Titre: Title:	Problème du voyageur de commerce : plus court chemin et optimisation par sous-gradient
Auteurs: Authors:	Jean-Claude Picard, & Maurice Queyranne
Date:	1976
Type:	Rapport / Report
Référence: Citation:	Picard, J.-C., & Queyranne, M. (1976). Problème du voyageur de commerce : plus court chemin et optimisation par sous-gradient. (Rapport technique n° EP-R-76-07). https://publications.polymtl.ca/6081/

Document publié chez l'éditeur officiel
Document issued by the official publisher

Institution:	École Polytechnique de Montréal
Numéro de rapport: Report number:	EP-R-76-07
URL officiel: Official URL:	
Mention légale: Legal notice:	

DÉPARTEMENT DE GÉNIE INDUSTRIEL

RAPPORT TECHNIQUE: EP76 - R - 7
Classification: Library of Congress

LE PROBLEME DU VOYAGEUR DE COMMERCE: PLUS COURT CHEMIN ET OPTIMISATION PAR SOUS-GRADIENT

PAR: Jean-Claude Picard, Ecole Polytechnique
Maurice Queyranne, Ecole Polytechnique

Le 18 mars 1976

Ecole Polytechnique de Montréal

C.P. 501
Snowdon
Montréal 248

LE PROBLEME DU VOYAGEUR DE COMMERCE: PLUS COURT CHEMIN ET OPTIMISATION PAR SOUS-GRADIENT

par

Jean-Claude Picard
Ecole Polytechnique, Montréal

et

Maurice Queyranne
Ecole Polytechnique, Montréal

*Cette recherche a été subventionnée par le fonds de recherche CNRC A8528.

R E S U M E

Dans un réseau à n+1 sommets, un O-circuit est un chemin d'origine x_o fixée, devant comporter n+1 arcs, et d'extrêmité x_o; une formulation par un programme linéaire en nombres entiers, à "n" contraintes du problème du voyageur de commerce est définie; une méthode par sous-gradient permet d'obtenir efficacement une borne inférieure sur la longueur du tour optimal; un algorithme par "Branch and Bound" en est dérivé, et donne une solution exacte du problème du voyageur de commerce.

INTRODUCTION

Le Problème du Voyageur de Commerce (TSP), qui consiste à trouver le plus court circuit qui traverse exactement une fois chaque sommet d'un réseau, a suscité de nombreuses recherches dans le domaine de la Recherche Opérationnelle (par exemple, voir [3], [12]).

Dans un rapport précédent [26], a été définie une approche à ce problème, caractérisée essentiellement par l'usage des trois techniques emboîtées suivantes:

(i) recherche d'un O-circuit de longueur minimale

(ii) introduction de pénalités dans les distances, et optimisation par Programmation Linéaire

(iii) Enumération Implicite.

Le présent rapport reprend et modifie cette approche. Le premier chapitre concerne le point (i); les O-circuits y sont définis, et interprétés sur un réseau multiparti associé et l'algorithme de Saigal est rappelé. Un second chapitre reprend succintement les propriétés du programme linéaire du point (ii) développées dans le précédent rapport. Le troisième chapitre est concerné par une autre méthode, dite de sous-gradient, pour réaliser l'optimisation des pénalités au point (ii); cette méthode est comparée sur un exemple avec l'algorithme du simplexe par génération de colonne utilisé précédemment. Le chapitre 4 décrit un algorithme par Enumération Implicite utilisant l'optimisation par sous-gradient et un exemple

est résolu par cet algorithme. Enfin dans le dernier chapitre sont discutées différentes extensions de cette approche, à des problèmes généraux de tournées incluant éventuellement plusieurs véhicules avec capacité, et des demandes multiples. Trois annexes décrivent des procédés mis au point pour réduire l'effort de calcul des plus courts chemins; l'annexe 1 indique un procédé ("pseudo-couplage") fournissant de bonnes pénalités de départ; l'annexe 2 décrit les "0-circuits sans va-et-vient" et un algorithme correspondant; ceci permet d'obtenir une meilleure borne au prix d'une faible augmentation de temps de calcul; enfin l'annexe 3 indique comment l'utilisation d'un graphe restreint permet de réduire considérablement le temps de calcul dans la plus grande partie des itérations. La mise en oeuvre et l'expérimentation des méthodes décrites dans ce rapport sont actuellement entreprises afin d'en vérifier l'efficacité.

1. 0-Circuits

1.1 Définitions

Soit R = (V, E, d) un réseau à V = n + 1 sommets, où l'arc (x,y)εE est muni d'une valuation d(x,y) représentant la "longueur" de l'arc (x,y). Nous supposerons ce réseau complet (on peut affecter une longueur infinie aux arcs inexistants) et sans boucle. Désignons par x_o un sommet quelconque de R et les autres sommets par $x_1,\ldots x_n$; dans toute la suite, nous noterons $d_{ij} = d(x_i,x_j)$ $(i \neq j)$.

Nous emploierons le mot tour pour désigner un circuit hamiltonien, c'est à dire un chemin traversant exactement une fois chacun des sommets $x_o, x_1,\ldots x_n$ du réseau. Un tour peut être considéré comme un chemin élémentaire issu de x_o ,comprenant n sommets intermédiaires, et d'extremité x_o.

Nous appellerons 0-circuit tout chemin (non nécessairement élémentaire) issu de x_o, comprenant n sommets intermédiaires distincts de x_o,et d'extrémité x_o; en particulier, un tour est un 0-circuit et, réciproquement, un 0-circuit c est un tour si et seulement s'il traverse exactement une fois chaque sommet intermédiaire; sinon il existe quelque sommet x_i $(\neq x_o)$ qui n'est pas atteint par c, et aussi quelque autre sommet x_j $(\neq x_o)$ qui est traversé au moins deux fois par c, c'est à dire que la "partie intermédiaire" obtenue en ôtant de c le sommet x_o et les deux arcs incidents à x_o, comprend (au moins) un circuit (non nécessairement

élémentaire) sur certains des "sommets intermédiaires" $x_1,\ldots x_n$.

Nous sommes naturellement conduits à définir, pour chaque 0-circuit c, son vecteur associé $a_c \in \mathbb{R}^n$ dont chacune des composantes $a_{c,i}$ représente le nombre de fois que c traverse le sommet x_i ($\neq x_0$). Clairement $a_c \geq \underline{0}$, $\underline{1} \cdot a_c = \sum_i a_{c,i} = n$ (où $\underline{1}$ désigne le n-vecteur dont toutes les composantes sont 1), et un 0-circuit T est un tour si et seulement si son vecteur associé $a_t = \underline{1}$.

1.2 Reseau multiparti associé

La recherche d'un 0-circuit de longueur minimale est équivalente à la recherche d'un plus court chemin dans un réseau multiparti RM = (N,A,c) défini comme suit :

(i) N est l'ensemble des "noeuds" (nommés ainsi pour les distinguer des sommets) regroupés en "phases" de la manière suivante

phase 0 (origine) noeud (0,0) associé à x_o

phases t (intermédiaires) où t = 1,2,...,n
noeuds (i,t) associés aux sommets x_i

phase n+1 (finale) noeud (0,n+1) associé à x_o

(ii) A est l'ensemble des arcs reliant les noeuds d'une phase aux noeuds de la phase suivante

l'arc ((i,t),(j,t+1)) est défini pour t=0,1,2,...,n si l'arc (x_i, x_j) est dans E
(si t = 0 alors i = 0
si t = n alors j = 0)

(iii) c définit la longueur des arcs

$c((i,t),(j,t+1)) = d_{ij}$ lorsque d_{ij} est défini et t = 0,1,...,n.

Remarque : il n'y a pas d'arc ((i,t),(i,t+1)) puisque R est supposé sans boucle.

Exemple 1 : tiré de Wagner [33] (les sommets ont été renumérotés)

↗	x_0	x_1	x_2	x_3	x_4
x_0		1	10	15	2
x_1	10		25	25	10
x_2	9	8		20	10
x_3	10	14	24		15
x_4	8	10	25	27	

Distances

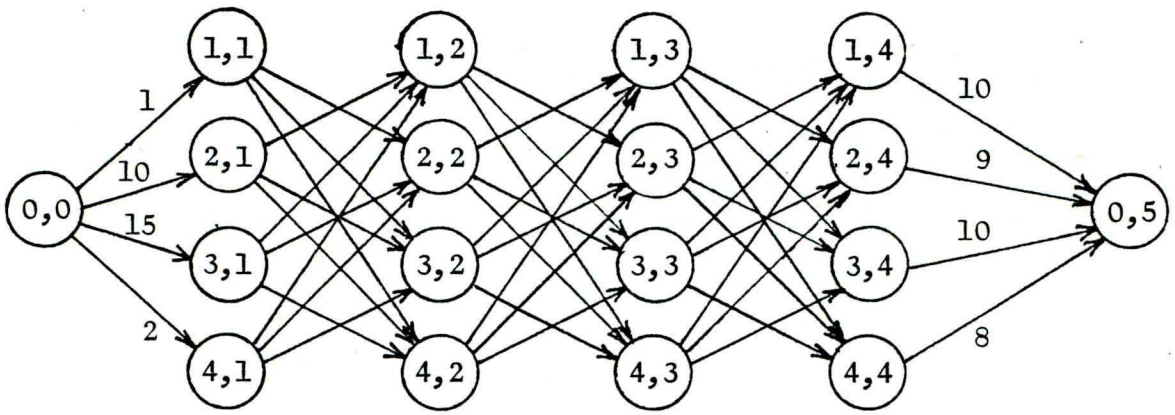

Figure 1. Réseau multiparti

Les longueurs des arcs entre les phases 1-2, 2-3 et 3-4 sont identiques et données dans le tableau des distances.

Un plus court chemin joignant l'origine (0,0) à l'extremité (0,5) est (0,0)-(1,1)-(4,2)-(1,3)-(4,4)-(0,5) , noté simplement 0-1-4-1-4-0 de longueur 39. Son vecteur associé est (2 , 0 , 0 , 2).

Un algorithme de programmation dynamique, décrit par R. Saigal [31] (voir aussi [29]) et reposant sur le principe d'optimalité suivant

$$l(i,t) = \text{Min}\left\{ l(j,t-1) + d_{ji} \;\middle|\; (x_i, x_j) \in E \right\} \qquad (1.1)$$

permet de trouver un plus court chemin (un 0-circuit de longueur minimale) en $O(n \times |E|)$ opérations (additions, comparaisons). Si le réseau est complet, ce nombre devient exactement :

$n\,(n-1)^2 + n$ additions
$(n-1)^3 + (n-1)$ comparaisons.

(Voir l'annexe 2 pour une variation sur cet algorithme).

2. Une formulation par la Programmation Linéaire du Problème du Voyageur de Commerce

2.1 Définitions et premières propriétés

Désignons par C l'ensemble des O-circuits définis dans le réseau R, et par ℓ_c la longueur (somme des longueurs des arcs) d'un O-circuit c.

