin{cases} 0 \leq x_{ij} \leq s_{ij} & \forall i \in X_1,\ \forall j \in J(i) \\ 0 \leq x_{jk} \leq s_{jk} & \forall k \in X_3,\ \forall j \in J(k) \end{cases} &
\end{array} \right.
$$

- les contraintes (4) ont été omises

- les contraintes (1) et (2) simplifiées.

De façon semblable à ce que nous avons fait pour le programme P_f nous transformons le graphe G(X,Y) sur lequel est défini le programme A en un réseau réduit fermé tel que les contraintes de ce programme soient prises en considération.

Nous nous proposons dans ce chapitre :

1) de déterminer un flot initial sur le réseau réduit fermé

2) de déterminer des variables nodales initiales des sommets de ce même réseau.

I - DÉTERMINATION DU FLOT INITIAL

Le principe évident en est le suivant :
tenter de satisfaire au moindre coût les demandes des destinations.

1 - Graphe tri-parti G(X,U)

Dans cette section nous proposons la *détermination d'un flot initial compatible avec le graphe.*

Pour une destinations k quelconque considérons l'ensemble E de tous les chemins existants reliant les différentes origines à cette destination.

A chacun de ces chemins attachons :

- un coût unitaire égal à la somme des coûts unitaires des deux arcs constituant ce chemin

- un nombre ϕ représentant la quantité qu'il est encore possible d'acheminer par ce chemin à l'instant considéré.

Notations

N : nombre d'éléments de E $N = |E|$

T : tableau dans lequel sont rangés les différents chemins (précédemment numérotés) par ordre de coûts croissants

D : tableau dans lequel sont rangées les demandes des destinations

K : nombre d'éléments de X_3

ALGORITHME

PAS 1 faire k = o

PAS 2 faire k = k+1
si k > K aller au PAS 5
construire le tableau T
faire p = o

PAS 3 faire p = p+1
si p > N aller au PAS 2
calculer ϕ correspondant au chemin T(p)

PAS 4 si ϕ = o aller au PAS 3
calculer M = max [ϕ, D(k)]
si M = ϕ prendre M = D(k)
ajouter M au flux de chacun des arcs constituant le chemin retirer M de la disponibilité de l'origine correspondante
faire D(k) = D(k) - M
si D(k) = o aller au PAS 2 sinon aller au PAS 3.

PAS 5 FIN.

VARIANTE DE L'ALGORITHME

Pour une destination k quelconque nous considérons l'ensemble E défini précédemment.

E : ensemble des chemins existants reliant les différentes origines à cette destinations.

Soit

T_k : ensemble des transits t_i par lesquels passent ces différents chemins.

Pour chacun de ces transits nous calculons un "coût unitaire moyen" :

$$CTM(t_i,k) = \text{somme des coûts des chemins passant par ce transit/nombre de ces chemins}$$

et à chaque destination k nous lui associons la quantité :

$$REG(k) = CTM(t_2,k) - CTM(t_1,k)$$

où t_1 est le transit de "coût unitaire moyen" le plus faible et t_2 celui dont le "coût unitaire moyen" est immédiatement supérieur à celui de t_1.

La variante que nous proposons consiste non plus à satisfaire les demandes des destinations dans un ordre de numérotation quelconque mais par ordre décroissant des quantités REG.

Remarques

1) Il est possible d'envisager d'autre critères pour déterminer l'ordre dans lequel les destinations doivent être servies. Personnellement pour ceux que nous avons envisagés c'est celui que nous proposons qui nous a donné, en moyenne, les meilleurs résultats, dans les meilleurs temps.

2) Si une destination n'est servie que par un seul transit, elle doit être servie en priorité.

3) S'il arrive que deux destinations doivent être servies en même temps, on choisit celle pour laquelle la quantité suivante est la plus grande :

$$REG(k) = CTM(t_3,k) - CTM(t_2,k)$$

où t_3 est le transit dont le "coût unitaire moyen" est immédiatement supérieur à celui du transit t_2.
Si une des deux destinations n'est servie que par deux transits, elle doit être servie en priorité.

2 - Réseau réduit fermé

On ajoute au graphe G(X,U) le sommet-entrée e et le sommet-sortie s, l'arc de retour r = (s,e) et les arcs :

$$(e,i) \qquad \forall i \in X_1$$

$$(k,s) \qquad \forall k \in X_3$$

Le flot initial proposé pour le graphe G(X,U) s'étend au réseau de la manière suivante :

$$\forall i \in X_1 : x_{ei} = \sum_{j \in J(i)} x_{ij}$$

$$\forall k \in X_3 : x_{ks} = \sum_{j \in J(k)} x_{jk}$$

$$x_r = x_{se} = \sum_{k \in X_3} \sum_{j \in J(k)} x_{jk}$$

Remarque

Tandis que le flot initial proposé est *compatible avec le graphe*, le flot étendu au réseau peut *ne pas être compatible avec les contraintes du réseau.*

II - DÉTERMINATION DES VARIABLES NODALES

Notre but est évident : mettre le plus possible d'arcs en état conforme ou tout au moins tenter de les rapprocher de cet état en leur donnant *l'indice de conformité le plus faible possible.*

1 - Graphe tri-parti

La détermination des variables nodales se fera de la manière dynamique suivante :

phase 1 prendre les variables nodales des origines nulles

phase 2 déterminer les variables nodales des transits à partir de l'algorithme I, en gardant fixes les variables nodales des origines

phase 3 déterminer les variables nodales des destinations à partir de l'algorithme I, en gardant fixes les variables nodales des transits déterminées en phase 2

phase 4 ajuster les variables nodales des origines à partir de l'algorithme II, en gardant fixes les variables nodales des transits déterminées en phase 2.

Remarque

Le flot initial proposé peut ne pas être compatible avec le réseau ; dans ce cas les arcs non conformes peuvent se trouver dans l'état α_1 ou dans l'état β_1 avec un indice de conformité $k=i-\phi$, où i est la borne inférieure de l'arc et ϕ le flux ; la *détermination des variables nodales initiales est sans influence sur leur indice de conformité.*

Notations

Si E est un sous-ensemble d'arcs du graphe tri-parti on posera :

$$(\alpha) = \{(x,y) \in E : \bar{c}(x,y) > o\}$$

$$(\beta) = \{(x,y) \in E : \bar{c}(x,y) = o\}$$

$$(\gamma) = \{(x,y) \in E : \bar{c}(x,y) < o\}$$

ALGORITHME I

Considérons un transit quelconque ou une destination quelconque y, soit :

E : ensemble des arcs incidents à l'un ou à l'autre de ces sommets

N : nombre d'éléments de E

S : nombre d'arcs saturés

$s : \max_{Es} [\pi(x) + c(x,y)]$

où $Es = \{(x,y) \in E : \phi(x,y) = s(x,y)\}$

$q : \min_{Eq} [\pi(x) + c(x,y)]$

où $Eq = \{(x,y) \in E : o \leq \phi(x,y) < s(x,y)\}$

PAS 1 calculer les quantités précédentes

PAS 2 si S=N aller au PAS 6

PAS 3 si S=0 aller au PAS 5

PAS 4 si $\delta s \leq \delta q$ aller au PAS 5
si $S \leq N/2$ aller au PAS 5 sinon aller au PAS 6

PAS 5 prendre $\pi(y) = s$
aller au PAS 7

PAS 6 prendre $\pi(y) = s$

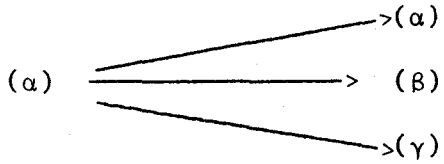
PAS 7 FIN.

La construction de cet algorithme est basée sur la remarque suivante :

Remarque

$$\forall (x,y) \in E \quad \bar{c}(x,y) = \pi(x) + c(x,y) - \pi(y)$$

Déterminons les transitions engendrées par une augmentation de la variable nodale $\pi(y)$, la quantité $m = \pi(x) + c(x,y)$ restant fixe.



avec diminution de l'indice de conformité des arcs en état α_2

(β) ——————> (γ)

(γ) ——————> (γ) avec augmentation de l'indice de conformité des arcs en état α_1

ALGORITHME II

Considérons une origine quelconque x. Soit :

E : ensemble des arcs issus de ce sommet

N : nombre d'éléments de E

C : nombre d'arcs conformes

S : nombre d'arcs saturés

P : nombre d'arcs dans l'état α_2

Q : nombre d'arcs dans l'état γ_1

B : nombre d'éléments du sous-ensemble E_B :

$$E_B : \{(x,y) \in E : \phi(x,y) = o\}$$

C : nombre d'éléments du sous-ensemble E_C :

$$E_C : \{(x,y) \in E : \phi(x,y) = s(x,y)\}$$

D : nombre d'éléments du sous-ensemble E_D :

$$E_D : \{(x,y) \in E : o < \phi(x,y) < s(x,y)\}$$

$$\delta_1 : \min_{(\alpha)} [\bar{c}(x,y)]$$

$$\delta_2 : \min_{(\gamma)} [-\bar{c}(x,y)]$$

$$s : \max_{(\alpha)} [\bar{c}(x,y)]$$

PAS 1 calculer les quantités précédentes

PAS 2 si C=N aller au PAS 9
si D≠o aller au PAS 9
si S=N aller au PAS 8
si P≠o et Q≠o aller au PAS 9
si B≠o et C≠o aller au PAS 9
si B≠o aller au PAS 3
si C≠o aller au PAS 4 sinon aller au PAS 5

PAS 3 si P≠o aller au PAS 9 sinon aller au PAS 6

PAS 4 si Q≠o aller au PAS 9 sinon aller au PAS 7

PAS 5 si P≠o aller au PAS 7

PAS 6 faire $\pi(x) = \pi(x) + \delta_2$ et aller au PAS 9

PAS 7 faire $\pi(x) = \pi(x) - \delta_1$ et aller au PAS 9

PAS 8 faire $\pi(x) = \pi(x) - s$

PAS 9 FIN.

Remarque

Soit la relation :

$$\bar{c}(x,y) = \pi(x) + c(x,y) - \pi(y)$$

Déterminons les transitions engendrées par une variation de la variable nodale $\pi(x)$, la quantité $m = c(x,y) - \pi(y)$ restant inchangée.

1) **Augmentation de** $\pi(x)$

$(\alpha) \longrightarrow (\alpha)$ avec augmentation de l'indice de conformité des arcs dans l'état α_2

$(\beta) \longrightarrow (\alpha)$

$(\gamma) \longrightarrow (\gamma)$, $(\gamma) \longrightarrow (\beta)$, $(\gamma) \longrightarrow (\alpha)$ avec diminution de l'indice de conformité des arcs dans l'état γ_1

2) **Une diminution de** $\pi(x)$

$(\gamma) \longrightarrow (\gamma)$ avec augmentation de l'indice de conformité des arcs dans l'état γ_1

$(\beta) \longrightarrow (\gamma)$

$(\alpha) \longrightarrow (\alpha)$, $(\alpha) \longrightarrow (\beta)$, $(\alpha) \longrightarrow (\gamma)$ avec diminution de l'indice de conformité des arcs dans l'état α_2

2 - Réseau réduit fermé

I - Détermination de la variable du sommet-entrée

On prend $\pi(e) = o$ et on ajuste cette variable nodale à l'aide de l'algorithme II.

II - Détermination de la variable nodale du sommet-sortie

On applique à ce sommet l'algorithme I en laissant fixes les variables nodales des destinations.

Nous appelerons *algorithme R.S.I.* (Recherche d'une Solution Initiale) l'algorithme groupant la recherche d'un flot initial et la recherche des variables nodales initiales.

Remarques

1 - On peut étendre l'algorithme de Recherche d'une Solution Initiale (R.S.I.) au programme P_f donné au début de ce chapitre.

Nous proposons d'appliquer l'algorithme de détermination du flot initial en deux temps :

- *premier objectif* :
  tenter de satisfaire la demande minimum de chaque client

- *deuxième objectif* :
  si le premier objectif a été atteint tenter de satisfaire la demande maximum de chaque client.

Le flot peut ou non être compatible avec le graphe

- s'il n'existe pas de contraintes (4), c'est-à-dire si aucune restriction de capacité n'est imposée sur les transits, déterminer les variables nodales par l'application des algorithmes I et II.

- s'il existe des contraintes (4), procéder comme ci-dessus pour la détermination des variables nodales mais donner la même variable nodale aux sommets dédoublés.

2 - Cas d'un graphe N-parti

Soit $n \in N$, $n \geq 2$ et soit un graphe G(X,U) où X est l'ensemble de ses sommets et U l'ensemble de ses arcs.

Nous dirons que ce graphe est un graphe n-parti si : $X_1, X_2, \ldots X_n$ étant une partition de X, ses arcs sont alors de la forme (x,y) avec

$$x \in X_i$$
$$\text{pour } i = 1 \text{ ou } 2 \ldots \text{ ou } (n-1)$$
$$y \in X_{i+1}$$

Les modifications à apporter à la mise en place de l'algorithme R.S.I sont évidentes. Nous n'avons pas traité d'exemples numériques.

C . RÉSULTATS NUMÉRIQUES

Les calculs ont été exécutés sur un ordinateur 10070 de la compagnie (C.I.I) sous système SIRIS 7, au Centre Interuniversitaire du Traitement de l'Information.

Nous avons mis au point une *procédure de construction du graphe* constituant notre modèle mathématique.

Procédure de construction

On utilise une procédure P [8] permettant d'engendrer une suite de nombres pseudo-aléatoires suivant la loi uniforme, ces nombres étant compris entre deux nombres donnés à priori.

Sont ainsi déterminés :

- le nombre d'arcs incidents à chaque transit et les numéros des origines correspondantes

- le nombre d'arcs incidents à chaque destination et les numéros des transits correspondants

- la production minimum et la production maximum de chaque origine

- la demande minimum et la demande maximum de chaque destination

- le coût de chaque arcs et la borne supérieure de son flux

- la quantité minimum et la quantité maximum pouvant passer par chaque transit.

Pour *utiliser* la procédure de construction nous devons donner à priori :

- le nombre des origines, celui des transits et celui des destinations

- les différents nombres en vue de l'utilisation de la procédure P.

Une procédure d'ajustement est ajoutée au programme de construction. En effet le graphe construit peut présenter certaines particularités :

- existence d'un ou plusieurs sommets "pendants"

- existence de deux arcs reliant deux sommets.

D'autre part il est nécessaire, par exemple, entre autres vérifications, de s'assurer que toute la production fournie peut être écoulée à travers le graphe.

A la fin du programme de construction est ajoutée une procédure de détermination du *réseau réduit fermé correspondant au graphe construit.*

Nous avons construit, pour chaque ligne de la table I, cinq graphes : nous avons retenu la *valeur minimale* et la *valeur maximale* des cinq gains correspondants et ce sont ces résultats qui figurent dans la quatrième colonne de la table I.

Nous notons :

$$\text{Gain en \% : } (t_1 - t_2) / t_1 \text{ où}$$

t_1 est le temps de résolution de l'algorithme "out-of-kilter" à partir d'une solution initiale nulle

t_2 est celui de la résolution du même algorithme à partir d'une solution initiale déterminée par l'algorithme R.S.I, compte-tenu du temps de recherche de cette solution initiale.

Sur les exemples traités nous avons constaté que l'efficacité de l'algorithme R.S.I dépendait de la *densité* et de la *structure du graphe*.

En effet ceci s'explique par la remarque suivante. Il est facile de vérifier que pendant le déroulement de l'algorithme "out-of-kilter" un *non-passage* prend plus de temps qu'un *passage*. Il est donc évident que la solution initiale sera d'autant meilleure qu'elle mettra le plus possible d'arcs en état conforme et qu'elle donnera aux autres un indice de conformité faible. Par conséquent, *pour un sommet donné*

- plus le nombre d'arcs incidents intérieurement à ce sommet est faible, meilleure sera la détermination de sa variable nodale

- pour un nombre donné d'arcs incidents la détermination de sa variable nodale sera meilleure si les bornes supérieures des arcs sont "équilibrées".

Par *"bornes supérieures équilibrées"* nous entendons que ces bornes sont de l'ordre de grandeur du flux susceptible de passer réellement sur chacun des arcs.

SOMMETS			GAIN
Nombre d'origines	Nombre de transits	Nombre de destinations	en %
3	3	3	57 à 60
5	5	5	65 à 80
5	3	12	58 à 60
3	6	12	58 à 61
8	8	8	55 à 62
2	8	30	70 à 80
12	12	12	35 à 50
15	15	15	33 à 52
4	15	30	50 à 72
10	30	20	40 à 80
4	25	150	60 à 70
5	50	500	50 à 65
10	50	500	50 à 65
25	50	400	40 à 70
30	70	600	50 à 80

Table I.

D . RÉSOLUTION DE PROGRAMMES SUCCESSIFS

Dans l'application d'une procédure par séparation et évaluation on est amené à résoudre successivement des sous-problèmes d'optimisation ne différant entre eux que par l'existence ou la non-existence d'un ou de plusieurs sommets.

Pour la simplification de l'exposé supposons que le *second programme ne diffère du premier que par la non-existence du transit* j.

On ferme ce transit en utilisant l'une ou l'autre des possibilités évoquées au début de ce chapitre, et l'on peut commencer la résolution du second programme en prenant la solution optimale du premier.

Nous proposons d'*appliquer partiellement l'algorithme R.S.I.*

1 - Détermination du flot initial

- prendre le flot optimal du problème précédent

- annuler les flux des arcs (i,j) $\forall i \in I(j)$
  annuler les flux des arcs (j,k) $\forall k \in K(j)$

- réajuster le flot dans le graphe

- appliquer aux destinations k_j alimentées partiellement ou totalement par le transit j l'algorithme de la recherche du flot initial.

2 - Détermination des variables nodales initiales

- garder les variables optimales du problème précédent sauf celles du sommet j et celles des sommets k_j que l'on recalcule par l'algorithme I.