Considérons le programme linéaire en nombres entiers suivant:

$$(PLI)\begin{cases} \text{Minimiser} \quad z(x) = \sum_{c \in C} \ell_c \, x_c \\ \text{s.c.} \begin{cases} x_c \geq \underline{0} \text{ , entier} \\ \sum_{c \in C} \partial_c \, x_c = \underline{1} \end{cases} \end{cases} \qquad (2\text{-}1)$$

et le programme linéaire déduit en relâchant les contraintes d'intégrité sur x:

$$(P.L.)\quad \begin{cases} \text{Minimiser} \quad z(x) = \sum_{c \in C} l_c x_c \\ \text{s.c.} \begin{cases} x \geq \underline{0} \\ \sum_{c \in C} a_c x_c = \underline{1} \end{cases} \end{cases} \qquad (2.2)$$

programme linéaire à $n(n-1)^{n-1}$ variables non-négatives et n contraintes.

Remarque:
Il est clair que, dans la solution optimale de (PLI) ou de (PL), si c et c' ont même vecteur associé, et si $l_c < l_{c'}$ alors $x_{c'} = 0$; ceci permet de réduire considérablement le nombre de variables. Dans toute la suite, l'ensemble C sera réduit aux 0-circuits de longueur minimale parmi ceux ayant le même vecteur associé.

Lemme 2-1. Il existe un seul point entier satisfaisant aux contraintes de (PLI), et ce point représente le tour optimal dans le réseau R.

Preuve :
En additionnant les n contraintes de (PLI) on obtient

$$\sum_{c \in C} n\, x_c = n$$

soit

$$\sum_{c \in C} x_c = 1 \qquad (2-3)$$

Comme x est non-négatif, les seules solutions entières à (2.3) sont telles qu'il existe un 0-circuit c* tel que

$$x_{c^*} = 1 \qquad (2.4)$$

et $x_c = 0$ pour tout $c \in C$, $c \neq c^*$

d'où $a_{c^*} = \underline{1}$ c'est à dire que c* est un tour dans R, et d'après la remarque ci-dessus, un tour optimal. #

Lemme 2.2: L'unique point entier, défini dans le lemme 2.1, est un point extrême du polyèdre P défini par les contraintes de (PL), et il est adjacent à tous les autres points extrêmes de P.

Preuve: D'après le lemme 2.1, le polyèdre P est non vide. Considérons alors un point extrême de P, une base réalisable B correspondante et la solution $\bar{x}$ définie par

$$\bar{x}^B = B^{-1}\,\underline{1}$$
$$\bar{x}^R = \underline{0} \qquad (2.5)$$

(les notations sont celles de [32]).

Remarquons que l'ensemble S des indices s tels que

$$\bar{x}^B_s > 0 \qquad (2.6)$$

est non vide.

Si le vecteur a_T, associé au tour optimal, est dans la base B, alors la solution x définie par les relations (2.4) est le programme de base, puisque $B^{-1}x = \underline{1}$. Le point extrême correspondant est bien le point entier du Lemme 2.1.

Si le vecteur a_T n'est pas dans la base B, nous allons démontrer qu'une simple opération de pivotage suffit à l'y faire entrer. En définissant

$$y_T = B^{-1}a_T = B^{-1}\underline{1} \qquad (2.7)$$

il vient immédiatement

$$y_T = \bar{x}^B \qquad (2.8)$$

Une première conséquence est que $y_T > 0$, donc l'opération de pivotage est permise. De plus, l'ensemble des indices s tels que $y_{sT} > 0$ est identique à S (ensemble des indices s tels que $\bar{x}_s > 0$). Alors le rapport $\frac{\bar{x}_s}{y_{sT}}$, dont le minimum sur S définit le vecteur sortant de la base, est égal à 1 pour tout s e S, donc on peut substituer a_T à tout vecteur a_s, pour s e S et l'on obtient le point entier par cette simple opération de pivotage. #

2.2 Génération de colonnes

Un procédé très général pour aborder ce genre de programme linéaire, où les variables sont en fait définies implicitement, est d'utiliser la méthode revisée du simplexe: La génération de la colonne entrant dans la base s'effectue en résolvant un sous-problème de nature généralement combinatoire (voir par exemple [11], [13], [27] [21, chap.4].

Lemme 2-3. La génération de la colonne devant entrer dans la base se ramène à la recherche d'un 0-circuit de longueur minimale dans un réseau associé.

Preuve :

Soit B une base réalisable de (PL) et soit la variable duale associée.

La variable candidate est choisie en calculant le minimum de

$$\hat{\ell}_c = \ell_c - \pi \cdot a_c$$

pour les variables hors-base (si c est dans la base $\hat{\ell}_c = 0$). Une variable x_r sera donc choisie pour entrer dans la base tant que $\hat{\ell}_r = \min_{c \in C} \hat{\ell}_c < 0$
Pour un circuit c le calcul de ℓ_c revient à affecter une "pénalité" $-i$ à la longueur de c chaque fois que le sommet $x_i \neq x_o$ est traversé; cette pénalité peut être répartie sur les arcs adjacents au sommet i en définissant les "longueurs pénalisées" suivantes:

$$d'_{ij} = d_{ij} - \lambda \pi_i - (1-\lambda)\pi_j$$

où $\lambda \in [0,1]$ est un paramètre: on utilisera par exemple les valeurs $\lambda = \frac{1}{2}$ pour conserver éventuellement la symétrie du réseau, ou $\lambda = 0$ (resp 1) pour préserver l'ordre des longueurs des arcs indicents en x_i (resp. issus de x_i). #

2.3 Pénalités optimales

Pour tout vecteur π, désignons par C_π un O-circuit de longueur pénalisée minimale, et par $W(\pi)$ cette longueur:

$$W(\pi) = \ell_{C_\pi} - \pi.a_{C_\pi} = \min_{c \in C} (\ell_c - \pi.a_c) \tag{2.10}$$

Pour tout $c \in C$, on a

$$W(\pi) \leq \ell_c - \pi.a_c \tag{2.11}$$

et en particulier pour le tour optimal T, de longueur ℓ_T

$$W(\pi) \leq \ell_T - \pi.\underline{1} \tag{2.12}$$

Lemme 2.4: Pour tout π et tout réel k

$$W(\pi + k\underline{1}) = W(\pi) - nk \tag{2.13}$$

Preuve: Pour tout O-circuit c, on a $\underline{1}.a_c = n$, d'où

$$\ell_c - (\pi + k\underline{1}).a_c = \ell_c - \pi.a_c - nk$$

donc une translation de π dans la direction de $\underline{1}$ se traduit par une même translation dans les longueurs pénalisées des O-circuits. #

On peut donc se restreindre aux pénalités π telles que

$$\pi.\underline{1} = 0 \tag{2.14}$$

Alors la quantité $W(\P)$ apparait, en combinant (2.12) et (2.14) comme une borne inférieure sur ℓ_T. Le problème de maximiser cette borne, soit

$$\text{Max}\{W(\P)/\P.\underline{1} = 0\} \quad \text{avec} \quad W(\P) = \underset{c \in C}{\text{Min}} (\ell_c - \P.a_c)$$

est un problème de "Maximin", et peut se formuler comme un programme linéaire:

$$\begin{aligned} (D) \quad & \text{Max} \quad W \\ & \text{s.c.} \quad W + \P.a_c \leqslant \ell_c \quad \text{pour tout } c \in C \\ & \qquad \P.\underline{1} = 0 \\ & \qquad (W, \P \text{ non-astreints}) \end{aligned}$$

Lemme 2.5: (PL) est équivalent au dual de (D)

Preuve: le dual de (D) s'écrit:

$$\begin{aligned} & \text{Min} \quad \sum_c \ell_c x_c \\ & \text{s.c.} \quad \sum_c x_c = 1 \qquad (2.15) \\ & \qquad \sum_c a_{c,i} x_c + y = 0 \quad \text{pour } i = 1,\ldots,n \qquad (2.16) \\ & \qquad (x_c \geqslant 0, \quad y \text{ non-astreint}) \end{aligned}$$

En additionnant les contraintes (2.16) et en comparant avec (2.15), il vient

$$y = -1$$

Les contraintes (2.16) sont donc équivalentes aux contraintes (2.1) et il a été démontré, dans la preuve du Lemme 2.1 que la relation (2.15) est une conséquence des contraintes (2.1). #

Remarque: (P) n'est pas formellement le dual de (D). En particulier, pour toute solution de base de (PL), de valeur z, les variables duales ¶ vérifient

$$¶.\underline{1} = z$$

ce qui est en contradiction avec (2.14)

THEOREME 2.6: Les deux propriétés suivantes sont équivalentes

(i) il existe des pénalités ¶ telles que le tour optimal T est un O-circuit de longueur pénalisée minimale.

(ii) la solution entière est solution optimale de (PL)

Preuve: (ii) $\Rightarrow$ (i) si la solution entière est optimale, le maximum z^* de la valeur z de (PL) vérifie:

$$z^* = \ell_T$$

Considérons une solution optimale de (PL) telle que tous les coûts relatifs soient non-positifs (cette solution n'est pas nécessairement la solution entière) et soient ¶* les variables duales correspondantes:

$$¶^*.\underline{1} = z^* = \ell_T$$

Pour ces pénalités ¶*, le tour optimal a une longueur nulle et tout chemin a une longueur non-positive donc la propriété (i) est vérifiée.

(i) = (ii) si la solution entière n'est pas optimale pour (PL) on a:

$$z^* < \ell_T$$

D'après le lemme 2.5, z^* est aussi la valeur optimale du dual de (D), donc pour toute pénalité ¶ vérifiant (2.14) on a:

$$W(¶) \leqslant z^*$$

d'où $$W(¶) < \ell_T \qquad (2.17)$$

Soit ¶ une pénalité quelconque, et posons:

$$P = ¶.\underline{1}$$

Supposons que le tour T soit un plus court chemin pour ¶, alors:

$$W(¶) = \ell_T - ¶.\underline{1} = \ell_T - P$$

définissons alors la pénalité ¶' par

$$¶' = ¶ - \frac{P}{n}\underline{1}$$

¶' vérifie (2.14) et, d'après le lemme 2.4,

$$W(¶') = W(¶) + n\frac{P}{n} = \ell_T$$

ce qui contredit (2.17) #

Bien que le point entier soit adjacent à tous les autres points extrêmes du polyèdre (P), il n'est possible de l'atteindre par un pivotage

basé sur le critère d'entrée "coût relatif minimum", que si ce point entier est, en fait, solution optimale de (PL).

2.4 Relation avec le Problème d'Affectation

Le théorème suivant, démontré dans [26], établit que la borne inférieure $\bar{z}$ fournie par (PL) est uniformément meilleure que la borne z_A obtenue en résolvant le Problème d'Affectation (PA) défini par le tableau des distances d (les éléments diagonaux sont supposés infinis). De plus, sous la restriction que l'affectation optimale est unique, il ne peut y avoir égalité entre $\bar{z}$ et z_A que si ces deux valeurs sont en fait confondues avec la longueur du tour optimal.

Théorème 2-5. Soit z_A la valeur de l'affectation optimale dans le réseau R, $\bar{z}$ la valeur optimale de la fonction objectif de (PL) et ℓ_T la longueur du tour optimal, on a :

(i) $$z_A \leqslant \bar{z} \leqslant \ell_T$$

(ii) si la solution optimale du Problème d'Affectation (PA) est unique et si

$$z_A = \bar{z}$$

alors le tour optimal est à la fois solution optimale de (PA) et de (PL).

Exemple 2 : il a été montré dans [26 p.19] que la solution optimale de (PL) est constituée par les deux 0-circuits

c_1 0-1-4-1-4-0 $\ell_{c_1} = 39$ (voir exemple 1)

c_2 0-2-3-2-3-0 $\ell_{c_2} = 84$

avec $\bar{x}_{c_1} = \bar{x}_{c_2} = \frac{1}{2}$ et $\bar{x}_c = 0$ pour $c \neq c_1$ et $c \neq c_2$

On a donc $\bar{z} = 61.5$

L'affectation optimale donne $z_A = 60.$ et le tour optimal $\ell_T = 62.$ (voir 33 p.103 . On a bien $60. \leqslant 61.5 \leqslant 62.$

(Remarque: les sommets ont été renumérotés par rapport à [26] ou [33])

3. Optimisation par sous-gradient

3.1 Itérations de sous-gradient

Une expérimentation préliminaire, utilisant l'algorithme du simplexe et la génération de colonnes pour résoudre (PL), a fait apparaître un phénomène appelé "tailing off" dans la littérature de langue anglaise, et caractérisé par une convergence très lente. Après les quelques premières itérations, la valeur de la fonction-objectif ne décroît plus que très lentement ("effet de plateau"), et plusieurs centaines d'itérations (résolution de sous-problèmes) sont nécessaires pour de faibles améliorations dans la valeur de la solution. Ce comportement, habituel pour les programmes linéaires de grande taille, apparaît, dans le cas de problèmes à variables implicites utilisant la génération de colonnes, déjà pour des problèmes de taille moyenne, ou même de petite taille (disons $n \geq 15$). Voir par exemple [11], [13].

De plus la solution optimale de (PL) ne fournit généralement qu'une borne inférieure sur la longueur du tour optimal, mais cette borne n'est valide que lorsque l'optimum est atteint. Pour obtenir, en un nombre raisonnable d'itérations, une borne inférieure valide et satisfaisante, il convient alors de recourir à des méthodes approchées appliquées au dual de (PL). Nous allons décrire la méthode d'optimisation par sous-gradient, définie par Held et Karp pour leur formulation du TSP symétrique, et généralisée dans [16].

Revenons au problème (D), défini au paragraphe 2.3:

$$(D) \quad \text{Max}\{W(\pi)/\pi.\underline{1} = 0\} \qquad \text{avec} \quad W(\pi) = \underset{c \in C}{\text{Min}} \ (\ell_c - \pi.a_c)$$

et rappelons que pour tout π, c_π désigne un 0-circuit de longueur pénalisée minimale.

Posons $Q = \{\pi \in R \ / \ \underline{1}.\pi = 0\}$. La projection sur Q, pour la norme euclidienne, est définie par:

$$P_Q(x) = x - \frac{\underline{1}.x}{n} \tag{3.1}$$

Une itération de sous-gradient pour la résolution de (D) est définie dans [15] comme:

$$\pi^{j+1} = P_Q(\pi^j - t_j.a_j) \quad \text{où } a_j \text{ désigne } a_{c_{\pi j}}$$

On a donc, si $\pi^o \in Q$

$$\pi^{j+1} = \pi^j + t_j(\underline{1} - a_j) \tag{3.2}$$

(puisque $\underline{1}.a_c = n$ pour tout $c \in C$)

Dans cette relation t_j désigne une suite de réels ("pas") positifs. Un choix possible pour t_j est:

$$t_j = \frac{B - W(\pi^j)}{||\underline{1} - a_j||^2} \tag{3.3}$$

où B est une borne supérieure sur Max W(π) (par exemple, B est la longueur d'un tour, déterminé par une méthode heuristique telle celle de Lin [22] ou Lin et Kernighan [23] dans le cas symétrique).

La relation (3.3) peut être considérée comme une tentative de forcer $W(\P^{j+1})$ à prendre la valeur B (en effet $\ell_j - \P^{(j+1)} = B$ lorsque $\P^{(j+1)}$ est défini par (3.2) et (3.3)). Pour une discussion du choix de t_j, voir [16].

Il est nécessaire de définir un critère d'arrêt pour garantir la finitude du procédé. Si pour un indice j, $c_{\P j}$ est un tour, alors le problème est résolu; sinon on peut décider d'arrêter l'itération lorsque la valeur maximale de W n'a pas été améliorée depuis P itérations, où P est un paramètre fixé à l'avance.

Exemple 3 : dans l'exemple 1, prenons $\pi^o = \underline{0}$. Le plus court chemin correspondant est, on l'a vu, 0-1-4-1-4-0 dont le vecteur associé est (2, 0, 0, 2) et $W(\pi^o) = 39$.

Considérons le tour 0-3-2-1-4-0 de longueur 65. et utilisons cette valeur pour B. Alors $\underline{1} - \underline{a}_o = (-1, 1, 1, -1)$ et $t_o = \frac{65-39}{4} = 6.5$ d'où $\pi^1 = (-6.5, 6.5, 6.5, -6.5)$

Les itérations suivantes sont résumées dans le tableau suivant:

Itération	π				c	W
0	0.	0.	0.	0.	0-1-4-1-4-0	39.
1	-6.5	6.5	6.5	-6.5	0-2-3-2-3-0	58.
2	-4.75	4.75	4.75	-4.75	0-2-1-4-1-0	57.5
3	-8.5	4.75	8.5	-4.75	0-2-3-2-3-0	57.5

4	-6.625	2.875	6.625	-2.875	0-2-4-1-4-0	57.5
5	-6.6 5	2.875	10.375	-6.625	0-2-3-2-3-0	57.5
6	-4.75	1.	8.5	-4.75	0-1-4-1-4-0	58.
7	-6.5	2.75	10.25	-6.5	0-2-3-2-3-0	58.
8	-4.75	1.	8.5	-4.75	0-1-4-1-4-0	58.
9	-6.5	2.75	10.25	-6.5	0-2-3-2-3-0	58.

A partir de la $6^{\text{ème}}$ itération, le procédé cycle et l'on a :

$$\pi^{j+2} = \pi^{j} \qquad \text{pour } j \geq 6$$

Ceci est dû au fait que les deux 0-circuits obtenus pour ces itérations définissent la solution optimale de (PL) (voir Exemple 2).

3.2 Comparaison avec la programmation linéaire

Comme il a été souligné plus haut, un tel procédé fournit à tout moment une borne inférieure valide sur la longueur du tour optimal, par contraste avec les itérations du simplexe. Toutefois, ainsi qu'il a été noté par Dantzig et al.[6], on peut déduire, à chaque pas de la minimisation d'un programme linéaire par le simplexe, une borne inférieure $\underline{z}$ sur la valeur de la solution optimale (et a fortiori sur la longueur du tour optimal) par:

$$\underline{z} = z + \operatorname{Min}_{c \in C}(\ell_c - \pi.a_c) \qquad (3.4)$$

où z est la valeur de la fonction-objectif

π les variables duales correspondantes

c l'indice de la variable entrante

(il suffit de rappeler que $z = \pi.\underline{1}$)

Ainsi les deux méthodes, simplexe et sous-gradient, peuvent être vues comme des procédés qui engendrent une séquence de pénalités π fournissant une séquence de bornes inférieures sur la longueur du tour optimal. Le caractère fini de la convergence du simplexe vers la meilleure borne, ne suffit pas à assurer que cette méthode est préférable à celle du sous-gradient, à cause du caractère extrêmement erratique des valeurs de $\underline{z}$ au fil des itérations. La figure 2 illustre le comportement des deux méthodes sur un problème de taille n = 32 tiré de [18], pour les 50 premières itérations. Les deux méthodes utilisent la même solution de départ ("pseudo-couplage", cf Annexe 1) et la même routine pour les plus courts-chemins (PCCR2 avec k = 4, cf Annexes 2 et 3). L'allure de la courbe supérieure,

valeurs successives de la fonction objective de PL, indique que plusieurs centaines d'itérations doivent être nécessaires avant que l'optimum ne soit atteint. Après 50 itérations, w a été amélioré 15 fois et se trouve à 2.75% de l'optimum; la valeur de $\underline{z}$ a été améliorée 5 fois et se trouve encore à 9.19% de l'optimum.

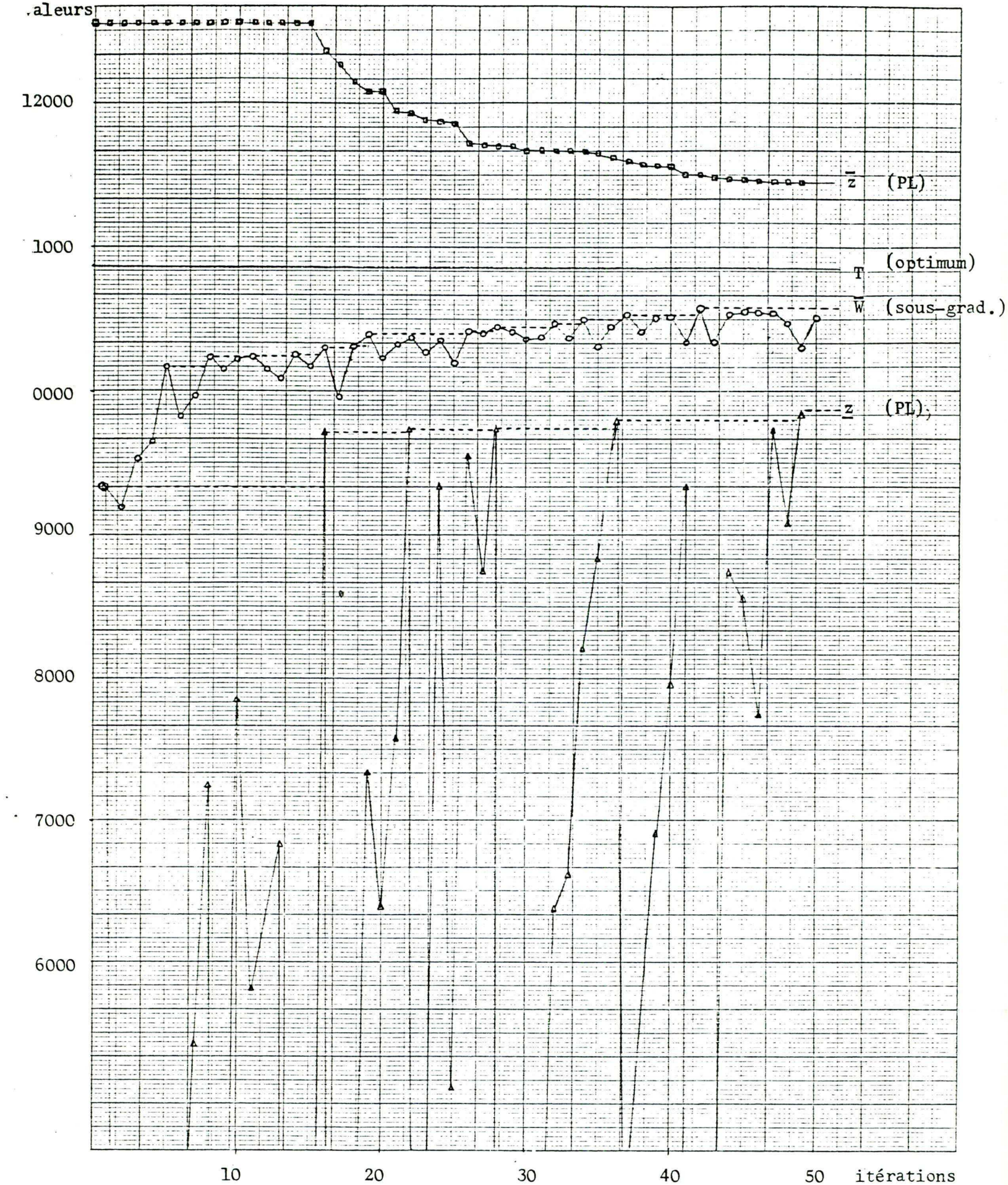
aleurs
12000
1000
0000
9000
8000
7000
6000
$\bar{z}$ (PL)
T (optimum)
$\bar{W}$ (sous-grad.)
$\underline{z}$ (PL)
10
20
30
40
50
itérations