Pour tester la proposition faite nous avons repris quelques problèmes qui figurent sur la table I.

Pour chacun de ces problèmes nous avons construit des sous-problèmes de la manière suivante :

SP_1 : problème P dans lequel seul le transit 1 est fermé

SP_2 : problème P dans lequel seul le transit 2 est fermé

etc.

Nous avons résolu les différents sous-problèmes par l'algorithme "out-of-kilter" :

Procédure A

solution initiale : solution optimale du sous-problème précédent ; pour le premier sous-problème : solution nulle

Procédure B

solution initiale : obtenue par l'application partielle de l'algorithme R.S.I. ; pour le premier sous-problème : algorithme R.S.I.

Nous notons : gain en % = $\frac{t_1 - t_2}{t_1}$ où

t_1 : temps de résolution des sous-programmes à l'aide de la procédure A

t_2 : temps de résolution des sous-programmes à l'aide de la procédure B.

Pour tous les exemples traités :

- gains positifs de l'ordre de 30 à 60 %
- pour la plupart d'entre eux les gains se situent aux environs de 48 %.

Remarques

Pour la résolution du second problème et des suivants on peut gagner du temps-calcul en ne faisant tester, par la procédure "out-of-kilter", que les arcs qui deviennent non conformes par la fermeture du transit j ; cela découle de la propriété de cet algorithme de ne pas détériorer la conformité des arcs en cours d'éxécution.

CHAPITRE IV

A . ETABLISSEMENT ET RÉSOLUTION DES PROGRAMMES AUXILIAIRES PA_n

Rappels des notations

$G(X,U)$: graphe sur lequel est défini le programme initial

i désigne une origine
j désigne un transit
k désigne une destination
ℓ désigne un indice de variable booléenne, soit i, soit j.

$I(j) = \{i \in X_1 : (i,j) \in U\}$
$K(j) = \{k \in X_3 : (j,k) \in U\}$
$J(i) = \{j \in X_2 : (i,j) \in U\}$
$J(k) = \{j \in X_2 : (j,k) \in U\}$

variables fixées : $J_1 = \{\ell : y_\ell = 1\}$
$J_0 = \{\ell : y_\ell = 0\}$

variables libres : $J_L = \{\ell : y_\ell \in \{0,1\}\}$

Reprenons le programme P formulé au premier chapitre, page 7.

$$
P \quad \left[\begin{array}{ll}
\min \sum_{i \in X_1} (f_i + \sum_{j \in J(i)} c_{ij} x_{ij}) + \sum_{j \in X_2} (f_j + \sum_{k \in K(j)} c_{jk} x_{jk}) & \\
y_i \underline{P}_i \leq \sum_{j \in J(i)} x_{ij} \leq y_i \bar{P}_i & \forall i \in X_1 \\
\underline{D}_k \leq \sum_{j \in J(k)} x_{jk} \leq \bar{D}_k & \forall k \in X_3 \\
\sum_{i \in I(j)} x_{ij} - \sum_{k \in K(j)} x_{jk} = 0 & \forall j \in X_2 \\
y_j \underline{Q}_j \leq \sum_{i \in I(j)} x_{ij} \leq y_j \bar{Q}_j & \forall j \in X_2 \\
\left\{ \begin{array}{ll} 0 \leq x_{ij} \leq s_{ij} & \forall i \in X_1, \forall j \in J(i) \\ 0 \leq x_{ik} \leq s_{jk} & \forall k \in X_3, \forall j \in J(k) \end{array} \right. & \\
y_\ell \in \{0,1\} & \forall \ell = i \in X_1 \text{ et } \forall \ell = j \in X_2
\end{array} \right.
$$

En un noeud γ de niveau n le programme P_n est le programme P dans lequel les variables fixées ont été remplacées par leurs valeurs.

Nous déterminons un programme auxiliaire PA_n *en vue :*

- d'obtenir un minorant de la fonction économique du programme P_n au noeud γ.

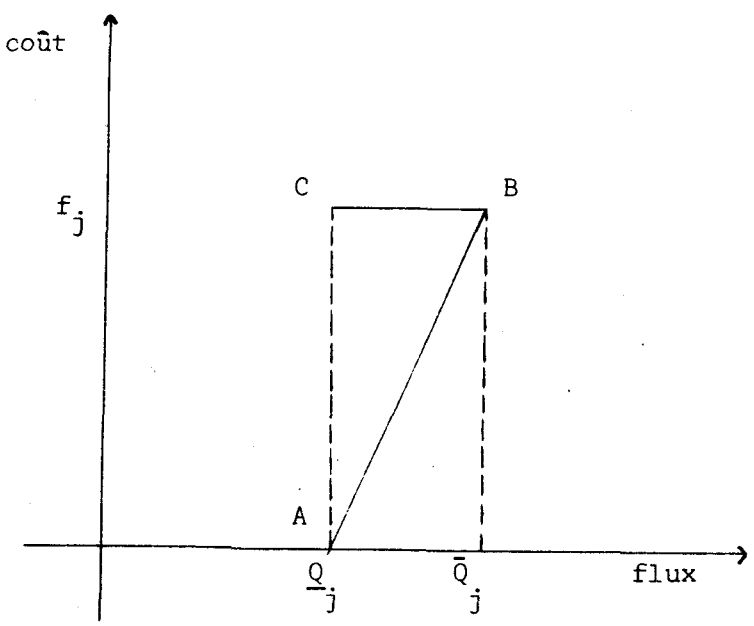
- de déterminer éventuellement l'*état* au noeud γ.

Remarque

Généralement l'algorithme de résolution des sous-programmes d'optimisation est celui du simplex ; le programme PA_n s'obtient par *relaxation* du programme P_n, relaxation portant sur toutes les variables libres : le même algorithme de résolution est alors utilisé pour résoudre le programme PA_n.

Compte-tenu de notre modèle mathématique qui confère une structure particulière à la matrice des contraintes nous allons définir différemment le programme PA_n.

Considérons, par exemple, *un transit* j dont la variable boolèenne correspondante est libre au noeud γ ; soit f_j le coût fixe attaché à l'existence de ce sommet.



Déterminons un coût unitaire de *passage* en remplaçant le segment de droite BC par le segment de droite BA ; soit q_j ce coût unitaire ;

$$q_j = f_j \mid (\bar{Q}_j - \underline{Q}_j)$$

- si toute la quantité $\bar{Q}_j$ passe par le transit le coût de passage est f_j

- si aucun flux ne transite le coût correspondant est nul

- sinon le coût est proportionnel à la quantité qui transite

On procéde de même pour une origine dont la variable booléenne est libre ; on détermine un coût unitaire de *passage* :

$$q_i = f_i \mid (\bar{P}_i - \underline{P}_i)$$

Remarque

- si aucune restriction de capacité n'existe sur le transit j, on calcule alors la quantité maximum qui peut passer :

$$Q_j = \min \left(\sum_{i \in I(j)} s_{ij}, \sum_{k \in K(j)} s_{jk} \right)$$

Pour *construire* le programme PA_n on procède de la manière suivante :

- on calcule un coût unitaire de *passage* pour chaque origine dont la variable booléenne est libre

- on calcule un coût unitaire de *passage* pour chaque transit dont la variable booléenne est libre.

Remarque

Pour ne pas alourdir l'écriture du programme PA_n les sommes sont prises telles que les arcs correspondants existent.

Le programme PA_n s'écrit :

$$PA_n \quad \left[\begin{array}{l}
\min \sum\sum_{ji} c_{ij}x_{ij} + \sum\sum_{jk} c_{jk}x_{jk} + \sum\sum_{ij} \frac{f_i}{\bar{P}_i - \underline{P}_i} x_{ij} \\
\quad + \sum\sum_{jk} \frac{f_j}{\bar{Q}_j - \underline{Q}_j} x_{jk} + \sum_{i \in J_1} f_i + \sum_{j \in J_1} f_j \\
\underline{P}_i \leq \sum_j x_{ij} \leq \bar{P}_i \qquad \forall i \in X_1 \cup J_1 \cup J_L \\
\underline{D}_k \leq \sum_j x_{jk} \leq \bar{D}_k \qquad \forall k \in X_3 \\
\sum_i x_{ij} - \sum_k x_{jk} = 0 \qquad \forall j \in X_2 \cup J_1 \cup J_L \\
\underline{Q}_j \leq \sum_i x_{ij} \leq \bar{Q}_j \qquad \forall j \in X_2 \cup J_1 \cup J_L \\
\left\{\begin{array}{l} 0 \leq x_{ij} \leq s_{ij} \\ 0 \leq x_{jk} \leq s_{jk} \end{array}\right.
\end{array}\right.$$

Propriété

La valeur optimale $\underline{z}$ *de la fonction économique du programme* PA_n *est un minorant de la fonction économique du programme* P_n.

Démonstration :

- faisons l'hypothèse que seuls les transits ont des variables booléennes libres ; cela n'enlève aucune généralité à la démonstration et en facilite la présentation.

- posons $Q_j = \underline{Q}_j - \bar{Q}_j$

- allégeons l'écriture des programmes P_n et PA_n

$$P_n \quad \begin{cases} \min\ cx + \sum_{j \in J_L} f_j y_j \\ \text{contraintes sur } x \\ y_j \in \{0,1\} \quad \forall j \in J_L \end{cases}$$

$$PA_n \quad \begin{cases} \min\ cx + \sum\sum_{jk} (f_j/Q_j)\, x_{jk} \\ \\ \text{contraintes sur } x \end{cases}$$

De façon équivalente ce dernier peut s'écrire

$$PA_n \quad \begin{cases} \min\ cx + \sum_{j} f_j y_j \\ \text{contraintes sur } x \\ y_j = \left(\sum_k x_{jk}\right) / Q_j \qquad \forall j \in J_L \end{cases}$$

Notons :

- $(\bar{x},\bar{y})$ la solution optimale du programme P_n
- $(\hat{x},\hat{y})$ celle du programme PA_n
- y le vecteur de composantes : $y_j = (\sum_k \bar{x}_{jk})/Q_j \qquad \forall j \in J_L$
- $z(x,y)$ la valeur de la fonction économique du programme P_n
- $\underline{z}(x,y)$ celle de la fonction économique du programme PA_n

$$\left.\begin{array}{l} - \ 0 \leq y_j \leq 1 \Longrightarrow z(\bar{x},y) \leq z(\bar{x},\bar{y}) \\ - \ (\bar{x},y) \text{ solution réalisable de } PA_n : \\ \qquad \Longrightarrow z(\hat{x},\hat{y}) \leq z(\bar{x},y) \\ - \qquad \underline{z}(\bar{x},y) = z(\bar{x},y) \end{array}\right\} \Longrightarrow \underline{z}(\hat{x},\hat{y}) \leq z(\bar{x},\bar{y})$$

$\underline{z}$ est donc un minorant de la fonction économique du programme P_n.

Résolution du programme PA_n

Nous résolvons ce programme à l'aide de l'algorithme *"out-of-kilter"*.

On commence cet algorithme avec une *solution initiale obtenue par l'algorithme R.S.I.*

Nous montrerons, à la fin de ce paragraphe, que la résolution du programme PA_n permet de déterminer un *état* au noeud γ.

Problème auxiliaire renforcé

Pour améliorer les résultats fournis par la solution optimale du programme PA_n :

- minorant $\underline{z}$

- détermination d'un *état*

nous proposons un *renforcement* du coût optimal $\underline{z}$.

Le programme PA_n a été construit en répartissant le coût fixe de chaque sommet *uniformément* sur les arcs issus de ce sommet.

Problème

"Est-il possible de trouver une autre répartition de ce coût fixe, telle que, pour le programme PAR_n correspondant :

- la solution optimale $\hat{x}$ du programme PA_n soit aussi solution optimale de ce programme

- la valeur optimale de sa fonction économique soit un meilleur minorant $\underline{z}$ de celle du programme P_n ?"

Nous allons mettre en évidence un tel programme PAR_n.

Pour simplifier nous choisissons de déterminer ce programme seulement dans le cas *d'implantations de transits* ; la manière de procéder s'étend facilement au cas d'implantations simultanées d'origines et de transits.

Considérons un transit j.

Rappelons que : $\bar{c}_{jk} = \pi(j) + c'_{jk} - \pi(k)$ est le *coût réduit* de l'arc (j,k)

et $c'_{jk} = c_{jk} + q_j$ où $q_j = \dfrac{f_j}{\bar{Q}_j - \underline{Q}_j}$.

Le programme PA_n a été résolu à l'aide de l'algorithme "out-of-kilter" ; à la sortie de cette procédure les arc (j,k) peuvent se trouver :

- dans l'état γ : $\bar{c}_{jk} < 0$ et $x_{jk} = s_{jk}$

- dans l'état β : $\bar{c}_{jk} = 0$ et $0 \leq x_{jk} \leq s_{jk}$

- dans l'état α : $c_{jk} > 0$ et $x_{jk} = 0$

Appelons $\alpha' = (\alpha'_{jk})$ la *répartition cherchée* et notons $\hat{x} = (\hat{x}_{ij}, \hat{x}_{jk})$ la solution optimale du programme PA_n.

Soit

- $K_j' = \{k \in K : \hat{x}_{jk} = s_{jk} \text{ et } \bar{c}_{jk} < 0\}$
- $N = \sum_{(j,k)} s_{jk} q_j \qquad \forall (j,k) : 0 \leq \hat{x}_{jk} \leq s_{jk} \text{ et } \bar{c}_{jk} = 0$
- $M = \sum_{(j,k)} s_{jk}(q_j - \bar{c}) \quad \forall (j,k) : \hat{x}_{jk} = 0 \text{ et } \bar{c} < q_j$
- $R = \sum_{(j,k)} s_{jk} q_j \qquad \forall (j,k) : \hat{x}_{jk} = s_{jk} \text{ et } \bar{c}_{jk} < 0$
- $S = N + M + R$

Remarque

Un *renforcement* n'est envisageable que si $S < f_j$.

La répartition α' *est la suivante :*

sur les arcs en état (β) : $\alpha'_{jk} = q_j$

$$(\alpha) : \begin{cases} \alpha'_{jk} = 0 \text{ si } \bar{c} \geq q_j \\ \alpha'_{jk} = q_j - \bar{c} \text{ sinon} \end{cases}$$

$$(\gamma) : \quad \alpha'_{jk} = \hat{\alpha}_{jk} + q_j$$

où les $(\hat{\alpha}_{jk})$ sont les composantes du vecteur $\hat{\alpha}$ solution optimale du programme :

$$\begin{cases} \max \sum_k \alpha_{jk} \hat{x}_{jk} & \\ (1) \quad \sum_k \alpha_{jk} \hat{x}_{jk} \leq f_j - S & \forall k \in K_j' \\ (2) \quad \alpha_{jk} \leq \pi(k) - \pi(j) - c_{jk} & \forall k \in K_j' \\ \alpha_{jk} \geq 0 & \forall k \in K_j' \end{cases}$$

la solution optimale est évidente.

Rangeons les destinations dont les indices appartiennent à K'_j par ordre décroissant de leurs demandes et soit K''_j le nouvel ensemble ordonné des indices k :

$$K''_j = \{k_1, k_2, \ldots, k_p, \ldots\}$$

on prend $\alpha_{jk_1} = \pi(k_1) - \pi(j) - c_{jk_1}$

.
.
.
etc.

Cette affectation se poursuivra :

- jusqu'à ce que l'on arrive à saturer la contrainte (1) ; s'il reste des α_{jk_i} non affectés on leur donne la valeur 0
- ou jusqu'à épuisement de l'affectation des α_{jk_i}.

Remarque

Le groupe de contraintes (2) indique que la conformité des arcs est assurée.

Le programme PAR_n s'écrit

$$PAR_n \left[\begin{array}{l} \min \sum\sum_{ij} c_{ij}x_{ij} + \sum\sum_{jk} c_{jk}x_{jk} + \sum\sum_{jk} \alpha'_{jk}x_{jk} \\ \sum_k \alpha'_{jk}x_{jk} \leq f_j \qquad \forall j \in J_L \\ \text{contraintes sur } x \text{ (les mêmes que celles de } PA_n) \end{array} \right.$$

La solution optimale de ce programme est évidemment $\hat{x} = (\hat{x}_{ij}, \hat{x}_{jk})$ et son coût optimal est $\underline{\underline{z}} \geq \underline{z}$.

Propriété

$\underline{z}$ est un minorant de la fonction économique du programme P_n.

Démonstration

* Montrons d'abord la propriété P1 suivante :

$$\sum_{(j,k)} \alpha'_{jk} \bar{x}_{jk} \leq f_j \qquad \forall j \in J_L$$

où $(\bar{x},\bar{y})$ est la solution optimale du programme P_n

Un arc (j,k), sur lequel passe le flux $\bar{x}_{jk}$, était à la fin de la résolution du programme PA_n soit :

dans l'état (β) : alors $\alpha'_{jk}\bar{x}_{jk} \leq q_j s_{jk}$

dans l'état (α) : alors $\begin{cases} \alpha'_{jk}\bar{x}_{jk} = 0 \text{ si } \bar{c} > q_j \\ \alpha'_{jk}\bar{x}_{jk} \leq s_{jk}(q_j - \bar{c}) \text{ sinon} \end{cases}$

dans l'état (γ) : alors $\alpha'_{jk}\bar{x}_{jk} \leq \alpha'_{jk}\hat{x}_{jk}$

en considérant tous les arcs (j,k) convenables :

$$\sum_{(j,k)} \alpha'_{jk}\bar{x}_{jk} \leq S + \sum_{k \in K'_j} \hat{\alpha}_{jk}\hat{x}_{jk} \leq f_j$$

* Le programme PAR_n peut se formuler de façon équivalente :

$$\begin{cases} \min \sum\sum_{ij} c_{ij}x_{ij} + \sum\sum_{jk} c_{jk}x_{jk} + \sum_{j \in J_L} y_j f_j \\ \quad \text{contraintes sur } x \\ \sum_{(j,k)} \alpha'_{jk}x_{jk} \leq f_j \qquad \forall j \in J_L \\ y_j = \dfrac{\sum \alpha'_{jk}x_{jk}}{f_j} \end{cases}$$

Notons

- $(\bar{x},\bar{y})$ la solution optimale du programme P_n
- $(\hat{x},\hat{y})$ celle du programme PA_n
- y le vecteur de composantes $y_j = \left(\sum_{(j,k)} \alpha'_{jk}\bar{x}_{jk}\right)\Big|f_j \quad \forall j \in J_L$
- $z(x,y)$ la valeur de la fonction économique du programme P_n
- $\underline{z}(x,y)$ celle du programme PAR_n

$$\left.\begin{array}{l} \text{- propriété P1} \Longrightarrow 0 \leq y_j \leq 1 \Longrightarrow z(\bar{x},y) \leq z(\bar{x},\bar{y}) \\ \text{- } (x,y) \text{ solution réalisable de } PAR_n \\ \qquad\qquad \Longrightarrow \underline{z}(\hat{x},\hat{y}) \leq \underline{z}(\bar{x},y) \\ \text{- } \underline{z}(\bar{x},y) = z(\bar{x},y) \end{array}\right\} \Longrightarrow \underline{z}(\hat{x},\hat{y}) \leq z(\bar{x},\bar{y})$$

Donc $\underline{z}$ est un minorant de la fonction économique du programme P_n.