Figure 2

4. Enumération implicite

Une méthode d'énumération implicite ("Branch and Bound", "Séparation et Evaluation Progressive") est définie par:

(i) un principe de séparation de l'ensemble des solutions réalisables (ici l'ensemble des n! tours dans le réseau R)

(ii) une procédure d'évaluation d'une borne (inférieure dans un problème de minimisation) sur la valeur de la meilleure solution (tour de longueur minimal) dans les sous-ensembles ainsi engendrés.

Un sous-problème (P) est défini de la manière suivante:

soit $F \subseteq E$ un ensemble d'arcs, dits "forcés"

$I \subseteq E$ un ensemble (disjoint de F) d'arcs, dits "interdits"

$C(F,I) \subseteq C$ l'ensemble des O-circuits qui empruntent exactement une fois chaque arc forcé, et aucun arc interdit.

On utilisera l'optimisation par sous-gradient pour calculer W(F,I) qui sera une borne inférieure sur la longueur des tours contenus dans C(F,I). Si C(F,I) ne contient aucun tour, cette borne pourra être arbitrairement grande.

Afin de réduire la taille des problèmes de plus court chemin rencontrés lors de l'évaluation de W(F,I), l'ensemble F constituera un chemin (x_u, x_o) d'extrémité x_o comprenant $|F| = k$ arcs. Si l'on désigne par V_F l'ensemble des sommets du chemin F (alors $|V_F| = k + 1$), le réseau multiparti associé aura pour origine (0,0), pour ensemble de sommets intermédiaires $V-V_F$, reproduits sur n-k phases, et pour extrémité (u,n-k + 1). De plus aucun des arcs interdits ne pourra figurer dans ce réseau.

Illustrations: a) dans l'exemple 1, supposons que

$F = \{(x_3, x_0)\}$ $\quad V_F = \{x_3\}$ $\quad k = 1$

$I = \emptyset$

le réseau multiparti associé est représenté Fig. 3

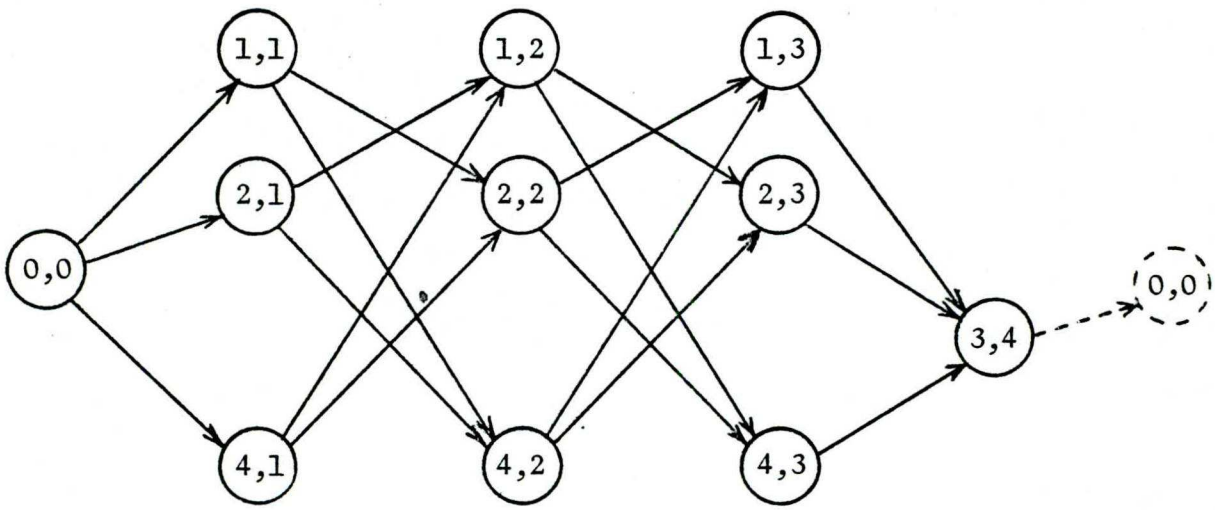

Fig. 3

b) dans l'exemple 1, supposons que

$$F = \emptyset$$

$$I = \{(x_3, x_0)\}$$

le réseau multiparti associé est représenté Fig. 4.

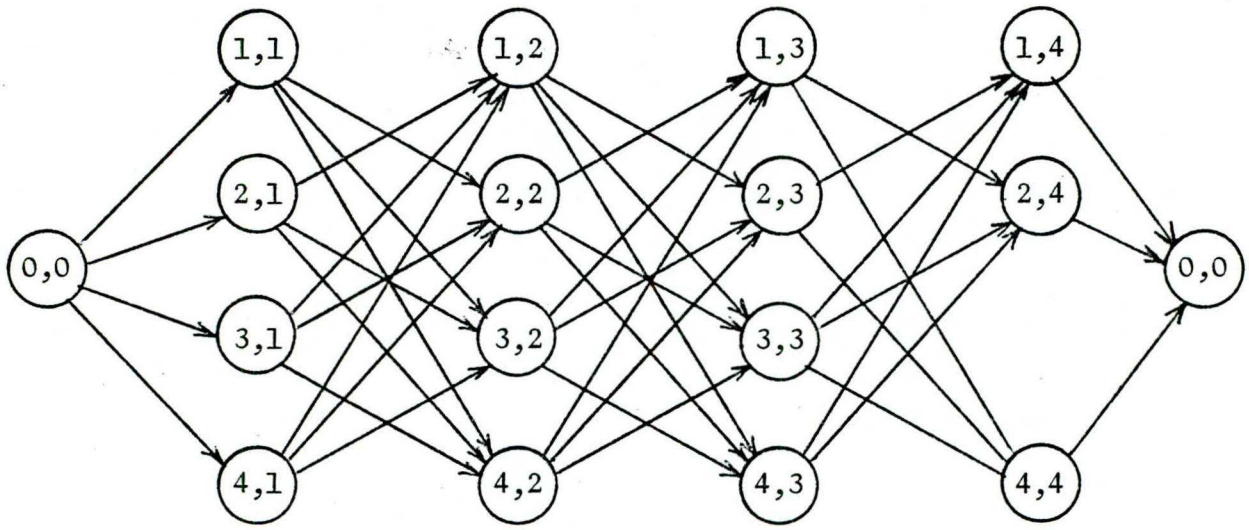

Fig. 4

Remarquons que l'on peut éliminer le noeud (3,4) et les arcs ((1,3),(3,4)), ((2,3),(3,4)), ((4,3),(3,4)) qui lui sont adjacents.

Le principe de séparation (i) sert à définir les ensembles F et I. Nous utiliserons pour cela le O-circuit c qui définit la plus grande valeur

W(F,I) rencontrée au cours de l'optimisation par sous-gradient. Soit $e = (x_v, x_u)$ le dernier arc de c, nous définirons deux sous-problèmes (P') et (P") du problème (P) de la manière suivante:

$$(P') \qquad F' = Fu\{e\} \qquad I' = I$$

$$(P") \qquad F" = F \qquad I" = Iu\{e\}$$

Le sous-problème (P') a une phase de moins que (P) et les noeuds définis par le sommet x_v ont été éliminés. Si $\pi \in R^{n-K}$ désigne la pénalité définissant max W(F,I) (pour laquelle c est un plus court chemin) alors une bonne pénalité de départ pour (P') pourra être $\pi' \in R^{n-K-1}$ défini par:

$$\pi'_i = \pi_i + \frac{\pi_v}{n-K-1} \quad \text{pour les indices i tels que } x_i \notin V_F u\{x_v\}$$

Exemple 4 dans l'exemple 1, on peut prendre $e = (x_3, x_0)$, le réseau associé à (P') est représenté Fig. 3.

On a: $\pi = [-6.5 \quad 6.5 \quad 6.5 \quad -6.5] \in R^4$

$\pi' = [-4.333 \quad 8.333 \quad -4.333] \in R^3$

Si le problème (P) contient exactement deux noeuds dans la dernière phase, alors (P') n'en contient plus qu'un seul. Le $(n-k)^{ème}$ sommet est alors fixé et (P") peut être traité comme (P').

Dans le cas général où (P) a au moins trois sommets dans la dernière phase, la dimension de (P") est la même que celle de (P) (même nombre de phases, même nombre de sommets intermédiaires), mais on peut éliminer (v,n-k) de la dernière phase, ainsi que les arcs qui lui sont adjacents.

De plus $\P$ est une bonne pénalité de départ et le plus court chemin dans C(F",I") peut être facilement obtenu.

Exemple 5 dans l'exemple 1, prenons toujours e = (x_3,x_0); le réseau associé est représenté Fig. 4; pour le vecteur $\P$, le plus court chemin s'obtient en cherchant

$$\min\{\ell(i,4) + d_{i0} \mid i \neq 3\}$$

ce minimum est atteint pour i = 1 et le plus court chemin correspondant est 0-2-3-2-1-0.

L'"arbre d'énumération" est une arborescence, ayant pour racine un noeud représentant le problème initial, et dont chaque noeud représente un sous-problème (P) et admet, à moins qu'il ne soit "terminal", deux suivants (P') et (P") définis plus haut. Un noeud est "terminal" dans l'un des cas suivants:

(i) un O-circuit minimal est un tour

(ii) l'évaluation W associée dépasse une borne supérieure B sur la longueur du tour optimal (dans le réseau initial).

On peut évaluer la taille de l'arbre d'énumération: un chemin partant de la racine contient au plus (n-1) problèmes de dimension n, (n-2) problèmes de dimension (n-1),..., deux problèmes de dimension 3, un problème de dimension 2 et un problème (trivial) de dimension 1. Le plus long chemin dans l'arbre d'énumération contiendra au plus $\frac{n(n-1)}{2} + 1$ problèmes.

Le caractère binaire de la séparation permet l'usage du principe d'exploration appelé "backtracking", que l'on peut énoncer ainsi

"Si le problème (P) considéré n'est pas terminal, alors
sélectionner (P')
sinon remonter dans l'arbre jusqu'à rencontrer un problème de type (Q'), (c'est-à-dire que l'autre problème (Q") n'a pas encore été sélectionné) et sélectionner (Q")".

Si tous les problèmes rencontrés dans la remontée jusqu'à la racine sont du type (Q"), alors l'énumération est terminée.

Exemple : Résolution du problème de Wagner [33], exposé dans l'exemple 1. On a choisi le sommet 2 comme origine pour rénuméroter les sommets car les quatre autres choix de l'origine donnent le tour optimal sans avoir besoin d'énumération.

Le problème initial donne une borne W = 58. (voir exemple 2)

On sélectionne d'abord le sous-problème (P') (voir exemple). La séquence des plus courts-chemins successifs est donnée ci-dessous.

Itération	¶			c	W
1	-4.3333	8.3333	-4.3333	0-2-1-2-3(-0)	60.
2	-4.3333	6.1667	-1.8333	0-2-4-2-3(-0)	64.5
3	-4.0833	5.9167	-1.8333	0-2-4-1-3(-0)	65.

on a obtenu un tour de longueur 65. (P') est terminal.

On sélectionne (P") (voir exemple 5)

Itération	¶				c	W
0	-6.5	6.5	6.5	-6.5	0-2-3-2-1-0	59.
1	-6.5	3.5	6.5	-3.5	0-2-4-1-4-0	58.
2	-6.5	3.5	10.	-7	0-2-3-2-1-0	61.5
3	-6.5	1.75	10.	-5.25	0-2-3-1-4-0	62.

on a obtenu un tour de longueur 62. et (P") est terminal.

On remonte jusqu'à la racine; l'énumération est donc terminée et le tour optimal est 0-2-3-1-4-0, de longueur 62. L'arbre d'énumération est représenté Fig. 5.

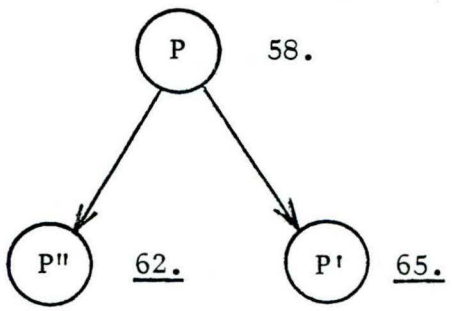

Fig. 5

L'inconvénient d'une exploration par "bactracking" est que l'on risque d'évaluer plus de problèmes qu'avec le principe "jumptracking", qui peut s'énoncer ainsi:

"Sélectionner le problème candidat (c'est-à-dire non-terminal et pas encore sélectionné) dont l'évaluation est la plus faible".

Son avantage est d'abord dans le volume de mémoire utilisé: il n'est pas nécessaire de conserver une liste de problèmes candidats, qui peut devenir de taille explosive. En effet, on peut adapter ici la représentation vectorielle introduite par Geoffrion (voir [9 p. 125] et la taille du vecteur est bornée par $\frac{n(n-1)}{2} + 1$. Quant au temps de calcul, la définition des sous-problèmes (P') étant facile dans les phases de descente, il se peut que ce fait compense le temps consacré à un plus grand nombre d'évaluations.

5. EXTENSIONS

L'approche présentée pour le Problème du Voyageur de Commerce peut s'appliquer à une large classe de problèmes de tournées, grâce au fait que la partie essentielle en est un problème de plus-court-chemin. C'est pourquoi on peut étendre facilement cette approche à des problèmes dont nous allons décrire quelques exemples et qui ne semblent pas pouvoir être formulés à l'aide d'"arbre de poids minimum" ou d'"affectation optimale".

5.1 PROBLEME GENERAL D'ORLOFF ET VARIATIONS

C.S. Orloff a défini dans [25] un problème général dans lequel un véhicule doit visiter certains noeuds et certaines arêtes, et il indique une méthode d'approche basée sur l'utilisation du b - couplage ("b - matching", voir [7]). Cette approche fonctionne mieux lorsque le problème est plus proche du type "Postier Chinois" (voir [8]) que du type "Voyageur de Commerce". De plus lorsque le réseau est orienté ou mixte (c'est-à-dire contient à la fois des arcs-orientés et des arêtes non-orientées), son approche s'applique moins facilement.

Nous allons donner une formulation, qui, au contraire, s'applique mieux lorsque le problème est orienté on mixte, et lorsqu'il est plus proche du type Voyageur de Commerce.

Soit un réseau mixte R = (V, A, E) (V ensemble des sommets, A ensemble des arcs orientés, E ensemble des arêtes non-orientées) et les sous-ensembles suivants:

$V' \subseteq V$ ensemble des sommets à visiter

$A' \subseteq A$ ensemble des arcs à visiter

$E' \subseteq E$ ensemble des arêtes à visiter

Soit $N = V' \cup E' \cup A$ où $|N| = n+1$

Nous définirons un réseau multiparti à n phases. Dans chaque phase t, les noeuds seront:

(i,t) pour les sommets $x_i \in V'$

(i,j,t) pour les arc $(x_i,x_j) \in A'$

(i,j,t) et (j,i,t) pour les arêtes $(x_i,x_j) \in E'$ (deux représentants pour chaque arête.)

Les distances seront calculées dans le réseau original et si

Δ_{ij} est la longueur du plus-court chemin de x_i à x_j dans R ($\Delta_{ii} = 0$), nous aurons:

$$c((i,t),(j,t+1)) = \Delta_{ij}$$
$$c((i,t),(j,k,t+1)) = \Delta_{ij} + d_{jk}$$
$$c((i,j,t),(k,t+1)) = \Delta_{jk}$$
$$c((i,j,t),(k,l,t+1)) = \Delta_{jk} + d_{k,l}$$

Il n'y aura évidemment pas d'arc $((i,j,t),(j,i,t+1))$.

On prendra pour origine, à moins que le point de départ ne soit imposé, un sommet dans V', ou bien l'extrémité d'un arc dans A'. De même l'extrémité sera le point d'arrivée imposé, le sommet déjà choisi comme origine ou l'origine de l'arc choisi.

ILLUSTRATION: Dans l'exemple 1, prenons

$$V' = \{0\}$$
$$A' = \{(2,3)\} \quad \text{avec } d_{23} = 20$$
$$E' = \{(1,4)\} \quad \text{avec } d_{14} = d_{41} = 10$$

Le réseau multiparti ayant le sommet 0 pour origine et pour extrêmité est représenté Fig. 6

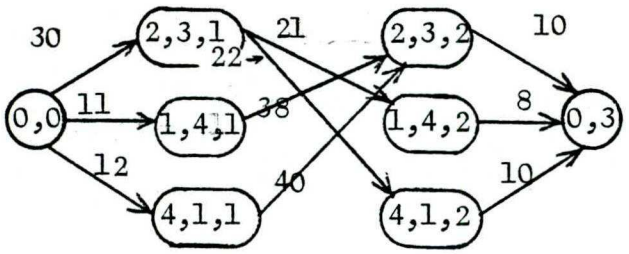

FIG. 6

Les distances sont indiquées sur la figure. Par exemple:

$$c((1,4,1),(2,3,2)) = \Delta_{4,2} + d_{2,3} = 18 + 20$$

Un plus court chemin est:

C=0-(1,4) - (2,3)-0 de longueur 59.

Le vecteur associé $a_c \in \mathbb{R}^n$ est défini comme pour le Problème du Voyageur de Commerce pour les sommets de V', et de même pour les arcs de A'; pour une arête $(x_i, x_j) \in E'$, a_c^{ij} dénote le nombre de fois que le chemin C emprunté l'un ou l'autre des représentants (i,j,t) ou (j,i,t) de cette arête (t=1,...,n).
De même la pénalité π_e associée à une arête $e=(x_i, x_j)$ sera affectée aux noeuds (i,j,t) et (j,i,t). Donc la taille du problème (nombre de phases, dimension de π) sera toujours n.

ILLUSTRATION: le vecteur $\mathbf{a}_c$ associé au chemin C ci-dessus est

$$\mathbf{a}_c = (1,1)$$

c représente une tournée optimale.

5.2 PLUSIEURS VEHICULES

Dans le "Problème à m Voyageurs de Commerce", les clients doivent être visités par l'un de ces m véhicules. Divers problèmes de ce type peuvent être réduit au problème classique du Voyageur de Commerce (voir [12]). Nous présentons maintenant une variante de ce problème qui ne semble pas pouvoir être réduite de la même façon. Dans ce problème, n est supposé être (sans perte de généralité) un multiple de m ($n = pm$) et chaque route, part d'un dépôt central et y revient après avoir visité exactement p clients.

Dans ce cas l'on recherche des 0-circuits comprenant $p+1$ arêtes; il suffit alors de considérer le réseau multiparti avec seulement p phases intermédiaires. Les formulations et méthodes de résolution précédentes s'adaptent alors facilement, par exemple on a:

$$\sum x_c = m$$

Notamment le temps de calcul d'un plus court chemin est de l'ordre de $0(pn^2)$ (dans le cas complet), c'est-à-dire réduit par un facteur de m.

Remarque: pour $m = \frac{1}{2} n$ ce problème se ramène à la recherche d'un couplage parfait de poids minimal, en définissant:

$$P_{ij} = \text{Min.} \left\{ d_{oi} + d_{ij} + d_{jo};\ d_{oj} + d_{ij} + d_{io} \right\}$$

voir l'Annexe 1.

5.3 PROBLEMES AVEC CAPACITE (VEHICLE SCHEDULING)

Dans les problèmes à plusieurs véhicules, supposons qu'une demande D_i soit associée à chaque sommet x_i du réseau et qu'à chaque route soit associée une capacité K (pour simplifier, nous supposerons qu'on ne dispose que d'un seul type de véhicule). La contrainte additionnelle introduite pour chaque route est:

$$\sum_{i \in V_r} D_i \leqslant K \quad \text{pour toute route } r \qquad (5.1)$$

où V_r est l'ensemble des sommets situés sur la route r.
Nous supposerons donc, pour simplifier, que

$$D_i \leqslant K \text{ pour tout } i$$

Dans ce modèle, on impose que chaque demande D_k doit être rencontrée par une livraison seulement.
Pour aborder ce problème, on peut utiliser une extension du réseau multiparti dans laquelle chaque noeud intermédiaire (i, t) est relié à l'extrémité (o,n+1) par un arc de longueur c_{io}.
L'algorithme de plus court chemin décrit au paragraphe 1.2, s'étend aisément à ce cas, ainsi que la définition du vecteur associé et l'usage de pénalités. La restriction introduite par la capacité K revient à ne considérer que l'ensemble des routes qui satisfont (5.1). Les méthodes de la programmation dynamique s'étendent assez bien au cas d'une contrainte supplémentaire (voir [24]). De plus, aussi bien dans la méthode du simplexe que du sous-gradient, il n'est nécessaire d'obtenir une solution exacte au problème du plus court chemin avec capacité (c'est-à-dire à la fois satisfaisant (5.1) et de longueur minimale) que pour garantir la validité du résultat. Enfin, le temps supplémentaire consacré à chaque problème de plus court chemin pour tenir compte de la capacité, pourrait bien être compensé par une réduction importante du nombre de solutions réalisables, et donc d'itérations.