Remarque 1

Il est facile d'étendre ce qui vient d'être dit au cas d'implantations simultanées d'origines et de transits.

Remarque 2

Le renforcement effectué en un sommet ne nuit nullement aux renforcements des autres sommets.

Détermination de l'état

La détermination d'un état est une particularité de la "State Enumeration Method" et de cette détermination dépend l'efficacité de l'algorithme.

Plusieurs quantités peuvent être examinées pour cette détermination.

* *Des quantités qui ne dépendent de la résolution d'aucun sous-problème* précédent mais uniquement des données du problème posé :

$$d_i = f_i \,/\, (\bar{P}_i - \underline{P}_i) \text{ et } d_j = f_j \,/\, (\bar{Q}_j - \underline{Q}_j)$$

*Des quantités qui dépendent de la résolution du programme auxiliaire PA_n

1) $a_i = (\sum_j q_i \hat{x}_{ij}) / f_i$ et $a_j = (\sum_k q_j \hat{x}_{jk}) / f_j$

2) $b_i = (\sum_j \hat{x}_{ij}) / (\bar{P}_i - \underline{P}_i)$ et $b_j = (\sum_k \hat{x}_{jk}) / (\bar{Q}_j - \underline{Q}_j)$

3) $c_i = f_i / \sum_j \hat{x}_{ij}$ et $c_j = f_j / \sum_k \hat{x}_{jk}$

Le *renforcement* du problème PA_n permet d'améliorer les quantités a_i et a_j :

$$a'_i = (\sum_j \alpha'_{ij} \hat{x}_{ij}) / f_i \quad \text{et} \quad a'_j = (\sum_k \alpha'_{jk} \hat{x}_{jk}) / f_j$$

où $\alpha' = (\alpha'_{ij}, \alpha'_{jk})$ est la répartition donnant le renforcement.

Ce sont ces dernières quantités que nous avons retenues ; nous montrerons au paragraphe A du chapitre suivant, comment, à l'aide d'une procédure d'arrondi, nous déterminons effectivement un état.

B .-ETABLISSEMENT ET RÉSOLUTION DU PROGRAMME D'ÉTAT PE_n

-DÉTERMINATION DES INÉGALITES : CONTRAINTES DE BENDERS

-RENFORCEMENT DES CONTRAINTES DE BENDERS

I - PROGRAMME D'ÉTAT PE_n

Rappels de notations

$$\left| \begin{array}{l} J_1 = \{\ell : y_\ell = 1\} \\ J_0 = \{\ell : y_\ell = 0\} \\ J_L = \{\ell : y_\ell \in \{0,1\}\} \end{array} \right.$$

La détermination de l'état au noeud γ de niveau n permet de donner la partition des variables libres :

$$\left| \begin{array}{l} J_{L1} = \{\ell : y_\ell = 1\} \\ J_{L2} = \{\ell : y_\ell = 0\} \\ J_{L3} = \{\ell : y_\ell \in \{0,1\}\} \end{array} \right.$$

Le *programme d'état* correspondant au problème P formulé au chapitre I, page 7 s'écrit :

$$PE_n \begin{cases} \min cx + \sum_{i \in (J_1 \cup J_{L1})} f_i + \sum_{j \in (J_1 \cup J_{L1})} f_j & \\ \underline{P}_i \leq \sum_{j \in J(i)} x_{ij} \leq \bar{P}_i & \forall i \in X_1 \cap J_1 \cap J_{L1} \\ \underline{D}_k \leq \sum_{j \in J(k)} x_{jk} \leq \bar{D}_k & \forall k \in X_3 \\ \sum_{i \in I(j)} x_{ij} - \sum_{k \in K(j)} x_{jk} = 0 & \forall j \in X_2 \cap J_1 \cap J_{L1} \\ \underline{Q}_j \leq \sum_{i \in I(j)} x_{ij} \leq \bar{Q}_j & \forall j \in X_2 \cap J_1 \cap J_{L1} \\ 0 \leq x_{ij} \leq s_{ij} & \forall i \in X_1 \cap J_1 \cap J_{L1},\ j \in J(i) \\ 0 \leq x_{jk} \leq s_{jk} & \forall j \in X_2 \cap J_1 \cap J_{L1},\ k \in K(j) \end{cases}$$

Compte-tenu de notre modèle mathématique nous avons choisi de garder l'ensemble J_{L3} vide.

Si l'on voulait s'affranchir de cette hypothèse il suffirait de répartir les coûts fixes des sommets dont les indices appartiennent à J_{L3}, de la manière dont on a procédé au paragraphe précédent ; ensuite on appliquerait l'algorithme "out-of-kilter".

On peut alors appliquer directement l'algorithme "out-of-kilter" au programme PE_n.

Remarque

Il est souhaitable d'éviter que le programme PE_n n'ait pas de solution réalisable. Pour cela il suffit de compléter le graphe G(X,U) de la manière suivante :

- $\forall i \in X_1$, $\forall j \in X_2$ si $(i,j) \notin U$ alors on le crée et l'on pose

$$s_{ij} = +\infty$$
$$c_{ij} = +\infty$$
$$i_{ij} = 0$$

- $\forall j \in X_2$, $\forall k \in X_3$ si $(j,k) \notin U$ alors on le crée et l'on pose

$$s_{jk} = +\infty$$
$$c_{jk} = +\infty$$
$$i_{jk} = 0$$

II - CONTRAINTES DE BENDERS

La détermination des contraintes de BENDERS nécessite la connaissance des variables duales optimales du programme PE_n.

a) Détermination des variables duales optimales du programme PE_n

Les variables duales optimales données par l'algorithme "out-of-kilter" ne sont pas celles du programme PE_n, mais celles correspondantes au programme défini sur le réseau réduit fermé correspondant.

Nous allons supposer que, dans le programme P initial :

- $\underline{P}_i = 0 \quad \forall i$ et nous poserons $\bar{P}_i = P_i$
- $\underline{D}_k = 0 \quad \forall k$ et nous poserons $\bar{D}_k = D_k$
- $\underline{Q}_j = 0$, $\bar{Q}_j = +\infty$ et nous poserons $Q_j = \min \left(\sum_{i \in I(j)} s_{ij}, \sum_{k \in K(j)} s_{jk} \right)$

Cela dans le but de diminuer le nombre de contraintes, donc le nombre de variables duales à déterminer : la présentation en sera plus claire.

Cette restriction n'enlève aucune généralité à cette détermination ; il sera facile d'adapter ce que nous proposons au cas du problème P pris sous sa forme la plus générale.

Nous formulons le problème auquel nous nous intéressons ainsi :

$$
P \quad \left[\begin{array}{ll}
\min cx + \sum_{i\in I} f_i y_i' + \sum_{j\in J} f_j y_j' & \\
\sum_{j\in J(i)} x_{ij} \leq y_i' P_i & \forall i \in X_1 \\
-\sum_{j\in J(k)} x_{jk} \leq -D_k & \forall k \in X_3 \\
\sum_{i\in I(j)} x_{ij} - \sum_{k\in K(j)} x_{jk} = 0 & \forall j \in X_2 \\
\left\{\begin{array}{l} 0 \leq x_{ij} \leq s_{ij}\, y_j' \\ 0 \leq x_{jk} \leq s_{jk} \end{array}\right. & \\
y_i', y_j' \in \{0,1\} &
\end{array}\right.
$$

Remarque

Nous avons "primé" les variables booléennes dans un but de simplification ultérieure.

Le programme P_n correspondant se formule

$$
P_n \quad \left[\begin{array}{ll}
\min cx + \sum_{i\in J_1} f_i + \sum_{j\in J_1} f_j + \sum_{i\in J_L} f_i y_i' + \sum_{j\in J_L} f_j y_j' & \\
\sum_j x_{ij} \leq P_i & \forall i \in X_1 \cap J_1 \\
\sum_j x_{ij} \leq y_i' P_i & \forall i \in X_1 \cap J_L \\
-\sum_j x_{jk} \leq -D_k & \forall k \in X_3 \\
\sum_i x_{ij} - \sum_k x_{jk} = 0 & \forall j \in X_2 \\
0 \leq x_{ij} \leq s_{ij} & \forall j \in X_2 \cap J_1 \\
0 \leq x_{ij} \leq s_{ij} y_j' & \forall j \in X_2 \cap J_L \\
0 \leq x_{jk} \leq s_{jk} & \forall k \in X_3,\ j \in J(k) \\
y_i', y_j' \in \{0,1\} & \forall i, \forall j \in J_L
\end{array}\right.
$$

Le *programme d'état* PE_n s'obtient à partir du programme P_n en faisant :

$$- y_i' = 1 \quad \forall i \in J_{L1} \quad \text{et} \quad y_j' = 1 \quad \forall j \in J_{L1}$$
$$- y_i' = 0 \quad \forall i \in J_{L2} \quad \text{et} \quad y_j' = 0 \quad \forall j \in J_{L1}$$

Nous remarquons alors que *le second membre des contraintes n'est pas le même dans les programmes* P_n *et* PE_n.

Pour éviter cet inconvénient nous allons procéder à un changement de variable dans le programme P_n.

Changement de variable

$$y_\ell = y_\ell' \quad : \forall \ell \in J_{L2}$$
$$y_\ell = 1 - y_\ell' \quad : \forall \ell \in J_{L1}$$

ℓ *désignant* i *ou* j ; *le programme* P_n *s'écrit :*

$$\min cx + \sum_{\ell \in J_1} f_\ell + \sum_{\ell \in J_{L1}} f_\ell - \sum_{\ell \in J_{L1}} f_\ell y_\ell + \sum_{\ell \in J_{L2}} f_\ell y_\ell$$

V_j : $\sum_i x_{ij} - \sum_k x_{jk} = 0 \qquad \forall j \in X_2$

W_k : $-\sum_j x_{jk} \leq -D_k \qquad \forall k \in X_3$

U_i :
$$\sum_j x_{ij} \leq P_i \qquad \forall i \in X_1 \cap J_1$$
$$\sum_j x_{ij} + y_i P_i \leq P_i \qquad \forall i \in X_1 \cap J_{L1}$$
$$\sum_j x_{ij} - y_i P_i \leq 0 \qquad \forall i \in X_1 \cap J_{L2}$$

V_{ij} :
$$x_{ij} \leq s_{ij} \qquad \forall j \in J_1$$
$$x_{ij} \leq 0 \qquad \forall j \in J_0$$
$$x_{ij} + y_j s_{ij} \leq s_{ij} \qquad \forall j \in J_{L1}$$
$$x_{ij} - y_j s_{ij} \leq 0 \qquad \forall j \in J_{L2}$$

$$x_{ij}, x_{jk} \geq 0$$

$$y_\ell \in \{0,1\} \qquad \forall \ell \in J_L$$

Aux contraintes du programmes P_n attachons les variables duales V_j, W_k, U_i, V_{ij}.

Le programme d'état PE_n s'obtient en faisant $y_\ell = 0, \forall \ell \in J_{L1} \cup J_{L2}$.

Pour résoudre le programme PE_n par l'algorithme "out-of-kilter" il faut transformer ce problème en un problème de circulation défini sur un réseau réduit fermé ; l'entrée-sortie est appelée e ; $E = J_1 \cup J_{L1}$;

$$
\left\lceil
\begin{array}{lll}
 & \min cx + \sum_{\ell \in E} f_\ell & \\
\pi_j & \sum_i x_{ij} - \sum_k x_{jk} = 0 & \forall j \in E \\
\pi_e & \sum_i x_{ei} - \sum_k x_{ke} = 0 & \forall k, \forall i \in E \\
\pi_k & \sum_j x_{jk} - x_{ke} = 0 & \forall k \\
\pi_i & x_{ei} - \sum_j x_{ij} = 0 & \forall i \in E \\
v_{ei} & 0 \leq x_{ei} \leq P_i & \forall i \in E \\
v_{ij} & 0 \leq x_{ij} \leq s_{ij} & \forall k \\
v_{ke} & D_k \leq x_{ke} \leq D_k & \\
v_{jk} & 0 \leq x_{jk} \leq s_{jk} &
\end{array}
\right.
$$

Soit π_j, π_e, π_k, π_i, v_{ei}, v_{ij}, v_{jk} les variables duales correspondantes. C'est à partir de ces variables duales optimales données par l'algorithme "out-of-kilter" que nous calculons les variables duales optimale du programme d'état PE_n.

Détermination des U_i

U_i est la variable duale attachée à la contrainte :

$$\sum_j x_{ij} \leq P_i$$

$$(\sum_j x_{ij} - x_{ei}) + x_{ei} \leq P_i \Longrightarrow x_{ei} \leq P_i$$

or à cette dernière contrainte est attachée la variable d'arc v_{ei}

$$v_{ei} = -(\pi_e - \pi_i + c_{ei}) = \pi_i - \pi_e \quad \text{car } c_{ei} = 0$$

d'où $$U_i = \pi_i - \pi_e \quad \forall i \in X_1 \cap (J_1 \cup J_{L1})$$

Détermination des V_j

Nous pourrions évidemment choisir $V_j = \pi_j$ mais pour garder une signification économique aux variables duales V_{ij} nous prenons :

$$V_j = \pi_e - \pi_j$$

Ceci est valide ; en effet V_j correspond à la contrainte :

$$\sum_i x_{ij} - \sum_k x_{jk} = 0$$

$$(\sum_i x_{ij} - \sum_k x_{jk}) - (\sum_i x_{ei} - \sum_k x_{ke}) = 0$$

et à cette inégalité est attachée : $\pi_e - \pi_j$.

Détermination de W_k

W_k est déterminé par la contrainte (2) du programme dual D_n

$$W_k = \max (0, \min_j (c_{jk} - V_j))$$

Détermination des V_{ij}

Considérons la contrainte (1) du programme dual D_n ; on peut prendre

$$\begin{aligned} V_{ij} &= \max (0, - c_{ij} - U_i - V_j) \\ &= \max (0, - c_{ij} - \pi_i + \pi_e - \pi_e + \pi_j) \\ &= \max (0, - \bar{c}_{ij}) \end{aligned}$$

V_{ij} *est un coût réduit.*

Il faut s'assurer que les variables duales ainsi déterminées satisfont aux contraintes du programme dual du programme PE_n.

En notant $E = J_1 \cup J_{L1}$ ce programme dual s'écrit :

$$D_n \quad \begin{cases} \min - \left(\sum_{i \in E} U_i P_i - \sum_k W_k D_k + \sum_{j \in E} \sum_i V_{ij} s_{ij} \right) & \\ (1) \quad c_{ij} + U_i + V_j + V_{ij} \geq 0 & \forall j \in E \\ (2) \quad c_{jk} - V_j - W_k \geq 0 & \forall j \in E \\ (3) \quad U_i, W_k, V_{ij} \geq 0 & \end{cases}$$

- (1) : vérifiées

- (2) : vérifiées

- (3) : - $W_k \geq 0$
  - $V_{ij} \geq 0$
  - si l'origine i est utilisée l'arc (e,i) est en état β ou en état γ c'est-à-dire que

$$\bar{c}_{ei} = \pi_e - \pi_i + c_{ei} = \pi_e - \pi_i \leq 0$$

  U_i est alors positif ou nul.

  - si l'origine i n'est pas utilisée l'arc (e,i) peut se trouver dans l'état α et alors U_i est négatif.
  Nous prendrons donc $U_i = \max (0, \pi_i - \pi_e)$.

b) Etablissement des contraintes de BENDERS

Rappels (Chapitre II, paragraphe A)

Considérons le programme

$$(1)\quad \begin{cases} \min z = cx + fy \\ Ax + By \leq b \\ x \geq 0 \quad y_j \in \{0,1\} \end{cases}$$

Soit z^* la meilleure solution déjà obtenue à une certaine itération de l'algorithme de BENDERS ; soit y^* la valeur correspondante du vecteur booléen y. Pour que l'on puisse trouver une solution améliorante y, il faut que

$$(f + uB)\, y - ub \leq z^*$$

où u est la variable duale attachée à la contrainte (1).

Notations

au noeud γ de niveau n

Z	:	valeur de la fonction économique du programme P
Z_n	:	valeur de la fonction économique du programme P_n
ZE_n	:	valeur de la fonction économique du programme d'état PE_n.

Le signe $\hat{}$, sur ces quantités, indique les valeurs optimales correspondantes.
Le signe *, sur les quantités Z et Z_n, représente les meilleures valeurs déjà trouvées.

$E = (J_1 \cup J_{L1})$.

Le programme P auquel nous nous intéressons est celui formulé page 75.