5.4 DEMANDES MULTIPLES

Dans certains problèmes, les véhicules représentent un facteur de production (engins de chantiers, machinerie...) et peuvent être transportés d'un lieu de production (chantier, mine,...) à un autre. Nous utiliserons la terminologie engins pour facteurs de production et chantiers pour lieu de production. Afin de rencontrer un programme de production, chaque chantier a une demande pour chaque type d'engins, exprimée en engins-jours, qui doit être remplie à l'intérieur de certains intervalles de temps. Les frais de transport d'un chantier à l'autre sont calculés pour chaque type d'engins, et l'on n'envisage que les déplacements qui peuvent s'effectuer la nuit, à la fois pour des contraintes routières ou de main-d'oeuvre.

Pour aborder ce problème, nous définissons pour chaque type d'engins un réseau multiparti dans lequel chaque phase t représente un jour de travail. Les noeuds (i,t) sont définis pour les chantiers x_i lorsqu'il est possible d'entreposer ou d'utiliser les engins sur ce chantier le jour t. Les arcs ((i,t),(j,t+1)) sont définis lorsque le transport de x_i vers x_j est possible, et sa longueur représente le coût du transport. On introduit aussi les arcs ((i,t), (i,t+1)) de longueur nulle. Un chemin c dans ce réseau représente la suite des affectations d'un engin. Nous définirons le vecteur associé a_c, dans lequel a_{ci} représente le nombre de jours de travail effectué par cet engin sur le chantier x_i; un noeud (i,t) traversé par c contribuera à a_{ci} seulement si t est à l'intérieur d'un intervalle pendant lequel l'engin peut travailler sur le chantier x_i. La formulation par programmation linéaire du problème est semblable à (PL), mais le second membre $\underline{1}$ est remplacé par un vecteur d, où d_i représente la demande (en engins-jours) du chantier x_i.

ANNEXE 1: Pseudo-couplage

Au cours d'une expérimentation préliminaire, aussi bien lors des itérations du simplexe que de sous-gradient, est apparu le phénomène suivant: pendant les premières itérations (disons approximativement les n premières itérations), les 0-circuits engendrés comportent seulement deux sommets intermédiaires distincts x_i et x_j et bouclent sur les arcs (x_i,x_j) et (x_j,x_i). Si n est pair, ces 0-circuits sont de la forme 0-i-j-i-j-...-i-j-0; si n est impair, ils sont de la forme 0-i-j-i-j-...-i-j-i-0.

Pour éviter le gaspillage de temps consacré à la recherche de ces 0-circuits triviaux, on y consacrera une phase d'initialisation, appelée "pseudo-couplage", dans laquelle l'ensemble C sera réduit à l'ensemble C^2 des 0-circuits de ce type. A la fin de cette phase, on disposera de pénalités ¶ (et, pour le simplexe, d'une base réalisable de départ), telles que, dans la suite, ces 0-circuits n'aient plus qu'une faible chance d'apparaître.

1 - Cas n pair

Si n est pair, la longueur du 0-circuit C = 0-i-j-i-j-...-i-j-0 est

$$\ell_{ij} = d_{oi} + \tfrac{1}{2}nd_{ij} + (\tfrac{1}{2}n-1)d_{ji} + d_{jo}$$

et son vecteur représentatif a_c est défini par

$$a_{ci} = a_{cj} = \tfrac{1}{2}n \quad \text{et} \quad a_{ck} = 0 \quad \text{pour} \quad k \neq i \text{ et } k \neq j$$

Le 0-circuit c' symétrique de c est c' = 0-j-i-j-i-...-j-i-0; il a le même vecteur associé a_c mais sa longueur est

$$\ell_{ji} = d_{oj} + \tfrac{1}{2}nd_{ji} + (\tfrac{1}{2}n-1)d_{ij} + d_{io}$$

On peut alors considérer que seul apparaîtra celui des deux 0-circuits c et c' qui a la plus petite longueur, puisque tous deux ont le même vecteur associé. On peut alors définir, pour la paire {i,j}, non ordonnée:

$$p_{ij} = \text{Min}\{d_{oi}+d_{ij}+d_{jo} \; ; \; d_{oj}+d_{ji}+d_{io}\} + (\tfrac{1}{2}n-1)(d_{ij}+d_{ji})$$

et la restriction à C^2 de (PL) devient:

$$\text{Min} \sum_{i,j} p_{ij}x_{ij}$$

$$\text{s.t.} \sum_{j\neq i} \tfrac{1}{2}nx_{ij} = 1 \qquad \text{pour } i = k,2,\ldots,n$$

$$x \geqslant 0$$

Par le changement de variable $X = \tfrac{1}{2}nx$, et en posant $P = \frac{2}{n}p$, on obtient:

$$\text{Min} \sum_{i,j} P_{ij} X_{ij}$$

$$\sum_{j \neq i} X_{ij} = 1$$

$$X \geq 0$$

Ce problème est la relaxation continue du problème du couplage de poids minimal, dans le graphe complet, non orienté, muni des poids P. On peut remarquer que ce problème est équivalent au problème du couplage fractionnaire de poids maximal (changer les poids en $P'_{ij} = Q - P_{ij}$ où Q est une constante, $Q \geq \underset{i,j}{\text{Max}} P_{ij}$ pour assurer $P' \geq 0$). L'algorithme du simplexe a été adapté par E. Johnson ([17], chap. 7 et 8) à ce problème, et chaque itération consiste en manipulations élémentaires sur le graphe donné.

2 - Cas n impair

Dans le cas où n est impair, la longueur du 0-circuit

c = 0-i-j-i-j-...-i-j-i-0 est

$$\ell_{ij} = d_{oi} + \tfrac{1}{2}(n-1)(d_{ij}+d_{ji}) + d_{io}$$

et son vecteur associé a_c est défini par

$$a_{ci} = \tfrac{1}{2}(n-1)+1 \ , \ a_{cj} = \tfrac{1}{2}(n-1) \quad \text{et} \quad a_{ck} = 0 \quad \text{pour} \quad k \neq i \text{ et } k \neq j$$

Le problème ne peut pas être ramené à un problème symétrique (non-orienté). Toutefois, la restriction de (PL) à C^2 à la propriété suivante:

chaque colonne a exactement deux termes non-nuls. On peut donc le formuler comme un "problème de flot avec gains" ([17], chap. 7), pour lequel l'algorithme du simplexe prend une forme particulièrement simplifiée.

3 - Résultats

La phase d'initialisation décrite ici a été employée dans tous les problèmes de taille $n \geq 10$. Pour les problèmes de taille $n = 10$, quelques comparaisons ont fait ressortir les points suivants:

(i) la phase d'initialisation requiert approximativement le temps d'une itération de plus court chemin.

(ii) pour obtenir une solution de même valeur en partant de $\P^o = 0$ ou de la "phase 1" du simplexe, il est nécessaire d'utiliser de l'ordre de n itérations de plus court-chemin.

(iii) pour $n = 10$, on économise environ la moitié du temps de calcul et cette proportion augmente avec n.

Dans l'énumération implicite exposée au chapitre 4, il n'est nécessaire d'utiliser cette phase d'initialisation que dans le problème initial, les valeurs de départ de $\P$ indiquées pour les sous-problèmes étant assez bonnes pour que de nouvelles initialisations ne soient pas nécessaires.

ANNEXE 2: Plus courts-chemins sans va-et-vient

Dans l'Annexe 1, il a été signalé que les 0-circuits consistant uniquement en un va-et-vient entre deux sommets intermédiaires sont les plus fréquents dans les premières itérations, et un procédé pour surmonter ce phénomène a été décrit. Dans les itérations suivantes apparaissent des 0-circuits contenant plus de deux sommets intermédiaires, mais effectuant encore des va-et-vient, c'est-à-dire dans lesquels le $k^{ème}$ et le $(k+2)^{ème}$ sommet sont identiques, pour certains k. Il est possible, au prix d'une augmentation raisonnable du temps de calcul à chaque itération, d'éviter de tels 0-circuits.

1 - Une relaxation plus fine du problème du Voyageur de Commerce

Soit un 0-circuit $c = 0\text{-}i_1\text{-}i_2\text{-}\ldots\text{-}i_{n-1}\text{-}i_n\text{-}0$ dans R. Par définition c sera dit "sans va-et-vient" si

$$i_k \neq i_{k+2} \quad \text{pour tout } k = 1,2,\ldots,n-2$$

On désignera par C' l'ensemble de ces 0-circuits sans va-et-vient. Evidemment $C' \subset C$. Si R est un réseau complet alors

$$|C'| = n(n-1)(n-2)^{n-2} \quad \text{et} \quad \frac{|C|}{|C'|} \# 2.718....$$

Puisque tout tour dans R est dans C', le problème (PL') obtenu en restreignant à C' l'ensemble des variables de (PL), est une relaxation du

Problème du Voyageur de Commerce, et cette relaxation est plus fine que (PL) c'est-à-dire

$$\overline{Z} \leqslant \overline{Z}' \leqslant \ell_T$$

où $\overline{Z}$ est la valeur optimale de (PL)

$\overline{Z}'$ celle de (PL')

ℓ_T la longueur minimale d'un tour.

De la même manière, pour tout vecteur $\P$, si l'on définit

$$W'(\P) = \underset{c \in C'}{\text{Min}}\{\ell_c - \P.a_c\}$$

alors, pour tout $\P$ tel que $\P.\underline{1} = 0$, on a

$$W(\P) \leqslant W'(\P) \leqslant \ell_T$$

Dans l'énumération implicite décrite au chap. 4, on peut ainsi disposer d'une borne plus serrée sur la longueur du tour optimal. Par conséquent on explorera en général moins de sous-problèmes et cette réduction compense largement le temps supplémentaire consacré à chaque évaluation.

2 - Algorithme PCC2

L'algorithme PCC2, que nous allons décrire, permet de trouver un 0-circuit sans va-et-vient de plus petite longueur. Il utilise deux fonctions récurrentes ℓ et ℓ', et deux marquages associés m et m' indiquant les

antécédents de chaque noeud, ainsi qu'un tableau auxilliaire L. Le rôle des deux fonctions ℓ et ℓ' sera explicité par les assertions servant de justification à l'algorithme, et énoncées après celui-ci. Enfin, pour simplifier l'énoncé de PCC2, le réseau R sera supposé complet.

2.1 Définition de l'algorithme PCC2

1 initialisation (phase t = 2)

pour chaque i = 1,...,n faire:

a - $\ell(i,2) = \underset{j}{\text{Min}}\{d_{oj} + d_{ji} \mid j \neq i\}$

$m(i,2) = j$ réalisant ce minimum

b - $\ell'(i,2) = \underset{k}{\text{Min}}\{d_{ok} + d_{ki} \mid k \neq i \text{ et } k \neq m(i,2)\}$

$m'(i,2) = k$ réalisant ce minimum

2 pas général (phases t = 3,...,n)

pour chaque t = 3 jusqu'à n faire 2-1 et 2-2

2-1 définition du tableau auxilliaire

pour chaque i et chaque j, i ≠ j, définir:

$L(i,j) = \ell'(j,t-1) + d_{ji}$ si $i = m(j,t-1)$

$L(i,j) = \ell(j,t-1) + d_{ji}$ sinon

2-2 calcul de ℓ et ℓ'

pour chaque i = 1,...,n faire

a - $\ell(i,t) = \underset{j}{\text{Min}}\{L(i,j) \mid j \neq i\}$

$m(i,t) = j$ réalisant ce minimum

b - Si $t \leqslant n-1$ faire

$\ell'(i,t) = \underset{k}{\text{Min}}\{L(i,k) \mid k \neq i \text{ et } k \neq m(i,t)\}$

$m'(i,t) = k$ réalisant ce minimum

3 dernier pas (phase $t = n+1$)

calculer $\ell_c = \underset{i}{\text{Min}}\{\ell(i,n) + d_{io} \;/\; i = 1,\ldots,n\}$

Cet algorithme calcule la longueur minimum d'un O-circuit sans va-et-vient. Pour identifier un tel circuit, on utilisera les marques m et m' par "retour arrière":

i_n = i réalisant le minimum définissant ℓ_c

$i_{n-1} = m(i,n)$

pour chaque $t = n-2, n-3, \ldots$ jusqu'à 1 poser $i = i_{t+1}$ et

$i_t = m(i,t+1)$ si $m(i,t+1) \neq i_{t+2}$

$m'(i,t+1)$ sinon

Exemple: reprenons l'exemple 1 tiré de Wagner

1 - initialisation t = 2

i	1	2	3	4
$\ell(i,2)$	12	26	26	11
m(i,2)	4	1	1	1
$\ell'(i,2)$	18	27	29	20
m'(i,2)	2	4	4	2

2 - phase t = 3

i	j	1	2	3	4
1		∞	35	43	30
2		37	∞	50	36
3		37	46	∞	38
4		28	36	41	∞

tableau L

i	1	2	3	4
$\ell(i,3)$	30	36	37	28
m(i,3)	4	4	1	1
$\ell'(i,3)$	35	37	38	36
m'(i,3)	2	2	4	2

phase t = 4

i	j	1	2	3	4
1		∞	44	52	46
2		55	∞	61	53
3		55	56	∞	55
4		45	47	52	∞

tableau L

i	1	2	3	4
$\ell(i,4)$	44	53	55	45
m(i,4)	2	4	lou4	1

3 - phase t = 5

i	1	2	3	4
$\ell(i,4) + d_{io}$	54	62	65	53

Le 0-circuit sans va-et-vient recherché a pour longueur 53.

Son dernier sommet est $i_4 = 4$ et $i_3 = m(4,4) = 1$.

Puisque $m(1,3) = 4$, $i_2 = m'(1,3) = 2$.

Puisque $m(2,2) = 1$, $i_1 = m'(2,2) = 4$

le 0-circuit recherché est donc 0-4-2-1-4-0.

2.2 Justification: l'algorithme PCC2 est justifié par les assertions récurrentes suivantes:

assertion 1: pour tout $t = 2,\ldots,n$

$\ell(i,t)$ est la longueur minimum d'un chemin

$0\text{-}i_1\text{-}i_2\text{-}\ldots\text{-}i_{t-1}\text{-}i$ dans le réseau multiparti, joignant (0,0) à (i,t) et tel que

$$i_k \neq i_{k+2} \quad \text{pour tout } k = 1,\ldots,t-2 \qquad (1)$$

Un tel chemin peut s'obtenir en posant $i_{t-1} = m(i,t)$ et en appliquant l'assertion 3 si $t \geqslant 3$.

assertion 2: pour tout $t = 2,\ldots,n$

$\ell'(i,t)$ est la longueur minimum d'un chemin

$0\text{-}i_1\text{-}i_2\text{-}\ldots\text{-}i_{t-1}\text{-}i$ dans le réseau multiparti, joignant (0,0) à (i,t), vérifiant la relation (1) et la relation suivante:

$$i_{t-1} \neq m(i,t) \qquad (2)$$

Un tel chemin peut s'obtenir en posant $i_{t-1} = m'(i,t)$ et en appliquant l'assertion 3 si $t \geqslant 3$.

assertion 3: un chemin minimal $0\text{-}i_1\text{-}i_2\text{-}\ldots\text{-}i_{t-2}\text{-}j\text{-}i$ défini dans l'assertion 1 (avec $j = m(i,t)$) ou dans l'assertion 2 (avec $j = m'(i,t)$), peut s'obtenir de la manière suivante:

si $m(j,t-1) \neq i$, alors prendre pour $0\text{-}i_1\text{-}i_2\text{-}\ldots\text{-}i_{t-2}\text{-}j$ un chemin de longueur $\ell(j,t-1)$ défini dans l'assertion 1 (en y remplaçant t par t-1)

si $m(j,t-1) = i$, alors $0\text{-}i_1\text{-}i_2\text{-}\ldots\text{-}i_{t-2}\text{-}j$ est un chemin de longueur $\ell'(j,t-1)$ défini dans l'assertion 2 (en y remplaçant t par t-1)

Les assertions 1 et 2 sont évidemment valides pour $t = 2$; la validité des assertions 1,2,3 pour t+1 découle directement de la validité des assertions 1 et 2 pour t. Enfin pour $t = n+1$, le pas 3 de l'algorithme PCC2 fournit la longueur du 0-circuit recherché.

2.3 Nombre d'opérations élémentaires

L'exécution des instructions a et b dans les phases $t = 1$, et $t = 2$ jusqu'à n-1 requiert la recherche des deux plus petits éléments d'une liste de n-1 éléments. Ces deux opérations peuvent s'effectuer simultanément en

$$(n-1) + \lceil \mathrm{Log}_2(n-1) \rceil - 2 \text{ comparaisons} \qquad (2)$$

(ici $\lceil x \rceil$ désigne le plus petit entier $k \geq x$)

Pour la description d'algorithmes qui réalisent (2) comme nombre exact ou moyen de comparaisons voir [1], [20].

Dans le cas où R est complet, le compte des opérations donne:

1 - initialisation (t = 2)

pour chaque i	(n-1)	additions
	$n + \lceil \mathrm{Log}_2(n-1) \rceil - 3$	comparaisons

2 - pas général

2.1 pour t=3 jusqu'à n	n^2	additions
2.2 pour t=3 jusqu à n-1		
pour chaque i	$n + \lceil \mathrm{Log}_2(n-1) \rceil - 3$	comparaisons
pour t = n		
pour chaque i	n-2	comparaisons

3 - dernier pas	n	additions
	n-1	comparaisons

Soit en tout	$n^2(n-1)$	additions
	$n(n-2)(n + \lceil \mathrm{Log}_2(n-1) \rceil - 2) + n-1$	comparaisons
soit, en ordre de grandeur	$O(n^3)$	additions
	$O(n^3 + n^2 \mathrm{Log}_2 n)$	comparaisons

On peut comparer ces performances à celles de l'algorithme de Saigal (voir chap. 1): pour $n \geqslant 32$ l'augmentation du nombre d'additions est inférieure à 3.2%, et l'augmentation du nombre de comparaisons inférieure à 13%. Pour $n \geqslant 51$ ces proportions tombent à 2% et 10% respectivement.

2.4 Remarques et extensions

1 - La restriction $j \neq i$ peut être traitée sans nécessiter de comparaisons (on introduirait alors $O(n^3)$ comparaisons supplémentaires)

-soit en définissant $d_{ii} = \infty$ (ou un nombre M suffisamment grand pour assurer qu'un plus court chemin ne contienne aucun arc d_{ii})

-soit en associant à chaque sommet i de R une liste de ses antécédents.

Cette dernière solution doit être adoptée si R n'est pas complet. Si R a m arcs, le nombre d'opérations élémentaires dans PCC2 est de l'ordre de

$$O(mn^2) \quad \text{additions}$$

$$O(mn^2 + n^2 \lceil \text{Log}_2 \, d^-_{Max} \rceil) \quad \text{comparaisons}$$

où d^-_{Max} est le maximum des degrés intérieurs dans R

$(n-1 > d^-_{Max} > \frac{m}{n})$

2 - L'instruction 2.1 peut s'effectuer sans comparaisons et en n^2 additions, en définissant L colonne par colonne:

pour j = 1 jusqu'à n

1 pour i = 1 jusqu'à n $(i \neq j)$ définir

$$L(i,j) = \ell(j,t-1) + d_{ji}$$

2 poser $L(m(j,t-1),j) = \ell'(j,t-1) + d_{j,m(j,t-1)}$

3 - Au cours de l'énumération implicite, l'algorithme PCC2 s'adapte à chaque sous-problème avec une seule modification:
dans les phases t = n et t = n+1, restreindre i à l'ensemble des sommets qui ne sont pas origine d'un arc interdit.

4 - Les O-circuits engendrés par PCC2 ont tendance à contenir des "triangles", c'est à dire à être tels que

$$i_j = i_{j+3} \quad \text{pour certains } j = 1,\ldots,n-3$$

Il paraît naturel d'essayer de définir des "O-circuits sans triangles". Et plus généralement des "O-circuits sans k-boucles", c'est à dire tels que

$$i_j \neq i_{j+2},\ldots,i_j \neq i_{j+k} \quad \text{pour tout } j = 1,\ldots,n-k.$$

Naturellement, pour k = n-1, les seuls O-circuits possibles sont les tours. Pour $k \geq 3$ le problème parait très difficile et les auteurs posent la conjecture suivante:

Conjecture: le problème de déterminer un "O-circuit sans k-boucle" de longueur minimale dans un réseau R est NP-complet au sens de Karp ([1], [19]) pour $k \geq 3$.

ANNEXE 3: Restriction

Un inconvénient important de l'approche basée sur les 0-circuits réside dans l'ordre de complexité $O(n^3)$ du temps de calcul d'un 0-circuit minimal. Dans le cas des distances symétriques, l'approche par "1-arbre de poids minimum" de Held et Karp ([13], [14]) utilise des sous-problèmes de complexité $O(n^2)$; comme il n'est pas apparu que les bornes fournies par les 0-circuits soient sensiblement meilleures que celles fournies par les 1-arbres, cette différence de complexité entraîne que l'approche de Held et Karp devra être préférée pour les problèmes à distances symétriques.

Tenter de réduire la complexité du temps de calcul d'un 0-circuit minimal a fait l'objet d'un effort de recherche particulier de la part des auteurs, et a abouti au choix d'une stratégie de restriction qui est décrite plus loin. Auparavant, il semble intéressant de signaler une direction de recherche à laquelle beaucoup d'efforts ont été consacrés sans succès, mais qui ne semble pas encore définitivement close. Dans un rapport précédent [26], il était indiqué que l'application de l'algorithme de Dijkstra permettait de définir un algorithme de complexité $O(n^2 \text{Log}_2 n)$, mais ceci n'a pas pu être confirmé et aucun algorithme d'une telle complexité n'a pu être défini. Quatre versions différentes de l'algorithme de Dijkstra, de complexité $O(n^3)$ ou $O(n^3 \text{Log}_2 n)$ ont été programmées et testées sur le réseau multiparti; la plus efficace, basée sur une méthode de tri dite "bucket sorting" ([1]), s'est avérée encore nettement plus lente (environ trois fois pour n = 10) que l'algorithme de Saigal. Dû à la régularité du réseau multiparti, il

demeure possible qu'il existe un algorithme de complexité $O(n^2 Log_2 n)$ pour ce problème, mais ceci reste encore une conjecture.