Au noeud γ, la contrainte de BENDERS s'écrit :

$$(1)\quad \left| \begin{array}{l} \sum_{i \in J_{L1}} (-f_i + U_i P_i) y_i + \sum_{i \in J_{L2}} (f_i - U_i P_i) y_i \\ + \sum_{j \in J_{L1}} (-f_j + \sum_i V_{ij} s_{ij}) y_j + \sum_{j \in J_{L2}} (f_j - \sum_i V_{ij} s_{ij}) y_j \\ \leqslant Z^* - \hat{ZE}_n \end{array} \right.$$

En effet, d'une part :

$$- u^{\gamma} b = - \sum_{i \in E} U_i P_i + \sum_k W_k D_k - \sum_{j \in E} (\sum_i V_{ij} s_{ij})$$

d'autre part :

$$Z^* = Z_n^* + \sum_{\ell \in E} f_\ell$$

Introduisons les notions de *pénalités* et d'*évaluations*.

Posons : ℓ représentant i ou j,

p_ℓ^o (resp. p_ℓ^1) : la pénalité sur la fonction économique si l'on fixe la variable y_ℓ à zéro (resp. à un) : ce sont des quantités positives

ev_ℓ^o (resp. ev_ℓ^1) : une évaluation de p_ℓ^o (resp. p_ℓ^1).

Pour les $\ell \in J_{L1}$ on pose :

* $p_\ell^1 = f_\ell$

* $ev_\ell^o = \begin{cases} U_i P_i & \text{si } \ell = i \\ \sum_i V_{ij} s_{ij} & \text{si } \ell = j \end{cases}$

Pour les $\ell \in J_{L2}$

* $p_\ell^0 = f_\ell$

* $ev_\ell^1 = \begin{cases} U_i P_i & \text{si } \ell = i \\ \sum_i V_{ij} s_{ij} & \text{si } \ell = j \end{cases}$

La condition (1) peut s'écrire :

$$\sum_{\ell \in J_{L1}} (-p_\ell^1 + ev_\ell^0) y_\ell + \sum_{\ell \in J_{L2}} (p_\ell^0 - ev_\ell^1) y_\ell \leq Z - \hat{ZE}_n$$

Pour que l'on puisse utiliser efficacement cette contrainte il faut que les évaluations ev_ℓ^0 et ev_ℓ^1 soient les plus proches possibles des pénalités exactes p_ℓ^0 et p_ℓ^1.

Remarque

Si $\ell = j$: ni dans ev_j^0, ni dans ev_j^1 ne figurent sous une forme ou sous une autre les quantités $\bar{c}_{jk}$ qui sont des coûts réduits.

Pour construire de meilleures évaluations des pénalités p_ℓ^0 et p_ℓ^1 nous ajoutons des *contraintes redondantes* au programme initial ; le programme P formulé page 75 s'écrit alors :

$$\begin{cases} \min cx + \sum_{i \in X_1} f_i y_i' + \sum_{j \in X_2} f_j y_j' \\ \sum_{j \in J(i)} x_{ij} \leq y_i' P_i & \forall i \in X_1 \\ -\sum_{k \in K(j)} x_{jk} \leq -D_k & \forall k \in X_3 \\ \sum_{i \in I(j)} x_{ij} - \sum_{k \in K(j)} x_{jk} = 0 & \forall j \in X_2 \\ 0 \leq x_{ij} \leq y_j' s_{ij} \\ 0 \leq x_{jk} \leq y_j' s_{ij} \\ y_j' \in \{0,1\},\ y_i' \in \{0,1\} \end{cases}$$

De ce programme on déduit le programme P_n dans lequel on fait le changement de variables :

$$y_\ell = y'_\ell \qquad \text{si } \ell \in J_{L2}$$

$$y = 1 - y' \text{ si } \in J_{L1}$$

ℓ étant égal à i ou à j.

La condition (1) s'écrit alors :

$$\left| \begin{array}{l} \sum_{i \in J_{L1}} (-f_i + U_i P_i) y_i + \sum_{i \in J_{L2}} (f_i - U_i P_i) y_i \\ + \sum_{j \in J_{L1}} (-f_j + \sum_i V_{ij} s_{ij} + \sum_k V_{jk} s_{jk}) y_j + \sum_{j \in J_{L2}} (f_j - \sum_i V_{ij} s_{ij} - \sum_k V_{jk} s_{jk}) y_j \\ \leq Z^* - \hat{ZE}_n \end{array} \right.$$

où $V_{jk} = \max (0, -\bar{c}_{jk})$ $\forall (j,k) \in U$.

les évaluations des pénalités p^0_ℓ et p^1_ℓ sont :

$\underline{\ell = i}$

$$ev^0_i = U_i P_i \qquad \forall i \in J_{L1}$$

$$ev^1_i = U_i P_i \qquad \forall i \in J_{L2}$$

$\underline{\ell = j}$

$$ev^0_j = \sum_i V_{ij} s_{ij} + \sum_k V_{jk} s_{jk} \qquad \forall j \in J_{L1}$$

$$ev^1_j = \sum_i V_{ij} s_{ij} + \sum_k V_{jk} s_{jk} \qquad \forall j \in J_{L2}$$

III . RENFORCEMENT DES CONTRAINTES DE BENDERS

Deux problèmes se posent :

1) Comment peut-on déterminer les évaluations ev^1_ℓ ?

2) Peut-on renforcer les évaluations ev^0_ℓ et ev^1_ℓ ?

I - Détermination de l'évaluation ev^1_ℓ

Nous sommes dans le cas où $\ell \in J_{L2}$, c'est-à-dire que $y^1_\ell = 0$.
La résolution du programme PE_n ne nous fournit aucun renseignement sur les variables duales U_i, V_{ij} et V_{jk} nécessaires au calcul de ev^1_ℓ, puisque le sommet ℓ correspondant n'existe pas dans le sous-graphe G(X',U') sur lequel est défini ce programme.

Il existe évidemment un moyen de calculer p^1_ℓ c'est de résoudre le programme correspondant à l'ouverture du sommet ℓ; ceci augmente le temps de résolution du programme initial, d'autant qu'il faut le faire pour tous les i et j de l'ensemble J_{L2} et ce, à chaque noeud de l'arborescence.

Nous proposons un autre moyen de déterminer ev^1_ℓ.

A) Cas où $\ell = j$

Soit j *le transit* dont on envisage "l'existence".

1) Déterminons une variable nodale $\pi'(j)$ en mettant tous les arcs $(i,j) \in U'$ en état conforme, c'est-à-dire en état α ou en état β, donc tels que :

$$\bar{c}_{ij} = \pi(i) + c_{ij} - \pi'(j) \geq 0 \qquad \forall (i,j) \in U'$$

d'où

$$\pi'(j) = \min_{i \in \{J_1 \cup J_{L1}\}} (\pi(i) + c_{ij})$$

2) Déterminons une variable nodale $\pi''(j)$ en mettant tous les arcs $(j,k) \in U'$ en état conforme

$$\pi''(j) = \max_{k \in K(j)} (\pi(k) - c_{jk})$$

. *Au sommet* j faisons correspondre *deux sommets* j_1 et j_2 ;

Posons :

$$\pi(j_1) = \pi'(j)$$

$$\pi(j_2) = \pi''(j)$$

$$c_{j_1 j_2} = 0$$

$$s_{j_1 j_2} = +\infty$$

$$i_{j_1 j_2} = 0$$

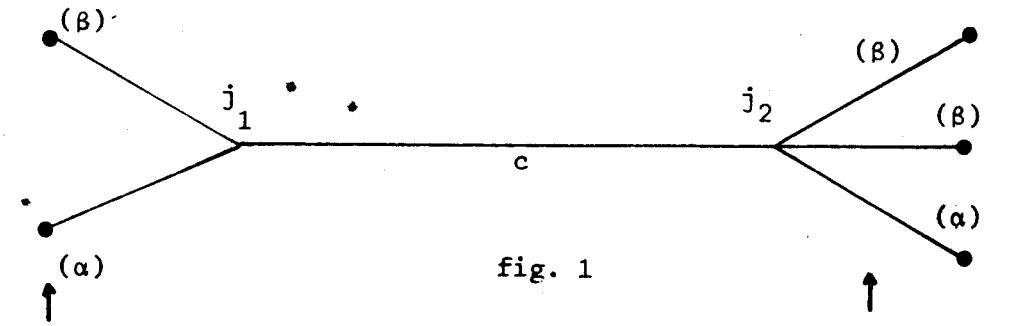


fig. 1

arcs (i,j) : $i \in X'$
$(i,j) \in U$

arcs (j,k) : $k \in X'$
$(j,k) \in U$

Propriété 1

Si $\pi'(j) - \pi''(j) \geq 0$ alors $ev_j^1 = 0$.

Démonstration

Insérons le graphe de la figure 1 dans le graphe G(X',U') : soit G(X",U") le graphe obtenu. Complétons le flot optimal circulant sur le premier graphe en faisant passer un flux nul sur tous les arcs du graphe "ajouté". Ce flot est optimal pour le programme défini sur le graphe G(X",U") ; par conséquent $ev_j^1 = 0$.

Intéressons nous maintenant au cas où $\pi'(j) - \pi''(j) < 0$.

L'arc $(j_1 j_2)$ n'est plus conforme : en effet

$$- x_{j_1 j_2} = 0$$

$$- \bar{c}_{j_1 j_2} = \pi(j_1) + c_{j_1 j_2} - \pi(j_2) \leq 0$$

Si l'on pose :

$$(1) \quad c_{j_1 j_2} = \pi(j_1) - \pi(j_2)$$

alors cet arc est conforme.

Cela signifie que le flot optimal du programme défini sur le graphe G(X',U'), complété comme il a été dit dans la démonstration précédente, est encore optimal pour le programme défini sur le graphe G(X",U") modifié par la condition (1).

Autrement dit l'algorithme "out-of-kilter" appliqué au programme défini sur le graphe G(X",U") initial donnera les mêmes valeurs aux variables nodales des sommets j_1 et j_2 *en compensant* "la dépense de passage" sur l'arc (j_1, j_2).

A toute unité de flux qui passe sur l'arc (j_1, j_2) correspond un coût négatif :

$$c = \pi'(j) - \pi''(j)$$

Nous proposons trois valeurs pour l'évaluation ev^1_j.

Première évaluation

Posons :

$$Q = \min \left(\sum_i s_{ij}, \sum_k s_{jk} \right) \qquad \forall (i,j) \text{ et } \forall (j,k) \in U''$$

En "ignorant" les arcs par lesquels le flux rentre et sort de l'arc (j_1,j_2) on peut prendre pour ev^1_j la valeur :

$$m_1 = -(\pi'(j) - \pi''(j)).Q$$

cette évaluation est évidemment assez grossière

Deuxième évaluation

Nous avons déterminé :

$$\pi'(j) = \min_{(i,j) \in U''} (\pi(i) + c_{ij})$$

Pour procéder à cette deuxième évaluation on ne considère que la partie "gauche" du graphe, c'est-à-dire que l'on ne prend en considération que l'état des arcs par lesquels le flux est susceptible de rentrer dans l'arc (j_1,j_2).

Soit :

$$S_1 = \sum_{(i,j_1)} s_{ij_1}.c \qquad \forall (i,j_1) \in U'' : \bar{c}_{ij_1} = 0$$

$$S_2 = \sum_{(i,j_1)} s_{ij_1}(c - \bar{c}_{ij_1}) \qquad \forall (i,j_1) \in U'' : \bar{c}_{ij_1} > 0$$

où $c = -(\pi'(j) - \pi''(j))$.

On peut prendre pour ev_j^1 la valeur :

$$m_2 = S_1 + S_2$$

Remarque

. *On peut améliorer cette évaluation* dans le cas où l'origine d'un des arcs considérés envoie toute sa production vers *un seul transit* j'

- si l'arc est en état β :

dans S_1 remplacer, $s_{ij_1}.c$ par $\begin{cases} s_{ij_1}(\bar{c} + \bar{c}_{ij'}) \text{ si } \bar{c} + \bar{c}_{ij'} > 0 \\ 0 \text{ sinon} \end{cases}$

- si l'arc est en état α :

dans S_2 remplacer $s_{ij_1}(c-c_{ij_1})$ par $\begin{cases} s_{ij_1}(\bar{c}-\bar{c}_{ij_1} \text{ si } \bar{c}-\bar{c}_{ij_1}+\bar{c}_{ij'} > 0 \\ 0 \text{ sinon} \end{cases}$

Troisième évaluation

On ne considère que la partie "droite" du graphe, c'est-à-dire que l'on ne prend en considération que l'état des arcs par lesquels le flux est susceptible de sortir de l'arc (j_1,j_2).

Soit :

$$S_1' = \sum_k s_{j_2k}.c \qquad \forall (j_2,k) \in U'' : \bar{c}_{j_2k} = 0$$

$$S_2' = \sum_k s_{j_2k}(\bar{c} - \bar{c}_{j_2k}) \qquad \forall (j_2,k) \in U'' : \bar{c}_{j_2k} < c$$

On peut prendre pour ev_j^1 la valeur :

$$m_3 = S_1' + S_2'$$

Remarque

On peut améliorer cette évaluation dans le cas où *l'une des destination* k *est servie à partir d'un seul transit* j' et que *l'arc* (j',k) *est en état* γ.

- si l'arc (j_2,k) est en état β :

dans S_1' remplacer $s_{j_2k}.c$ par $\begin{cases} s_{j_2k}(c + \bar{c}_{j'k}) \text{ si } c + \bar{c}_{j'k} > 0 \\ 0 \text{ sinon} \end{cases}$

- si l'arc (j_2,k) est en état α :

dans S_2' remplacer $s_{j_2k}(c-\bar{c}_{j_2k})$ par : $\begin{cases} s_{j_2k}(c-\bar{c}_{j_2k}+\bar{c}_{j'k}) \text{ si } c-\bar{c}_{j_2k}+\bar{c}_{j'k}>0 \\ 0 \text{ sinon} \end{cases}$

Propriété

Si $\pi'(j) - \pi''(j) < 0$ alors $ev_j^1 = \min(m_1, m_2, m_3)$ est une évaluation par excès de p_j^1.

Rappelons que :

- G(X,U) est le graphe sur lequel est défini le programme P initial
- G(X',U') est le graphe sur lequel est défini le programme d'état PE_n.

Notons :

- $G(X_j,U_j)$ le sous-graphe de G(X,U) défini par : $X_j = \{X' \cup \{j\}\}$

Appliquons l'algorithme "out-of-kilter" au programme défini sur le graphe $G(X_j, U_j)$: si le nouveau flot est tel que :

1) Pour *entrer* ou pour *sortir* du transit j il n'emprunte que des arcs (i,j) mis précédemment :

$$\text{en état } \beta$$
$$\text{en état } \gamma \text{ avec } \bar{c}_{ij} = -(\pi'(j) - \pi''(j))$$

2) Que *tout arc précédemment en état* γ (non compris ceux qui seraient éventuellement intervenus dans la somme S_2' de la remarque ci-dessus) est toujours *saturé*

alors *on peut affirmer* que $p_j^1 = ev_j^1$; si l'une des conditions ci-dessus n'est plus satisfaite il s'ensuit une "diminution" de p_j^1 et l'on a $p_j^1 \leq ev_j^1$

B) Cas où ℓ=i

Soit i l'origine dont en envisage l'ouverture

- déterminons une variable nodale $\pi'(i)$ en la prenant égale à la variable nodale du sommet-entrée :

$$\pi'(i) = \pi(e)$$

- déterminons une variable nodale $\pi''(i)$ en mettant tous les arcs $(i,j) \in U'$ en état conforme :

$$\pi''(i) = \max_{(i,j)} (\pi(j) - c_{ij})$$

Propriété

Si $\pi'(i) - \pi''(i) \geq 0$ alors $ev_i^1 = 0$.

Démonstration

Elle est identique à celle faite pour un transit j ; ici l'on créé deux sommets i_1 et i_2.

On pose :

$$\pi(i_1) = \pi'(i)$$

$$\pi(i_2) = \pi''(i)$$

$$c_{i_1 i_2} = 0$$

$$s_{i_1 i_2} = 0$$

$$i_{i_1 i_2} = 0$$

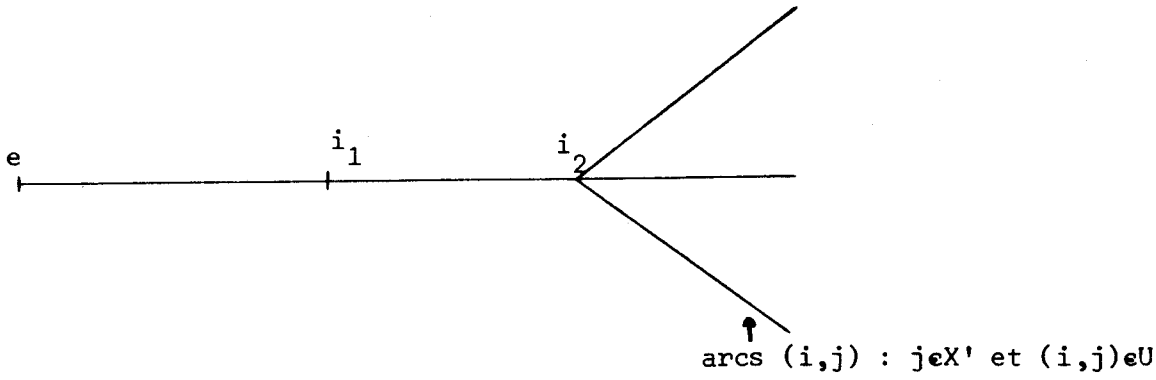


fig. 2

et l'on insère le graphe de la figure 2 dans le graphe G(X',U') : soit G(X",U") le graphe obtenu.

Intéressons nous au cas où $\pi'(i) - \pi''(i) < 0$.

Comme dans le cas où $\ell = i$ nous proposons des évaluations de p_i^1.

Première évaluation

$$m_1 = c.P_i$$

où $c = -(\pi'(i) - \pi''(i))$.

Deuxième évaluation

Considérons l'état des arcs par lesquels est susceptible de passer le flux sortant de l'arc (i_1,i_2).

Soit :

$$S_1 = \sum_j s_{i_2 j} . c\ (i_2, j) \qquad \forall U'' : \bar{c}_{ij} = 0$$

$$S_2 = \sum_j s_{i_2 j} (c - \bar{c}_{i_2 j}) \qquad \forall (i_2, j) \in U'' : c - \bar{c}_{i_2 j} > 0$$

$$m_2 = S_1 + S_2$$

C'est cette évaluation que nous retiendrons.

Propriété

Si $\pi'(i) - \pi''(i) < 0$ alors $ev_i^1 = m_2$ est une évaluation par excès de P_i^1.

Le raisonnement est identique à celui fait précédemment dans le cas où $\ell = j$.

2 - Renforcement des évaluations ev_1^0 et ev_1^1

Une évaluation ne doit pas être faite au détriment des autres, c'est-à-dire qu'elle ne doit pas nécessiter la modification de certaines quantités (variables duales, coûts réduits par exemple) qui doivent être utilisées dans d'autres évaluations.

Rappelons quelques notations :

- G(X,U) : graphe sur lequel est défini le programme initial P
- G(X',U') : graphe sur lequel est défini le programme d'état PE_n
- $J_1 = \{\ell : y'_\ell = 1\}$ et $J_o = \{\ell : y'_\ell = 0\}$ (variables fixées)
  $J_{L1} = \{\ell : y'_\ell = 1\}$ et $J_{L2} = \{\ell : y'_\ell = 0\}$ (variables libres)

Soit G(X",U") le *sous-graphe* de G(X,U) déterminé par :

$$X'' = \{\ell \in J_1 \cup J_{L1} \cup J_{L2}\}$$

Nous proposons un *algorithme de renforcement* des évaluations ev^0_ℓ et ev^1_ℓ, qui utilisera en sous-algorithme la détermination des évaluations ev^1_ℓ proposée au paragraphe 1 précédent.

ALGORITHME

1 - Diminuer les variables nodales des destinations k qui sont servies par un ou plusieurs transits définitivement ouverts

$$\text{faire } \pi(k) = \pi(k) + d$$

$$\text{où } d = \max_{j \in X'} [\bar{c}_{jk} \leq 0 : 0 < x_{jk} \leq s_{jk}]$$

2 - Déterminer les évaluations ev^1_ℓ, de tous les sommets ℓ temporairement fermés ; calculer les coûts réduits correspondants

$$\text{- si } \ell = i \; : \quad \bar{c}_{ij} = \pi''(i) + c_{ij} - \pi(j) \qquad \forall j \in X'$$

$$\text{- si } \ell = j \; : \quad \bar{c}_{ij} = \pi(i) + c_{ij} - \pi'(j) \qquad \forall i \in X'$$

$$\bar{c}_{jk} = \pi''(j) + c_{ij} - \pi(k) \qquad \forall k \in X'$$

3 - Augmenter les variables nodales des destinations k servies par un seul transit temporairement ouvert

$$\text{faire } \pi(k) = \pi(k) + d$$

$$\text{où } d = \min_{j \in X''} [\bar{c}_{jk} \geq 0 : 0 \leq x_{jk} < s_{jk}]$$

4 - Diminuer les variables nodales des origines définitivement ouvertes

$$\text{faire } \pi(i) = \pi(i) - d$$

$$\text{où } d = \min (d_1, d_2)$$

$$d_1 = - \bar{c}_{e_i}$$

$$d_2 = \min_{j \in X''} [\bar{c}_{ij} \geq 0 : x_{ij} < s_{ij}]$$

5 - Augmenter les variables nodales des origines temporairement ouvertes et envoyant toute leur production vers des transits définitivement ouverts

faire $\pi(i) = \pi(i) - d$

où $d = \min(d_1, d_2)$

$d_1 = +\infty$ si $x_{ei} = s_{ei}$

$= 0$ sinon

$d_2 = \max_{j \in X''} [\bar{c}_{ij} \leqslant 0 : 0 < x_{ij} \leqslant s_{ij}]$

6 - Calculer les évaluations ev^o_ℓ

Reprenons une à une les différentes phases de l'algorithme

1 - Cette phase permet un renforcement des évaluations ev^1_j.
La restriction "transits définitivement ouverts" empêche la détérioration des évaluations ultérieures ev^o_j.

2 - Application directe du sous-algorithme présenté au paragraphe précédent.
Les coûts réduits calculés seront utilisés dans les phases suivantes.

3 - Cette phase permet un renforcement des évaluations ultérieures ev^o_j.
Les restrictions a) "$j \in X$" dans le calcul de d
b) "servies par un seul transit temporairement ouvert"
garantissent la non-détérioration des autres évaluations ev^1_j précédemment calculées.

4 - Cette phase permet également un renforcement des évaluations ultérieures ev^o_j.

5 - Cette phase permet un renforcement des évaluations ev^o_i et les restrictions apportées permettent d'assurer la non-détérioration des autres évaluations.

6 - Cette phase exécute les calculs demandés à l'aide des formules

$$ev^o_i = U_i P_i$$
$$ev^o_j = \sum_i V_{ij} s_{ij} + \sum_k V_{jk} s_{jk}$$

avec $U_i = \max [0, \pi(i) - \pi(e)]$

$V_{ij} = \max [0, -\bar{c}_{ij}]$

$V_{jk} = \max [0, -\bar{c}_{jk}]$

Rappelons que *les évaluations* ev^0_ℓ *sont des évaluations par défaut* des pénalités p^0_ℓ, tandis que *les évaluations* ev^1_ℓ *sont des évaluations par excès* des pénalités p^1_ℓ.

C . EXPLOITATION DES INÉGALITÉS

Rappels

Notations

$$J_1 = \{\ell : y'_\ell = 1\} \text{ et } J_o = \{\ell : y' = 0\} \qquad \text{(variables fixées)}$$

$$J_{L1} = \{\ell : y'_\ell = 1\} \text{ et } J_{L2} = \{\ell : y'_\ell = 0\} \qquad \text{(variables libres)}$$

Formulation du problème auquel nous nous intéressons

$$\left[\begin{array}{ll}
\min cx + \sum_{i \in X_1} f_i y'_i + \sum_{j \in X_2} f_j y'_j & \\
\sum_{j \in J(i)} x_{ij} \leq y'_i P_i & \forall i \in X_1 \\
-\sum_{j \in J(k)} x_{jk} \leq -D_k & \forall k \in X_3 \\
\sum_{i \in I(j)} x_{ij} - \sum_{k \in K(j)} x_{jk} = 0 & \forall j \in X_2 \\
\left\{\begin{array}{l} 0 \leq x_{ij} \leq y'_j s_{ij} \\ 0 \leq x_{jk} \leq y'_j s_{jk} \end{array}\right. & \\
y'_\ell \in \{0,1\} & \forall \ell \in X_1 \cup X_2
\end{array}\right.$$

Après avoir effectué les changements de variables

$$y_\ell = 1 - y'_\ell \qquad \forall \ell \in J_{L1}$$

$$y_\ell = y'_\ell \qquad \forall \ell \in J_{L2}$$

La résolution du programme d'état PE_n nous permet d'écrire la contrainte de BENDERS :

$$(1)\quad \left[\begin{array}{l} \sum_{i\in J_{L1}} (-f_i + U_i P_i) y_i + \sum_{i\in J_{L2}} (f_i - U_i P_i) y_i \\ + \sum_{j\in J_{L1}} (-f_j + \sum_i V_{ij} s_{ij} + \sum_k V_{jk} s_{jk}) \\ + \sum_{j\in J_{L2}} (f_j - \sum_i V_{ij} s_{ij} - \sum_k V_{jk} s_{jk}) \\ \leq Z^* - \hat{ZE}_n \end{array}\right.$$

où $\hat{ZE}_n$ est la valeur de la fonction économique correspondant à la solution optimale du programme d'état PE_n.

Z^* est la meilleure valeur connue de la fonction économique du programme initial P.

Z^* peut être *obtenue* :

- soit à partir d'une solution réalisable déjà atteinte dans l'exploration de l'arborescence ;
- soit à partir d'une solution donnée par une procédure heuristique.

Remarque

Si l'on n'a aucun renseignement on prend $Z^* = +\infty$.

Nous supposons que $Z^* - \hat{ZE}_n \leq 0$; en effet le cas où cette quantité est positive conduit à une amélioration de Z^*

Posons, comme nous l'avons fait précédemment,

pour $\ell \in J_{L1}$ $\quad p_\ell^1 = f_\ell$

ev_ℓ° : évaluation de p_ℓ°

pour $\ell \in J_{L2}$ $\quad p_\ell^\circ = f_\ell$

ev_ℓ^1 : évaluation de p_ℓ^1

L'inégalité (1) s'écrit :

$$(2) \qquad \sum_{\ell \in J_{L1}} (-p_\ell^1 + ev_\ell^\circ) y_\ell + \sum_{\ell \in J_{L2}} (p_\ell^\circ - ev_\ell^1) y_\ell \leq Z^* - \hat{ZE}_n$$

Rappelons que ev_ℓ° est une évaluation par *défaut* de p_ℓ° et que ev_ℓ^1 est une évaluation par *excès* de p_ℓ^1.

I. TEST A

Supposons que nous connaissons les valeurs exactes de p_ℓ° et de p_ℓ^1 $\forall \ell \in J_L$.

Posons :

$$PEN_\ell = |p_\ell^1 - p_\ell^\circ| \quad \forall \ell$$

PEN_ℓ représente une *pénalité* liée au choix de mettre y_ℓ à zéro ou à un

- si $\ell \in J_{L1}$:

1) $-p_\ell^1 + p_\ell^\circ < 0 \Longrightarrow p_\ell^1 > p_\ell^\circ$

le choix de mettre ℓ dans l'ensemble J_{L1} ne correspond pas à celui donné par le minimum de p^1 et de p°.
PEN_ℓ représente une *perte* sur la fonction économique ; on dit aussi un *regret*.

2) $-p_\ell^1 + p_\ell^\circ > 0 \Longrightarrow p_\ell^\circ > p_\ell^1$

PEN_ℓ est un *gain* lié au choix d'avoir mis ℓ dans l'ensemble J_{L1}.

- si $\ell \in J_{L2}$:

3) $p_\ell^o - p_\ell^1 < 0 \Longrightarrow p_\ell^1 > p_\ell^o$

PEN_ℓ est une perte liée au choix d'avoir mis ℓ dans l'ensemble J_{L2}.

4) $p_\ell^o - p_\ell^1 > 0 \Longrightarrow p_\ell^o > p_\ell^1$

le choix de mettre ℓ dans l'ensemble J_{L2} ne correspond pas à celui donné par le minimum de p_ℓ^o et p_ℓ^1. PEN_ℓ représente un *gain* sur la fonction économique.

Revenons à notre problème ; nous ne connaissons que des valeurs approchées des pénalités.
Nous ne pouvons pas conclure dans les quatre cas envisagés ci-dessus mais seulement dans *deux cas suivants* :

Enoncé du test A

- si $\ell \in J_{L1}$ et si $-p_\ell^1 + ev_\ell^o > 0$ alors *transférer* ℓ de l'ensemble J_{L1} à l'ensemble J_1.
  En effet $-p_\ell^1 + ev_\ell^o > 0 \Longrightarrow ev_\ell^o > p_\ell^1 \Longrightarrow p_\ell^o > p_\ell^1$

- si $\ell \in J_{L2}$ et si $p_\ell^o - ev_\ell^1 > 0$ alors *transférer* ℓ de l'ensemble J_{L2} à l'ensemble J_0.
  En effet : $p_\ell^o - ev_\ell^1 > 0 \Longrightarrow ev_\ell^1 < p_\ell^o \Longrightarrow {}^1\ell < p_\ell^o$

II. TEST B

Pour que l'inégalité $\sum_k a_k y_k \leq b$

où $a_k < 0, \forall k$

$b < 0$

$y_k \in \{0,1\} \ \forall k$

soit réalisable il faut que $\sum_k |a_k| > |b|$.

Exemples a) $-5y_1 - 3y_2 \leq -2$

b) $-5y_1 - 3y_2 \leq -18$

Considérons la contrainte de Benders engendrée en un noeud γ de l'arborescence.

Après application du test A elle se met sous la forme

$$\sum_{\ell} a_\ell y_\ell \leq b$$

$$a_\ell = - f_\ell + ev^0_\ell < 0 \qquad \forall \ell \in J_{L1}$$

$$a_\ell = f_\ell - ev^1_\ell < 0 \qquad \forall \ell \in J_{L2}$$

$$b < 0$$

$$y_\ell \in \{0,1\} \qquad \forall \ell$$

"Compte-tenu que les a_ℓ ne sont que des évaluations, si cette inégalité n'est pas réalisable peut-on conclure à une régression dans l'exploration de l'arborescence ?"

La réponse est affirmative.

Posons :

$$\bar{a}_\ell = -f_\ell + p^0_\ell \qquad \forall \ell \in J_{L1}$$

$$\bar{a}_\ell = f_\ell - p^1_\ell \qquad \forall \ell \in J_{L2}$$

1) supposons que $\bar{a}_\ell < 0$
si $\ell \in J_{L1}$ ou $\ell \in J_{L2}$ $\Longrightarrow$ $|a_\ell| > |\bar{a}_\ell|$ on peut donc conclure à une régression

2) supposons que $\bar{a}_\ell > 0$
le test A fait disparaître le terme $a_\ell y_\ell$ de l'inégalité qui, *à fortiori*, ne sera plus réalisable. On peut donc conclure, dans ce cas également, à une régression.

Enoncé du test B

Si l'inégalité de BENDERS n'est pas réalisable elle implique un "pas-en-arrière" dans l'exploration de l'arborescence.

III. TEST C

a) Considérons l'inégalité :

$$(1) \qquad \sum_k a_k y_k \leq b$$

avec $a_k > 0 \quad \forall k$

$b > 0$

$y_k \in \{0,1\} \quad \forall k$

- s'il existe des k_i tels que $a_{k_i} > b$ alors pour que cette inégalité puisse être réalisée il faut que tous les y_{k_i} soient égaux à zéro.

b) Considérons l'inégalité :

$$(2) \qquad \sum_k a_k y_k \leq b$$

avec $a_k < 0 \quad \forall k$

$b < 0$

$y_k \in \{0,1\} \quad \forall k$

supposons que cette inégalité est réalisable, c'est-à-dire que :

$$\sum_k |a_k| > |b|$$

Faisons le changement de variables $\bar{y}_k = 1 - y_k$; elle s'écrit :

$$(2) \qquad \sum_k - a_k \bar{y}_k \leq b + \sum_k - a_k = b'$$

le premier membre est à termes positifs, le second membre est positif ou nul.

Remarque

Dans l'inégalité (2), *la suppression d'un seul terme quelconque* tel que $-a_{k_1} > b'$ a pour conséquence de rendre l'inégalité irréalisable.

Supposons, par exemple, qu'il existe deux termes $a_{k_1} y_{k_1}$ et $a_{k_2} y_{k_2}$ tels que :

$$-a_{k_1} > b' \quad \text{et} \quad -a_{k_2} > b'$$

l'inégalité (2') peut s'écrire : (les sommes portant sur k, $k \neq k_1$ et $k \neq k_2$)

$$(3) \qquad \sum(-a_k \bar{y}_k) - a_{k_1} \bar{y}_{k_1} - a_{k_2} \bar{y}_{k_2} \leq b + \sum(-a_k) - a_{k_1} - a_{k_2} = b'$$

Supprimons, par exemple, le terme $a_{k_1} y_{k_1}$ de l'inégalité (2) : de l'inégalité (3) on tire :

$$b + \sum(-a_k) - a_{k_2} = b' + a_{k_1} < 0$$

c'est-à-dire que :

$$\sum(|-a_k|) + |-a_{k_2}| \leq |b|$$

ce qui montre que l'inégalité est devenue irréalisable.

Exemples : a) $-2y_1 - 4y_2 - 18y_3 \leq -7$
$2\bar{y}_1 + 4\bar{y}_2 + 18\bar{y}_3 \leq 17$
réalisable si $\bar{y}_3 = 0 \Longrightarrow y_3 = 1$

b) $-80y_1 - 397y_{10} - 482y_{14} \leq -570$
$80\bar{y}_1 + 397\bar{y}_{10} + 482\bar{y}_{14} \leq 389$
inégalité réalisable si $\bar{y}_{14} = \bar{y}_{10} = 0$
$\Longrightarrow \quad y_{14} = y_{10} = 1$

On en déduit que :

"si dans l'inégalité (2) il existe des indices k_i tels que $-a_{k_i} > b'$ alors, pour que cette inégalité puisse être réalisée il faut que tous les $\bar{y}_{k_i}$ soient nuls c'est-à-dire que tous les y_{k_i} soient égaux à un".

Si l'inégalité est à termes quelconques il est facile par des changements de variables adéquates de la mettre sous la forme (1).

Considérons la *contrainte de BENDERS* engendrée à un noeud γ de l'arborescence.

Après application du test A et du test B, elle s'écrit :

$$(3) \qquad \sum_{\ell} a_\ell y_\ell \leq b$$

avec :

$$a_\ell = -f_\ell + ev^0_\ell \quad \text{si } \ell \in J_{L1}$$

$$a_\ell = f_\ell - ev^1_\ell \quad \text{si } \ell \in J_{L2}$$

cette inégalité est à termes négatifs.

Effectuons les changements de variables $\bar{y}_\ell = 1 - y_\ell \ \forall \ell$
l'inégalité (3) s'écrit :

$$(3') \qquad \sum_{\ell} - a_\ell \bar{y}_\ell \leq b + \sum_{\ell} - a_\ell = b'$$

- s'il existe un $\hat{\ell}$ tel que $- a_{\hat{\ell}} > b'$, peut-on en conclure que $\bar{y}_{\hat{\ell}}$ doit être nul pour que l'inégalité puisse être réalisée en un noeud successeur du noeud γ, compte-tenu que $a_{\hat{\ell}}$ n'est connu qu'avec une certaine approximation ?