La "stratégie de restriction" qui va être décrite, permet de réduire par un facteur constant le temps consacré à chaque itération de O-circuit minimal, en restreignant à un sous-ensemble des arcs de R.

1 - Réseau restreint

La stratégie de restriction exposée ici part de l'observation suivante: le tour optimal dans R emprunte plutôt des arcs courts du réseau et tend à éviter les arcs longs. En chaque sommet s_i, le tour optimal doit utiliser un arc d'extrémité x_i et un arc d'origine x_i. On définira alors le sous-ensemble E_k des arcs (x_i, x_j) de R tels que:

(x_i, x_j) est l'un des k plus courts arcs d'extrémité x_j

ou bien l'un des k plus courts arcs d'origine x_i

(k est un entier compris entre 1 et n-1)

Le réseau restreint est $R_k = (V, E_k, d)$ où d est restreint à E_k.

Remarque: afin de simplifier le calcul des O-circuits minimaux empruntant les arcs dans E_k, on a convenu que E_k contient tous les arcs d'origine ou d'extrémité x_o. Ceci n'enlève en fait rien à la généralité de ce qui suit.

Evidemment on a $E_1 \subseteq E_2 \subseteq \ldots \subseteq E_{n-2} \subseteq E_{n-1} = E$. De plus le tour optimal appartiendra probablement à E_k pour une valeur assez faible de k.

Exemple 1: tiré de [18] avec n = 32, symétrique:

E_3 contient toutes les arêtes du tour optimal, sauf l'arête (5,7)

E_4 contient les 33 arêtes du tour optimal.

Exemple 2: Dans ce contrexemple, le tour optimal contient la plus longue arête du réseau, donc seul l'ensemble E_{n-1} (= E) contient le tour optimal. Sur un demi-cercle d'extrémité x_{n-1} et x_n (voir Fig. 1), sont répartis n-1 sommets $x_o, x_1, \ldots, x_{n-2}$ dans cet ordre, en partant de x_n vers x_{n-1}.

Le tour optimal est 0-1-2-...-(n-2)-(n-1)-n-0 (voir [3, théorème 3]).

Les sous-ensembles E_k peuvent être définis après avoir trié par ordre de longueur croissante, pour chaque sommet x_i, d'une part les arcs d'origine x_i et d'autre part les arcs d'extrémité x_i. Si R est complet, chaque tri de (n-1) arcs peut s'effectuer en $O((n-1).Log_2(n-1))$ comparaisons (voir [1]) et l'ensemble des 2n tris peut se faire en $O(2n^2 Log_2 n)$ comparaisons. Si les distances sont symétriques, ce travail est réduit de moitié.

Le nombre d'arcs dans E_k dépend de la manière dont on tranche les cas d'égalité. Si l'on suppose que les distances sont toutes différentes (par exemple choix lexicographique en cas d'égalité) alors le nombre $|E_k|$ d'arcs dans E_k, en ne tenant pas compte des arcs incidents à x_o, est compris entre k_n et $2k_{n-1}$ (remarquons que l'arc le plus court figure pour ses deux extrémités). Généralement, ce nombre est assez proche de k_n:

dans l'exemple 1 (n = 32)

pour k = 3 $|E_3|$ = 112 soit 16.66% au-dessus du minimum

pour K = 4 $|E_4|$ = 152 soit 18.75% au-dessus du minimum.

En utilisant une représentation de R_k par des listes d'adjacences, on ramène la complexité du temps de calcul d'un O-circuit minimal à $O(|E_k|n)$, soit une réduction par un facteur compris entre $\frac{n}{2k}$ et $\frac{n}{k}$.

Dans l'exemple 1, ce facteur de réduction est

8.85 pour k = 3

6.52 pour k = 4

On peut également utiliser l'algorithme PCC2 sur le réseau restreint R_k, et la réduction dans le temps de calcul est comparable. C'est cette version, désignée par PCCR2, qui a été utilisée dans l'exemple illustrant le paragraphe 3.2.

2 - Stratégie de restriction

Tant que k < n-1, rien ne garantit la validité de la borne inférieure fournie par les itérations de sous-gradient sur le réseau R_k. Il faut donc que les dernières évaluations de $\overline{W}$, pour être valides, proviennent de réseau R initial. Néanmoins, les premières itérations, effectuées pour des valeurs faibles de k, fourniront de manière économique des pénalités assez proches des pénalités optimales pour le réseau R.

Une stratégie de restriction peut être définie par une séquence croissante de r entiers:

$$k_1 < k_2 < \ldots < k_r = n-1$$

et peut être décrite de la manière suivante:

1 - Poser i = 0; soit π^0 une pénalité initiale
(par exemple π^0 peut être fournie par le procédé décrit dans l'annexe 1, ou bien déduit des pénalités du problème antérieur dans l'arbre d'énumération, voir Chap. 4)

2 - Remplacer i par i+1 et appliquer des itérations de sous-gradient sur R_{k_i} en partant de π^{i-1} jusqu'à obtenir des pénalités π^i satisfaisantes dans ce réseau.

3 - Si i = r terminer, sinon aller en 2.

Le problème de déterminer une bonne séquence $k_1,\ldots,k_r$ reste largement ouvert à l'expérimentation. Dans l'exemple (n = 32) nous avons utilisé la stratégie définie par r = 2 et $k_1 = 4$, $k_2 = 32$.

Remarquons qu'une telle stratégie de restriction peut être appliquée à la plupart des problèmes combinatoires pour réduire l'effort de calcul consacré à des sous-problèmes d'évaluation. Par exemple, on pourrait l'utiliser pour essayer d'améliorer l'algorithme de Held et Karp pour le Problème du Voyageur de Commerce symétrique; dans ce cas la recherche d'un 1-arbre de poids minimum est accélérée, et la réduction du temps de calcul paraît du même ordre que pour les 0-circuits.

Enfin signalons que la restriction peut être utilisée pour définir une méthode heuristique de solution: la résolution totale du problème (avec énumération implicite) peut être entreprise sur le réseau restreint R_k pour une valeur faible, mais raisonnable, de k. Ceci sera en général beaucoup

plus facile que pour le problème global (réduction du nombre de solutions réalisables, et du temps de calcul des évaluations), mais il reste à vérifier qu'une telle heuristique peut concurrencer les méthodes déjà existantes (par exemple [22], [23]).

Exemple: dans l'exemple 1, pour k = 3, le tour optimal contenu dans E_3 a pour longueur 10894, soit 0.3% de plus que l'optimum global (10861).

REFERENCES

[1] Aho A., Hopcroft J. and Ullman J.: "The Design and Analysis of Computer Algorithm".
Addison-Wesley, (1974).

[2] Bellmore M. et Malone J.: "Pathology of TSP Subtour-Elimination Algorithms".
Operations Research, Vol. 16, pp 538-558 (1968).

[3] Bellmore M. et Nemhauser G.L.: "The TSP: A Survey".
Operations Research, Vol. 16, pp 538-558 (1968).

[4] Berge C.: "Graphes et hypergraphes".
Dunod (1970).

[5] Christofides N.: "The Vehicle Routing Problem".
NATO, July 1974, Conference on Combinatorial Optimization, Paris (1974)

[6] Dantzig, Fulkerson and Johnson: "Solution of a Large Scale TSP".
ORSA 2 pp393-410 (1954).

[7] Edmonds and Johnson: "Matchings: a well-solved class of Integer Linear Programs".
Proc. of the Calgary Int. Conf. on Combinatorial Structures and Their Applications. pp89-92, Gordon and Breach (1970).

[8] Edmonds and Johnson: "Matching,Euler Tours and the Chinese Postman".
Math. Prog. 5-1 pp88-124 (1973).

[9] Garfinkel and Nemhauser: "Integer Programming". Wiley (1972).

[10] Geoffrion, : "Elements of Large Scale Mathematical Programming".
Man.Sc. 16, pp652-691 (1970).

[11] Gilmore P.C. et Gomory R.E.: "A L.P. approach to the Cutting Stock problem".
Operations Research I, vol.9, pp849-859 (1961).
II, vol.11, pp863-887 (1963).

[12] Golden B.: "Vehicle Routing Problems: Formulation and Heuristic Solution Techniques".
MIT, ORC 113 (1975).

[13] Held M. and Karp R.M.: "The TSP and Minimum Spanning Trees".
Operations Research, vol.18, pp1138-1162 (1970).

[14] Held M. and Karp R.M.: "The TSP and Minimum Spanning Trees", part 2,
Mathematical Programming, vol.1, pp6-25 (1971).

[15] Held, Karp and Wolfe: "Large-Scale Optimization and the Relaxation Method".
Proc. ACM, pp507-509 (1972).

[16] Held, Wolfe and Crowder: "Validation of the Subgradient Optimization".
Math Prog, 6 pp62-88 (1974).

[17] Johnson E.: "Programming in Networks and Graphs".
ORC 65.1, University of California - Berkeley (1965).

[18] Karg, L.L. and Thompson, G.L.:"A Heuristic Approach to Solving TSP".
Man.Sc. 10, pp225-248 (1964).

[19] Karp R.M.: "Reducibility among Combinatorial Problems".
in Miller R.M., and Thatches J.W. (Eds).
Complexity of Computer Computation, Plenium Press, NY (1972).

[20] Kislitsyn, S.S.: "On the selection of the k^{th} element of an ordered set by pairwise comparison".
Sibirsk. Math. Zh.5, pp557-564.

[21] Lasdon L.: "Optimization Theory for Large Systems".
McMillan, (1970).

[22] Lin S.: "Computer Solution of the TSP".
Bell Syst. Techn. J.44, pp2745, (1965)

[23] Lin and Kernighan: "An Effective Heuristic Algorith for the TSP."
Operations Research 21, p498, (1973).

[24] Nemhauser G.: "Dynamic Programming".
Wiley, 1966.

[25] Orloff C.S.: "A Fundamental Problem in Vehicle Routing".
Networks 4, pp35-64 (1974).

[26] Picard and Queyranne: "Le Problème du Voyageur de Commerce: Une formulation par la Programmation Linéaire".
EP75-7, Ecole Polytechnique, Montréal, (Février 1975).

[27] Rao M.R. et Zionts S.: "Allocation of transportation Units to Alternative Trips".
Operations Research, vol.16, pp52-63 (1968).

[28] Rockafellar R.T.: "Convex Analysis".
Princeton, 1970.

[29] Rosseel M.: "Comments on a Paper by R. Saigal".
ORSA 16, ppl232-1234, (1968).

[30] Roy B.: "Algebre moderne et Théorie des Graphes".
Dunod (1970).

[31] Saigal R.: "A constrained Shortest Route Problem". Operations Research, Vol.16, pp388-401 (1970).

[32] Simonnard: "Linear Programming". Prentice Hall (1966).

[33] Wagner H.M.: "Principles of Management Science". Prentice-Hall (1970).