  La réponse est affirmative.

Supposons par exemple que lors de l'établissement de l'inégalité (3) $\hat{\ell}$ *appartienne à l'ensemble* J_{L1}.

Posons $\bar{a}_{\hat{\ell}} = -f_{\hat{\ell}} + p^0_{\hat{\ell}}$

$a_{\hat{\ell}} \leq \bar{a}_{\hat{\ell}}$ puisque $ev^0_{\hat{\ell}}$ est une approximation par défaut de $p^0_{\hat{\ell}}$

a) Supposons que $\bar{a}_{\hat{\ell}}$ soit négatif

et considérons les deux inégalités suivantes :

$$(4) \qquad \sum_{\ell \neq \hat{\ell}} a_\ell y_\ell + a_{\hat{\ell}} y_{\hat{\ell}} \leq b$$

$$(5) \qquad \sum_{\ell \neq \hat{\ell}} a_\ell y_\ell + \bar{a}_{\hat{\ell}} y_{\hat{\ell}} \leq b$$

effectuons les changements de variables pour rendre ces inégalités à termes positifs ; elles s'écrivent :

$$(4') \qquad \sum_{\ell \neq \hat{\ell}} - a_\ell \bar{y}_\ell - a_{\hat{\ell}} \bar{y}_{\hat{\ell}} \leq b + \sum_{\ell \neq \hat{\ell}} - a_\ell - a_{\hat{\ell}} = b_1$$

$$(5') \qquad - a_\ell \bar{y}_\ell - \bar{a}_{\hat{\ell}} \bar{y}_{\hat{\ell}} \leq b + \sum_{\ell \neq \hat{\ell}} - a_\ell - \bar{a}_{\hat{\ell}} = b_2$$

nous voyons immédiatement que :

$$- a_{\hat{\ell}} > b_1 \Longrightarrow - \bar{a}_{\hat{\ell}} > b_2$$

en effet : $b_2 - b_1 = - \bar{a}_{\hat{\ell}} + a_{\hat{\ell}} \Longrightarrow b_2 + \bar{a}_{\hat{\ell}} = b_1 + a_{\hat{\ell}} < 0$

donc : $- \bar{a}_{\hat{\ell}} > b_2$

ceci nous permet de conclure.

Traduisons ce résultat en variable $y^!_{\hat{\ell}}$.

Nous avons supposé que $\hat{\ell}$ appartenait à l'ensemble J_{L1} donc :

$$\bar{y}_{\hat{\ell}} = 0 \Longrightarrow y_{\hat{\ell}} = 1 \Longrightarrow y^!_{\hat{\ell}} = 0$$

l'origine ou le transit $\hat{\ell}$ doit être fermé.

b) Supposons que $\bar{a}_{\hat{\ell}}$ soit positif

si $- a_{\hat{\ell}} > b'$ est-il encore licite de mettre $\bar{y}_{\hat{\ell}}$ à zéro, c'est-à-dire de fermer l'origine ou le transit correspondant ?

Si l'évaluation $a_{\hat{\ell}}$ avait donné la valeur exacte $\bar{a}_{\hat{\ell}}$ le test A aurait permis de transférer l'indice $\hat{\ell}$ de l'ensemble J_{L1} à l'ensemble J_1, le terme $a_{\hat{\ell}} y_{\hat{\ell}}$ aurait disparu de la contrainte de BENDERS et l'inégalité correspondante serait devenue irréalisable, ce qui conduisait à une régression dans l'exploration de l'arborescence.

Si l'on applique le test proposé, c'est-à-dire si l'on décide de fermer l'origine ou le transit $\hat{\ell}$, l'inégalité peut être réalisée en un noeud successeur du noeud γ.

Si l'on n'applique pas le test et que l'on continue l'exploration, le problème de la fixation de la variable $y_{\hat{\ell}}$ se posera et l'on sait que la seule valeur possible sera zéro. Donc le test conduit à anticiper cette décision mais en aucun cas ne conduit à une contradiction.

Si nous supposons que l'indice $\hat{\ell}$ appartient à l'ensemble J_{L2}, le raisonnement est le même, seule la conclusion est différente.

Enoncé du test C

Rendre les termes de l'inégalité positifs par le changement de variables $\bar{y}_\ell = 1 - y_\ell$; soit b' le second membre obtenu.
S'il existe un indice $\hat{\ell}$ tel que $-a_{\hat{\ell}} > b'$ alors :

- si $\hat{\ell} \in J_{L1}$
  transférer $\hat{\ell}$ de l'ensemble J_{L1} à l'ensemble J_0, c'est-à-dire faire $y_{\hat{\ell}} = 0$
- si $\hat{\ell} \in J_{L2}$
  transférer $\hat{\ell}$ de l'ensemble J_{L2} à l'ensemble J_1, c'est-à-dire faire $y_{\hat{\ell}} = 1$

III . ENSEMBLE DE VARIABLES PRÉFÉRÉES

C'est dans cet ensemble que l'on choisit la variable sur laquelle on effectue la séparation.
Rappelons que pour déterminer cet ensemble C.E. LEMKE et K. SPIELBERG [24] réduisent l'inégalité par *réduction complète*.

Soit :

$$(1) \qquad \sum_k a_k y_k \leq b$$

l'inégalité avec

$$a_k < 0 \qquad \forall k$$

$$b < 0$$

$$y_k \in \{0,1\} \quad \forall k$$

Réduire cette inégalité par réduction complète c'est faire une combinaison de cette inégalité avec les contraintes :

$$(2) \qquad 0 \leq y_k \leq 1 \quad \forall k$$

Cela revient à éliminer les variables y_k dans l'ordre des a_k décroissants tant que les a_k restent plus grands que le membre de droite.

Après réduction complète (1) s'écrit :

$$(3) \qquad \sum a_{k'} y_{k'} \leq b' \text{ avec } |a_{k'}| > b' \quad \forall k'$$

Cette inégalité sera réalisée si :

$$(4) \qquad \sum y_{k'} > 1$$

C'est parmi l'ensemble des indices k' que l'on choisit celui de la variable sur laquelle s'effectue la séparation. Il est inutile de faire des "pas-en-avant" sur d'autres variables ;

Exemple :

$$(1) \qquad -2y_1 - 6y_2 - 7y_3 - 4y_4 \leq -11$$

après réduction complète (1) s'écrit

$$(2) \qquad - 6y_2 - 7y_3 \leq -5$$

pour que l'inégalité (1) reste réalisable il faut que $y_2 = 1$ ou $y_3 = 1$.

Si l'une au moins de ces deux conditions n'est pas réalisée il est inutile de continuer l'exploration.

Traduisons ces résultats en variables y'.

Supposons que lors de l'établissement de la contrainte de BENDERS :

- $\ell \in J_{L1}$ alors $y_\ell \geq 1 \Longrightarrow y'_\ell = 0$
  c'est-à-dire que l'origine ou le transit temporairement ouvert doit être fermé

- $\ell \in J_{L2}$ alors $y_\ell \geq 1 \Longrightarrow y'_\ell = 1$
  c'est-à-dire que l'origine ou le transit temporairement fermé doit être ouvert.

D . OBTENTION D'UNE SOLUTION RÉALISABLE PAR UNE MÉTHODE HEURISTIQUE

Le choix optimal des implantations d'origines et de transits peut être fait en calculant les coûts de toutes les solutions possibles et en retenant la ou les meilleures.

Etant donné qu'en chaque origine et en chaque transit il existe deux solutions : ouvrir ou fermer le sommet correspondant, le nombre de solutions possibles est égal à 2^n où n est le nombre d'origines et de transits. La procédure heuristique proposée doit donc réduire considérablement le nombre de solutions à examiner tout en donnant une solution réalisable assez "voisine" de la solution optimale.

Nous allons d'abord envisager le cas du problème d'implantations de transits et ensuite nous passerons à celui d'implantations simultanées d'origines et de transits.

I - IMPLANTATIONS DE TRANSITS

Nous allons décrire la méthode dite *méthode d'élimination progressive* [20].

On suppose qu'au départ de la méthode un transit est *ouvert* en chaque site proposé.

Une itération se déroulera en deux temps :

1) fermer tout à tour un transit en laissant les autres ouverts.

2) *fermer définitivement* le transit dont la fermeture amène la plus grande économie.

Lorsqu'il ne sera plus possible de réaliser des économies en fermant un transit supplémentaire la solution "optimale" sera atteinte.
Nous n'avons aucune garantie que la solution obtenue soit optimale ; néanmoins la forme de la courbe de la figure (1) laisse supposer que la solution réalisable obtenue est assez voisine de la solution optimale.

L'évolution des coûts est donnée par la figure (1) lorsque le nombre de transits ouverts décroît :

- le coût de transport croît

- le coût fixe dû à l'ouverture des transits décroît

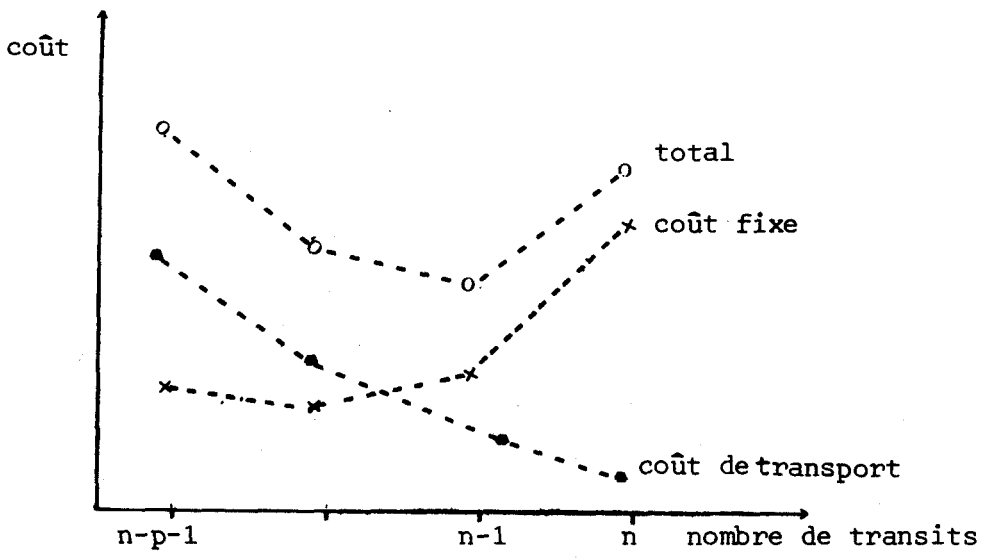


fig. 1

La figure (2) schématise le déroulement de la procédure ; en R tous les transits sont ouverts.

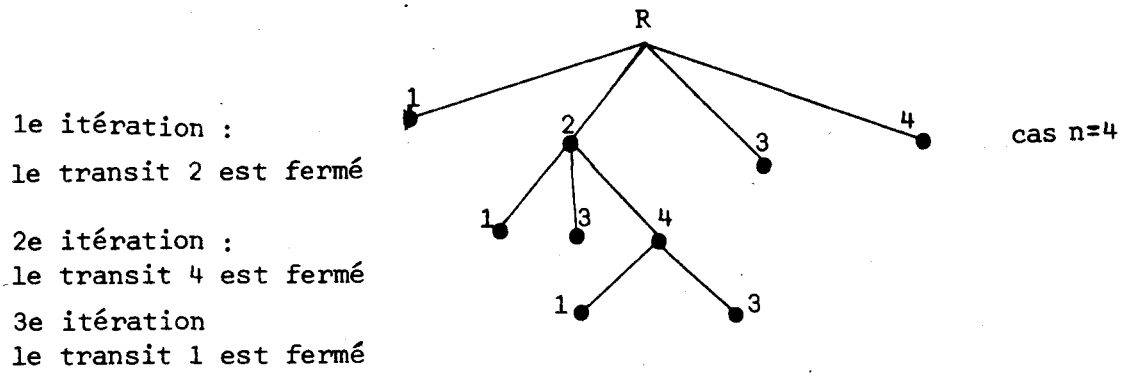


fig. 2

Il apparaît clairement que, si la solution "optimale" contient n-p transits ouverts, pour l'obtenir nous aurons résolu :

$$N = n(n+1) + (n-1)(n-2) + \ldots (n-p)\,(n-p-1)$$

programmes linéaires. Ce nombre est très inférieur à 2^n.

Exemple :

prenons n=20 et p=10 :

$$2^n = 2^{20} = 1048576$$

$$N = 2420$$

Remarquons que la *taille* de ces programmes *diminue* en fonction du nombre d'itérations.

Remarque

Cette procédure est un *algorithme glouton* défini par J. EDMONDS [9] : à chaque itération on fixe définitivement une variable booléenne.

Remarque

En langage "pénalités", à chaque itération, la variable booléenne retenue est celle dont l'indice j_p est déterminé par :

$$Z_{j_p} = Z + p^o_{j_p} - p^1_{j_p} = \min (Z + p^o_j - p^1_j : j \in J_L)$$

où J_L est l'ensemble des indices des vairables libres, et où Z est la valeur de la fonction économique de la solution retenue à l'étape précédente.

Si $Z_j < Z$ alors la variable booléenne d'indice j_p est mise à zéro, sinon la procédure est terminée.

Notons que l'on progresse dans l'arborescence en mettant à zéro la variable booléenne dont la mise à un correspond *au plus grand regret*.

Remarque

Nous verrons, au paragraphe B du chapitre suivant que l'on peut réduire le nombre d'itérations de la procédure précédente grâce aux renseignements fournis par la résolution du programme auxiliaire et par le *renforcement* que l'on peut y apporter éventuellement.

Remarque

L'utilisation de l'algorithme de Recherche d'une Solution Initiale (R.S.I.) diminue d'une manière considérable, de 40 à 60 %, le temps d'exécution de la procédure heuristique.

Remarque

On peut utiliser, à la place de la méthode décrite ci-dessus et appelée *méthode d'élimination progressive* la méthode dite "d'addition progressive".

Dans cette dernière méthode une itération se déroulera en deux temps :

1) ajouter tour à tour un transit

2) garder définitivement ouvert celui dont l'ouverture amène la plus grande économie./Lorsqu'il ne sera plus possible de réaliser des économies en ajoutant un transit supplémentaire la solution *optimale* sera atteinte. Cette méthode est préférable à la première décrite, lorsque l'on sait à priori qu'il y aura peu de transits ouverts à l'optimum.
Par la suite nous ne distinguerons pas ces deux méthodes.

II . IMPLANTATION SIMULTANÉES D'ORIGINES ET DE TRANSITS

Le problème est beaucoup plus "*contraint*" que le problème précédent ; en effet la fermeture de certaines origines conditionne l'existence de certains transits et vice versa.

Il est plus difficile dans ce cas de proposer une procédure heuristique donnant une "bonne" solution réalisable dans un laps de temps acceptable.

Notons que le nombre n d'origines est beaucoup plus petit que le nombre m de transits.

Il semble donc raisonnable de proposer l'heuristique suivante :

1) fermer tout à tour une origine et appliquer la procédure heuristique précédente pour éliminer les transits

2) fermer définitivement l'origine dont la fermeture réalise la plus grande économie.

Continuer jusqu'à ce qu'il ne soit plus possible de réaliser des économies.

Si l'on compte le nombre de programmes linéaires à résoudre il est évidemment plus faible que celui qui consisterait à calculer toutes les solutions possibles mais il peut être encore trop grand.

Nous basant sur les renseignements fournis par la résolution du programme auxiliaire PA_n et par son *renforcement* nous verrons qu'il est raisonnable de proposer la solution suivante :

- n'appliquer l'heuristique énoncée ci-dessus que pour des *origines déterminées*.

Cela réduit considérablement le temps de calcul et donne une "bonne" solution réalisable.

Remarque

Dans ce cas également nous utilisons l'algorithme R.S.I.

CHAPITRE V

A . MISE AU POINT DE L'ALGORITHME

A partir de plusieurs groupes de données définies par :

- le nombre d'origines
- le nombre de transits
- le nombre de destinations
- le nombre de liaisons : origines-transits
  transits-destinations

nous avons construit des graphes à l'aide de la procédure aléatoire P [8].

A chacun de ces graphes nousavons associé quelques problèmes d'optimisation ne différant entre eux que par les coûts fixes attachés à l'existence des origines et des transits ; les coûts fixes ont été tirés au hasard suivant la même procédure.

Les premiers problèmes d'optimisation portent sur quinze variables booléennes (4 origines, 11 transits) et les derniers sur une soixantaine (10 origines, 50 transits).

1 - Problème auxiliaire et son renforcement

a) Il est intéressant de noter que la valeur de la fonction économique du problème renforcé (quand ce renforcement existe évidemment) peut être de 6 à 18 % plus élevée que celle de la fonction économique du problème auxiliaire : d'où l'obtention d'un meilleur minorant de la fonction économique du sous-problème courant

b) Il est également intéressant de savoir que le temps de calcul de ce renforcement est faible devant celui correspondant à la résolution du programme auxiliaire ; il est de l'ordre de 1/15 à 1/20 par rapport au temps précédent.

2 - Détermination de l'état

Considérons le programme auxiliaire correspondant à la racine de l'arborescence, c'est-à-dire celui correspondant au cas où toutes les variables booléennes sont temporairement fixées à un. Dès le début des expériences numériques nous nous sommes intéressés à des quantités paraissant susceptibles de donner des indications sur la solution optimale :

$$- a_i = (\sum_j q_i \hat{x}_{ij}) / f_i \qquad \text{et } a_j = (\sum q_j \hat{x}_{jk}) / f_j$$

$$- b_i = (\sum_j \hat{x}_{ij}) / (\bar{P}_i - \underline{P}_i) \qquad \text{et } b_j = (\sum \hat{x}_{jk}) / (\bar{Q}_j - \underline{Q}_j)$$

$$\text{ou } b_i = (\sum_j \hat{x}_{ij}) / P_i \qquad \text{et } b_j = (\sum \hat{x}_{jk}) / Q_j$$

$$- c_i = f_i / \sum_j \hat{x}_{ij} \qquad \text{et } c_j = f_j / \sum \hat{x}_{jk}$$

Parmi ces quantités ce sont les *premières* que nous avons retenues.

Appelons :

X_1 : l'ensemble des indices des origines

X_2 : l'ensemble des indices des transits

I_{opt} : l'ensemble des indices des origines ouvertes à l'optimum

J_{opt} : l'ensemble des indices des transits ouverts à l'optimum

Nous avons remarqué qu'en *moyenne* les pourcentages a_i, $i \in I_{opt}$, figurent parmi les plus élevés des pourcentages a_i, $i \in X_1$. La même remarque est valable pour les transits ouverts à l'optimum.

Comme nous l'avons déjà dit le problème "implantations simultanées d'origines et de transits" est très *contraint* en ce sens que la fermeture d'une ou plusieurs origines peut entraîner la fermeture d'un ou plusieurs transits et vice versa.

Le problème "implantations de transits" est moins contraint : il est alors intéressant de noter que la résolution du programme auxiliaire où toutes les variables sont temporairement fixées à un permet de tirer des renseignements plus précis quant aux transits ouverts à l'optimum : les a_j correspondants à ces transits sont *parmi* les plus élevés.

De préférence aux quantités a_i et a_j nous avons retenues celles données par le programme auxiliaire *renforcé* :

$$a'_i = (\sum_j {}'_{ij}x_{ij}) / f_i \quad \text{et} \quad a'_j = (\sum_k {}'_{jk}x_{jk}) / f_j$$

En effet, quand il existe, le *renforcement* joue dans le "bon sens", c'est-à-dire qu'il confirme toujours les conclusions tirées des quantités a_i et a_j et souvent renforce certaines de ces quantités, ce qui contribue à rendre plus efficaces les décisions adoptées.

Exemples

Dans la présentation des exemples nous adoptons les conventions suivantes :

- si n est le nombre d'origines et m le nombre de transits : les origines sont numérotées de 1 à n et les transits de n+1 à n+m
- un numéro accompagné d'une astérisque indique que l'origine ou le transit correspondant est ouvert à l'optimum.

Premier exemple

6 origines, 20 transits, 967 variables continues ; les coûts fixes des origines sont dix foix plus élevés que ceux des transits.

Solution optimale $z^* = 4021$

2*, 7*, 8*, 16*, 17*, 23*, 24*, 25*

Le premier programme auxiliaire a été résolu en considérant toutes les variables comme libres ; du *renforcement* de ce problème nous avons tirés les a'_i et le a'_j suivants :

0.10	0.41*	0.18	0.	0.	0.30	} origines
0.32*	0.73*	0.	0.23	0.	0.04	
0.	0.	0.	0.92*	0.13*	0.09	} transits
0.	0.32	0.05	0.34	0.33*	0.24*	
0.37*	0.					

Le *renforcement* de la fonction économique du programme auxiliaire est de 6 %.

Dès que l'origine 2 a été fixée définitivement ouverte, on a résolu le programme auxiliaire correspondant en laissant toutes les autres variables libres (origines et transits) voici les a_i' et a_j' :

0.	×	0.	0.	0.	0.
0.27*	0.77*	0.	0.	0.	0.
0.	0.22	0.	0.98*	0.71*	0.22
0.	0.	0.	0.9	0.52*	0.35*
0.75*	0.				

Les indications sont plus précises quant aux transits ouverts à l'optimum.

× signifie que la variable y_2' est fixée.

Deuxième exemple

6 origines, 20 transits, 427 variables continues ; les coûts fixes des origines sont du même ordre de grandeur que ceux des transits.

solution optimale : valeur de la fonction économique
$z^* = 5793$
: 10*, 13*, 14*, 16*, 20*, 22*, 23*, 24*, 25*

voici les a_i et les a_j tirés de la résolution du programme auxiliaire où toutes les variables sont libres.

0.	0.20	0.35*	0.20	0.02	0.19	origines
0.04	0.18	0.19	0.34*	0.	0.06	transits
0.11*	0.10*	0.14	0.27*	0.	0.	
0.31	0.27*	0.26	0.35*	0.14*	0.34*	
0.20*	0.					

voici les a_i' et les a_j' tirés du *renforcement* du programme auxiliaire

0.	0.22	0.59*	0.24	0.02	0.19
0.04	0.37	0.20	0.63*	0.	0.11
0.12*	0.11*	0.24	0.64*	0.	0.
0.84	0.62*	0.50	0.71*	0.17*	0.93*
0.70*	0.				

renforcement de la fonction économique : 15 %.

Dans la pratique *deux points* sont à préciser :

1) Quels sont les seuils τ_i et τ_j permettant de déterminer l'état ?

$$y_\ell = 1 \text{ si } a'_\ell \geq \tau_\ell \qquad (\ell = i \text{ ou } j)$$
$$= 0 \text{ sinon}$$

2) Faudra-t-il procéder à des *réajustements* d'état en cours d'algorithme ?

Détermination des seuils τ_i et τ_j

En nous basant sur les exemples traités que nous avons voulu *variés* et *aussi proches que possible de la réalité*, nous proposons de prendre :

$$\tau_i = 25 \text{ à } 30 \ \%$$
$$\tau_j = 15 \text{ à } 20 \ \%$$

Réajustement d'état

Il est inutile de résoudre le programme auxiliaire à chaque noeud de l'arborescence ; nous proposons cette résolution que lorsque l'on apporte des modifications aux variables booléennes des origines. Cela découle des remarques précédentes.

3 - Procédure heuristique

a- Implantation de transits

Il est intéressant de noter que l'élimination des transits se fait à chaque itération en ordre "croissant" (à moins de 2 % près) des quantités $a_j^!$, les premiers éliminés étant ceux dont les $a_j^!$ sont nuls.

Compte-tenu que cette procédure a toujours donné la solution optimale, on peut en utiliser les résultats pour déterminer l'état : on prend pour le seuil τ_j la plus petite quantité $a_i^!$ correspondant aux indices des transits ouverts de la solution "optimale" proposée.

Implantations simultanées d'origines et de transits

Si l'on applique la méthode proposée au chapitre IV paragraphe D page

1) fermer tour à tour une origine et appliquer l'heuristique de fermeture de transits

2) fermer définitivement l'origine dont la fermeture entraîne la plus forte économie

3) continuer jusqu'à ce qu'il ne soit plus possible de réaliser des économies

On s'aperçoit que l'élimination des origines se fait en ordre croissant des quantités $a_i^!$, les premières éliminées étant celles dont les $a_i^!$ sont nuls.

Nous proposons donc de n'appliquer la méthode précédente qu'aux origines dont les $a_i^!$ sont supérieurs au seuil τ_i.

Nous avons toujours obtenu la solution optimale.

Remarque

La connaissance d'une "bonne solution" réalisable diminue le temps de calcul de l'algorithme général proposé au paragraphe suivant.

4 - Contraintes de BENDERS

Calcul des évaluations

Ce calculs est de l'ordre de 1/10 à 1/20 du temps nécessaire à la résolution du programme d'état correspondant ; il dépend relativement peu du nombre de sommets temporairement ouverts ou fermés : cela tient au fait que l'on travaille toujours sur le graphe initial.

Les évaluations sont d'autant meilleures que l'on avance dans l'arborescence ce qui contribue à rendre les inégalités de BENDERS de plus en plus efficaces.

5 - Choix de la variable sur laquelle s'opère la séparation

Ce choix est d'un intérêt primordial dans l'efficacité de l'algorithme.

Nous avons tenu compte dans ce choix

- du calcul des évaluations que nous avons voulues précises
- des quantités $a_i^!$ et $a_j^!$ qui nous semblent assez représentatives de la solution réalisable que l'on tente d'atteindre.

Nous proposons donc
de choisir la variable à séparer dans l'*ensemble de variables préférées* suivant les critères :

1 - **critère de priorité**
donner la préférence à la variable dont l'indice est celui d'une origine

2 - **critère d'égalité**
à égalité de priorité donner la préférence à la variable dont le $a_i^!$ ou le $a_j^!$ est le plus élevé.

Remarque 1

Il est entendu que les quantités $a_i^!$ et $a_j^!$ dont il est question précédemment sont les dernières calculées à la suite d'un précédent changement dans la valeur d'une variable booléenne d'une origine.

Remarque 2

Comme critère d'égalité nous aurions pu retenir le suivant :
à égalité de priorité donner la préférence à la variable ℓ pour laquelle *la quantité* $|f_\ell - ev_\ell|$ *est la plus grande* avec :

- ℓ = i ou j
- $ev_\ell = ev_\ell^o$ ou ev_ℓ^1 suivant l'appartenance de ℓ à l'ensemble J_{L1} ou à l'ensemble J_{L2}

Nous avons préféré retenir le premier critère d'égalité énoncé par ce qu'il nous semble raisonnable de ne pas tout faire reposer sur la détermination de l'état ; or si cette détermination se fait effectivement à partir des quantités $a_i^!$ et $a_j^!$ il n'en reste pas moins vrai que le seuil est déterminé d'une façon plus ou moins empirique et que cette détermination a une influence sur les évaluations.

Il est intéressant de noter que très souvent, surtout en cours d'algorithme, les deux critères conduisent au même choix ; néanmoins l'application systèmatique du premier est préférable à celle du second.

Remarque 3

La séparation ne s'effectuera jamais :

- sur une variable correspondant à une origine desservie par des transits déjà fermés
- sur une variable correspondant à un transit alimenté par des origines déjà fermées.

En effet dans la détermination de l'état, la variable correspondante sera mise temporairement à zéro car la quantité $a_i^!$ sera nulle.

L'évaluation correspondante sera nulle également et dès le premier test A d'exploitation des inégalités la variable booléenne sera fixée à zéro : la variable y correspondante ne figurera donc plus dans l'inégalité de BENDERS correspondante.

B . ALGORITHME DE RÉSOLUTION

ℓ : indice correspondant soit à une origine, soit à un transit

n : niveau d'un noeud courant, c'est le nombre de variables fixées par choix

M : tableau unidimensionnel

MEM : tableau bidimensionnel

(x^*, y^*) solution optimale et z^* valeur correspondante de la fonction économique

(x_h, y_h) solution "optimale" donnée par la procédure heuristique et z_h la valeur correspondante de la fonction économique

J_1 : ensemble des indices des variables fixées à un

J_0 : ensemble des indices des variables fixées à zéro

J_{L1} : ensemble des indices des variables temporairement fixées à un

J_{L2} : ensemble des variables temporairement fixées à zéro.

ALGORITHME

a) Entrée des données.

b) Initialisation et phase préliminaire

- si l'on connaît une solution "optimale" heuristique faire $z^* = z_h$ sinon prendre z^* arbitrairement grand

- ouvrir temporairement toutes les origines et tous les transits :

- $M = \emptyset$

- $n = 0$

- résoudre le problème auxiliaire correspondant
déterminer l'état
résoudre le programme d'état correspondant : soit z la valeur optimale de la fonction économique
si $z < z^*$ faire $z^* = z$ et retenir la solution optimale
calculer les évaluations
écrire la contrainte de BENDERS
appliquer les tests A, B, C et construire l'ensemble des variables préférées
choisir la variable sur laquelle va s'effectuer la première séparation ; soit ℓ^* son indice
transférer ℓ^* de J_{L1} vers J_1 ou J_0 et empiler ℓ^* dans M.
libérer toutes les autres variables en transférant leurs indices dans J_{L1}
aller en c).

c) 1 - résoudre le programme auxiliaire PA_n et calculer son renforcement $\underline{z}$
si $\underline{z} > z^*$ aller en h) sinon calculer les quantités $a^!_\ell \quad \forall \ell \in J_{L1}$

2 - déterminer l'état et transférer les indices convenables de J_{L1} vers J_{L2}

d) résoudre le programme d'état PE_n, soit z_n sa fonction économique
si $z_n < z^*$ faire $z^* = z_n$, retenir la solution
calculer les évaluations ; rentrer l'inégalité de BENDERS dans MEM_n
appliquer le test A : si une ou plusieurs variables sont fixées par ce test, empiler les indices correspondants dans M en faisant précéder chaque indice d'un signe moins ; faire les transferts d'indices des ensembles J_{L1} et J_{L2} vers les ensembles J_1 et J_0 ;

modifier l'inégalité de BENDERS en conséquence ;
appliquer le test B : si le test joue aller en h)
appliquer le test C : si le test joue empiler les indices dans la mémoire M en les faisant précéder d'un signe moins, faire les transferts d'indices vers les ensembles J_1 et J_0 et modifier l'inégalité de BENDERS en conséquence.
si toutes les variables sont fixées aller en f)

e) choix
construire l'ensemble de variables préférées ; choisir ℓ^* suivant les critères de choix précédemment décrits, faire passer dans l'ensemble convenable J_1 ou J_0
empiler l'indice ℓ^* dans la mémoire M ;
faire n = n+1 ;
reprendre les dernières inégalités de BENDERS écrites pour tenter de fixer de nouvelles variables ; si cela est possible empiler les indices correspondants dans la mémoire M en les faisant précéder d'un signe moins, faire les transferts d'indices, si toutes les variables sont fixées aller en f)
si une variable relative à une origine a été modifée depuis la dernière résolution du programme auxiliaire aller en c) 1
sinon aller en c) 2

f) résolution
résoudre le programme correspondant, soit z sa fonction économique ; si $z > z^*$ aller en h)
si $z \leqslant z^*$ faire $z^* = z$ et retenir la solution optimale, aller en h)

h) régression
dépiler la mémoire M jusqu'à rencontrer un indice sans signe en libérant les variables et en faisant passer les indices correspondants de J_1 et J_0 vers J_{L1}
si l'on ne rencontre pas d'indice ℓ sans signe la procédure est terminée, sinon changer ℓ en $-\ell$ dans la mémoire M, faire passer l'indice ℓ de l'ensemble J_1 vers J_0 ou de J_0 vers J_1

effacer la dernière inégalité ; faire n = n-1 ; si aucune variable d'origine n'a été modifiée depuis la résolution du dernier programme auxiliaire résolu aller en c) 2 sinon aller en c) 1.

Remarque 1

Lorsque, depuis la dernière résolution du programme d'état, les seuls transferts d'indices se sont effectués de J_{L1} vers J_1 ou de J_{L2} vers J_0 il n'est pas nécessaire de résoudre le programme d'état de la phase d) : la nouvelle fonction économique se déduit immédiatement de la précédente.

Remarque 2

Dans l'étape e) il est inutile de reprendre plus de quatre ou cinq inégalités ; effectivement les évaluations devenant de plus en plus précises au fur et à mesure que l'on avance dans l'exploration de l'arborescence les dernières inégalités écrites sont les plus efficaces.

Remarque 3

Il est intéressant de remarquer que l'algorithme donne des solutions réalisables *très voisines* de la solution optimale ; ceci provient du fait que nous utilisons des évaluations et que celles-ci ne permettent pas toujours d'arrêter l'exploration assez rapidement. Ceci n'est pas un désavantage puisqu'il peut être intéressant de connaître ces solutions qui peuvent être retenues ultérieurement. Ces solutions très proches de la solution optimale ne sont pas toujours données par la procédure heuristique.

Remarque 4

La solution optimale du programme d'état résolu dans la phase b) donne une *bonne* solution réalisable de départ pour z*, dans le cas où l'on ne connaît pas de solution heuristique.

C . EXEMPLES D'APPLICATION ET RÉSULTATS NUMÉRIQUES

Exemple 1

Convention

caractérisation de l'état d'une variable

$\underline{1}$ variable fixée à 1

$\underline{0}$ variable fixée à 0

1 variable temporairement fixée à 1

0 variable temporairement fixée à 0

Rappel de notations

$\underline{z}$ valeur optimale de la fonction économique du programme auxiliaire PA_n

$\underline{\underline{z}}$ valeur optimale de la fonction économique du programme auxiliaire renforcé PAR_n

z valeur optimale de la fonction économique du programme d'état PE_n

Données

4 origines

10 transits

les coûts fixes des transits sont du même ordre de grandeur que ceux des origines

solution heuristique $z_h = 4536$

1*, 4*, 10*, 14*

seuil $\tau_i = \tau_j = 0.20$.

Indices des variables	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
z^h=4536															
z*=4536															
état	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
$\underline{z}$=2536															
$\underline{\underline{z}}$=2680															
$a_i^!$ et $a_j^!$	0.35	0.	0.	0.65	0.	0.	0.11	0.	0.	0.27	0.28	0.	0.28	0.26	
état	1	0	0	1	0	0	0	0	0	1	1	0	1	1	
z=5695															
inégalité	-1000	1032	1500	-1050	926	990	940	810	793	-542	-671	920	-674	-576	-1159
test A	1	0	0	1	0	0	0	0	0	1	1	0	1	1	
test B	ne joue pas														
test C	ne joue pas														
variables préférées	×			×							×		×		
choix	$y_4^!$ = 1														
MP =	[4]														
état	1	0	0	1	0	0	0	0	0	1	1	0	1	1	
z=5695															
inégalité	-1000	1032	1500		926	990	940	810	793	-542	-671	920	-674	-576	-1159

test A	1	0	0	1	0	0	0	0	0	1	1	0	1	1		
test B	ne joue pas															
test C	ne joue pas															
variables préférées	×										×		×			
choix	$y'_1 = 1$															
MP	[4, -2, -3, -5, -6, -7, -8, -9, -12, 1]															
z=4250																
z=4350																
a'_i et a'_j										0.50	0.50		0.48	0.45		
état	1	0	0	1	0	0	0	0	0	1	1	0	1	1		
z=5695																
inégalité										-542	-671		-674	-576	-1159	
test A	ne joue pas															
test B	ne joue pas															
test C	ne joue pas															
variables préférées											×		×			
choix	$y'_{11} = 1$															

Indices des variables	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
	ce choix entraîne la fixation de y'_{13} à 0														
MP =	[4, -2, -3, -5, -6, -7, -8, -9, -12, 1, 11, -13]														
état	1	0	0	1	0	0	0	0	0	1	1	0	0	1	
z=5106															
inégalité										-436				-496	-570
test A	ne joue pas														
test B	ne joue pas														
test C	joue														
MP	[4, -2, -3, -5, -6, -7, -8, -9, -12, 1, 11, -13, -10, -14]														
état	1	0	0	1	0	0	0	0	0	1	0	0	1	1	
	toutes les variables sont fixées mais problème impossible														
régression	$y'_{11} = 0$														
MP	[4, -2, -3, -5, -6, -7, -8, -9, -12, 1, -11]														
état	1	0	0	1	0	0	0	0	0	1	0	0	1	1	
z=5076															
inégalité										-424			-556	-496	-540

test A	ne joue pas														
test B	ne joue pas														
test C	ne joue pas														
variables préférées														×	×
choix	$y'_{13} = 1$													×	
	ce choix entraîne la fixation de y'_{10} et de y'_{14} à 0														
MP	[4, -2, -3, -5, -6, -7, -8, -9, -12, 1, -11, 13, -10, -14]														
	toutes les variables sont fixées mais problème impossible														
régression	$y'_{13} = 0$														
MP	[4, -2, -3, -5, -6, -7, -8, -9, -12, 1, -11, -13]														
état	1	0	0	1	0	0	0	0	0	1	0	0	0		1
z=4536	la fermeture des transits 10 ou 14 entraîne problèmes correspondants impossibles														
z=4536	1	0	0	1	0	0	0	0	0	1	0	0	0		1
	est solution optimale														
régression	$y'_1 = 0$														
MP	[4, -2, -3, -5, -6, -7, -8, -9, -12, -1]														
z=4778															
z=4778															
régression	$y'_4 = 0$														
MP	[-4]														

$\underline{z}$=2888															
$\underline{z}$=3031															
a'_i et a'_j	1	0	0.		0.	0.	0.30	0.33	0.	0.22	0.	0.	0.16	0.33	
état	1	0	0	0	0	0	1	1	0	1	0	0	0	1	
z=4947	-844	1032	1032		926	926	-432	-620	840	-528	0	886	682	-551	-411
test A	1	0	0	0	0	0	1	1	0	1	0	0	0	1	
test B	ne joue pas														
test C	ne joue pas														
MP	[-4, -2, -3, -5, -6, -9, -11, -12, -13]														
variables préférées	×						×	×		×				×	
choix	$y'_1 = 1$														
MP	[-4, -2, -3, -5, -6, -9, -11, -12, -13, 1]														
$\underline{z}$=4865															
$\underline{z}$=4835															
régression	$y'_1 = 0$														
MP	[-4, -2, -3, -5, -6, -9, -11, -12, -13, -1]														
	problème impossible														
régression															
MP	[Ø]														
	procédure terminée														

Exemple 2

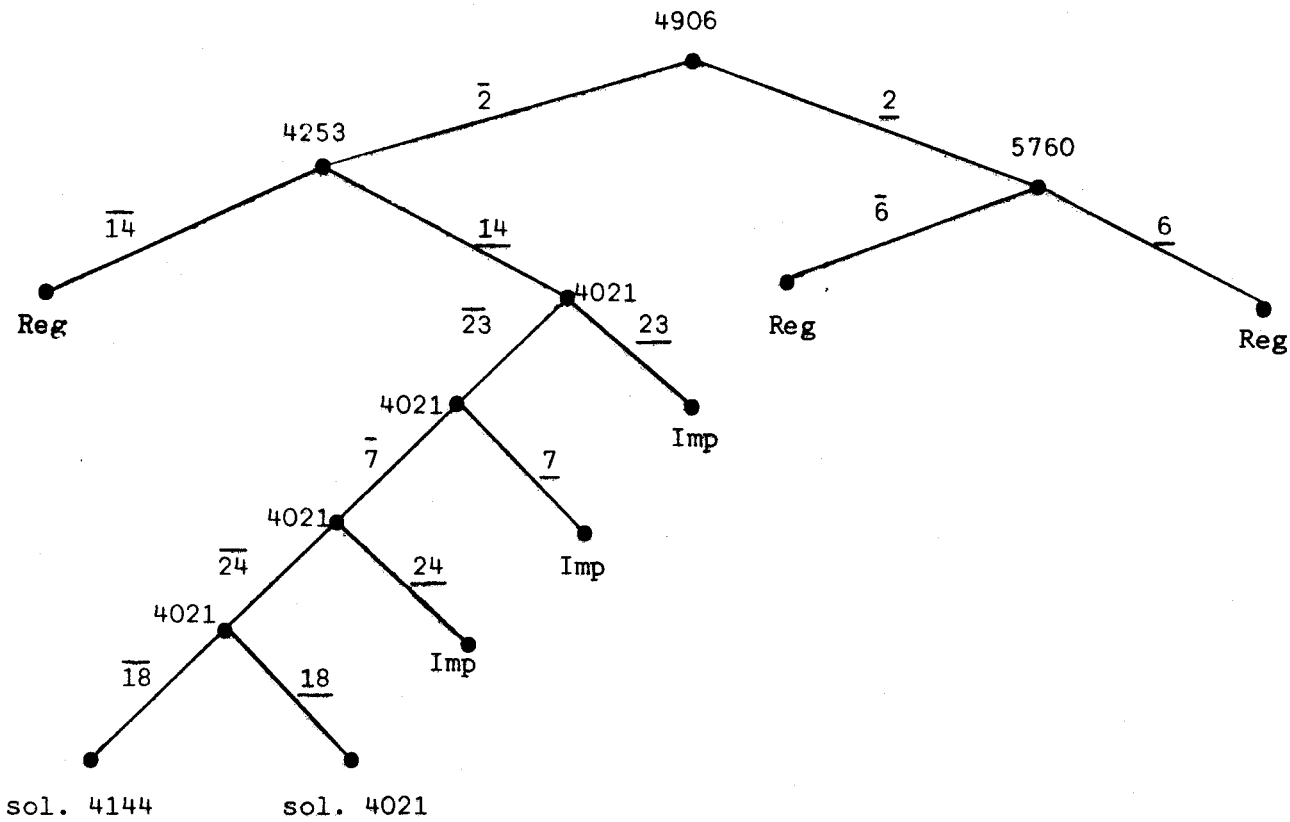
Données

6 origines

25 transits

les coûts fixes des transits sont dix foix plus petits que ceux des origines

1 - Arborescence décrite par l'algorithme en partant de $z^* = z_h = 4021$



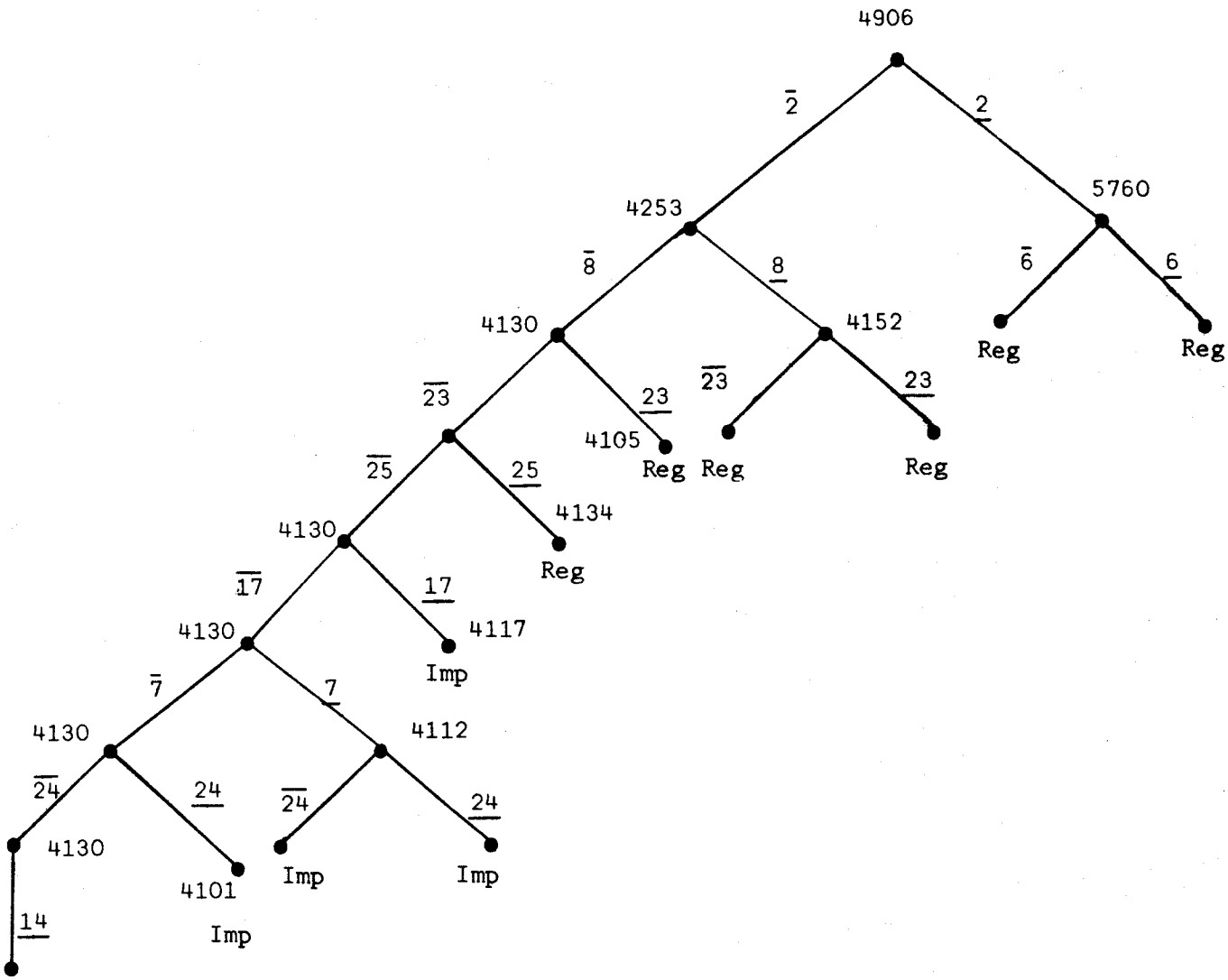
convention: ex. $\bar{4}$ origine 4 ouverte

$\underline{4}$ origine 4 fermée

Les nombres inscrits a côté de chaque sommet de l'arborescence donne la valeur de la fonction économique du problème d'état PE_n correspondant.

2 - Arborescence décrite lorsque l'on part de z* arbitrairement grand

Dès le début z* = 4906



sol. 4021

Gain de temps apporté par la connaissance de la solution heuristique : 67 %.

Procédure heuristique

Les $a_i^!$ de départ étaient : 0.10 0.41 0.18 0. 0. 0.30
nous avons appliqué la procédure en supposant seulement l'ouverture de l'origine 2.

Compte-tenu du temps d'obtention de la solution heuristique, l'algorithme commencé avec cette solution est de durée comparable à celle de l'algorithme commencé avec une valeur arbitrairement grande.

Notons que, si nous avions appliqué la procédure heuristique en supposant que l'origine 6 puisse être ouverte également, alors le temps de calcul d'obtention de la solution heuristique aurait été beaucoup plus important

Résultats numériques

(voir tableau)

- densité du graphe : nombre d'arcs existants sur nombre d'arcs possibles

- nombre de clients : environ cinquante.

- **les gains sont calculés sans tenir compte du temps nécessaire à l'obtention de la solution heuristique.**

nombre origines	nombre transits	densité graphe	valeurs de départ	gain	nombre de solutions atteintes
4	10	100 %	z_h	60 %	1
			arb. grande		2
6	25	100 %	z_h	67 %	3
			arb. grande		10
6	25	46 %	z_h	54 %	2
			arb. grande		1
8	25	36 %	z_h	52 %	2
			arb. grande		4
8	25	100 %	z_h	50 %	2
			arb. grande		5
10	35	37 %	z_h	35 %	2
			arb. grande		5
10	40	40 %	z_h	48 %	3
			arb. grande		8
10	40	32 %	z_h	45 %	2
			arb. grande		6
12	50	35 %	z_h	60 %	5
			arb. grande		10

Nous présentons quelques résultats numériques ; ceux-ci nous paraissant encourageants, nous avons l'intention de poursuivre une étude systématique sur des exemples plus importants et plus variés ; nous pensons également traiter quelques exemples concrets.

Des premiers résultats il découle que la donnée d'une bonne solution heuristique raccourcit le temps du déroulement de l'algorithme ; mais il faut préciser que cette solution est parfois coûteuse à atteindre ; les quelques exemples traités semblent prouver qu'il est plus économique de prendre comme valeur de départ celle donnée dans la phase préliminaire de l'algorithme.

BIBLIOGRAPHIE

[1] BALAS (E)
"An additive algorithm for solvieng linear programs with zero-one variables"
Op. Res. 13, 4, 1965.

[2] BALAS (E)
"Discrete programming by the filter method"
Op. Res. 15, 5, 1967.

[3] BALINSKI (M.L) et SPIELBERG (K)
"Methods for Integer Programming : Algebraic, Combinatorial and Enumerative"
Progress in Operations Research, Vol. 3, editor J. Aronofsky, John Wiley and Sons, 1969.

[4] BARR (R.S), GLOVER (F) et KLINGMAN (D)
"An Improved Version of the out-kilter Method and comparitive study of computer codes"
Management Science Report Series, University of Colorado, Report n° 72-15, nov. 72.

[5] BENDERS (J.F)
"Partitioning Procedures for Solving Mixed Variables programming Problems"
Numerische Mathematik, Vol. 4, 1962, p. 238-252.

[6] BERGE (C)
"Graphes et hypergraphes"
Dunod, 1970.

[7] BERTIER (P) et ROY (B)
"Procédure de résolution pour une classe de problèmes pouvant avoir un caractère combinatoire"
Cahiers du Centre d'Etudes de Recherche Opérationnelle, 6, 1964, p. 202-208.

[8] COQUET (M) et DENEL (J)
"Approximation des polyèdres de R^n"
Rapport de D.E.A, Laboratoire de Calcul de Lille, juin 1969.

[9] EDMONDS (J)
"Matroids and the greedy algorithm"
Mathematical programming 1 (1971), p. 127-136.

[10] FULKERSON (D.R)
"An out-of-kilter Method for Minimal Cost Flow Problems"
J. Soc. Indust. Appl. Math. Vol 9, 1961, p. 18-27.

[11] GEOFFRION (A.M)
"Integer Programming by Implicit Enumeration and Balas Method"
SIAM Review, n°9, 1967, p. 178-190.

[12] GEOFFRION (A.M)
"Elements of large-scale Mathematical Programming, Part-1 : concepts"
Na, Sc., V. 16, n°11, 1970, p. 652-675.

[13] GLOVER (F)
"A multiphase-dual algorithm for the zero-one integer programming problem"
Op. Res. 13, 6, 1965.

[14] GOMORY (R.E)
"An Algorithm for Integer Solutions to Linear Programs"
Recent Advances in Mathematical Programming, eds. R.L. Graves, P. Wolfe, Mc Graw-Hill, New-York, 1963.

[15] GOMORY (R.E) et JOHNSON (E.L)
"Some Continuous Functions related to Corner Polyhedra and their Application to Integer Programming"
Mathematical Programming 3, (1972), p. 23-85.

[16] GUIGNARD (M) et SPIELBERG (K)
"Search Techniques with Adaptative Features for certain Integer and Mixed Integer Programming Problems"
Proceedings, IFIPS Congress, 1968.

[17] GUIGNARD (M) et SPIELBERG (K)
"The State Enumeration Method for Mixed-Zero-One Programming"
IBM Technical Report n° 320-3000, Philadelphia Center, 1971.

[18] HANSEN (P)
"Programmes Mathématiques en variables 0-1"
Dissertation présentée à l'Université Libre de Bruxelles, 1974.

[19] HAYEZ (J.M)
"Procédure UCTIME"
Centre Interuniversitaire du Traitement de l'Information (C.I.T.I), Lille.

[20] HENRY (G)
"Recherche d'un réseau de dépôts optimum"
R.I.R.O, 1968, n°11, p. 61-70.

[21] HUARD (P)
"Programmes Mathématiques non Linéaires à Variables Bivalentes"
6 th Symposium on Mathematical Programming, Princeton, N.J, 1967.

[22] JOHNSON (E)
"Programming in networks and graphe"
O.R.C 1965.

[23] LAND (A.H) et DOIG (A.G)
"An automatic Method for Solving Discrete Programming Problems"
Econometrica, 28, 1960, p. 497-520.

[24] LEMKE (C.E) et SPIELBERG (K)
"Direct Search Algorithms for zero-one and Mixed Integer Programming"
Op. Res. Vol. 15, n°5, 1967, p. 892-914.

[25] LITTLE (J.D.C), MURTY (K.G), SWEENY (D.W), KAREL (C)
"An Algorithm for the Travelling Salesman Problem"
Op. Res. 11, 1963, p. 972-989.

[26] ROY (B)
"Procédures d'exploration par séparation et évaluation (P.S.E.P et P.S.E.S)"
Revue Française d'Informatique et de Recherche Opérationnelle, V-1, 1969, p. 61-90.

[27] ROY (B) et BENAYOUN (R)
"Programmes linéaires en variables bivalentes et continues sur un graphe"
METRA, V.6, n°4, p. 675,710.

[28] SPIELBERG (K)
"On solving plant Location Problems"
NATO Conference on Applications of Mathematical Programming Techniques, Cambridge, Grande-Bretagne, 1968.

[29] WERTHEIMER (D)
"Recherche d'une solution initiale pour l'algorithme "out-of-kilter" de FULKERSON appliqué à un graphe tri-parti"
Pub. Lab. de calcul (1974) Université de Lille.