REVUE FRANÇAISE D'AUTOMATIQUE, INFORMATIQUE, RECHERCHE OPÉRATIONNELLE. RECHERCHE OPÉRATIONNELLE

B. ROY

D. GALLAND

Énumération des chemins ε-minimum admissibles entre deux points

Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle, tome 7, n° V3 (1973), p. 3-20

<http://www.numdam.org/item?id=RO_1973__7_3_3_0>

R.A.I.R.O.
(7e année, septembre 1973, V-3, p. 3 à 20)

ENUMERATION DES CHEMINS ε-MINIMUM ADMISSIBLES ENTRE DEUX POINTS

par B. ROY (1) et D. GALLAND (2)

Résumé. — *Soit $G = (X, U)$ un graphe dont chaque arc $u \in U$ est doté d'une longueur $a(u) \in R$. Un chemin μ allant du sommet x_1 au sommet x_n est dit ε-minimum si sa longueur $a(\mu)$ diffère d'au plus ε % de celle du plus court chemin de G allant de x_1 à x_n. Un tel chemin sera admissible s'il vérifie en outre certaines contraintes portant sur la séquence de sommets et d'arcs qui le compose.*

Cet article décrit un algorithme permettant l'énumération de tous les chemins ε-minimum admissibles allant de x_1 à x_n pour une classe de contraintes d'admissibilité assez générale. Il repose sur une procédure du type P.S.E.S. (Procédure par Séparation et Evaluation en Séquence). Un exemple d'application ainsi que des précisions concernant un programme écrit pour C.D.C. 6600 et ses performances achèvent l'article.

1. LE PROBLEME ET SES APPLICATIONS

1.1. Introduction

Nombreux sont les travaux traitant (cf. notamment [2], [3], [7], [8 *bis*], [12], [13], [18], [19], [20]) de procédures permettant de construire successivement le 1er, le 2e, le $k^{\text{ième}}$ plus court chemin entre deux sommets d'un graphe G (dont les arcs sont dotés d'une longueur). Les algorithmes proposés procèdent généralement par itération sur k d'où une répétition de calculs de plus courts chemins ou de déviations, qui les rend assez coûteux dans certains cas (k élevé par exemple).

Une autre façon d'aborder l'étude des chemins entre deux sommets de G ayant une longueur voisine du minimum, consiste à énumérer ceux dont la longueur ne s'écarte pas du minimum de plus de ε %. Outre son caractère fréquemment plus opérationnel, cette seconde approche devrait conduire à des procédures plus économiques, au moins dans certains cas. En effet, l'ordre et le nombre des solutions importent peu, l'écart au minimum compte souvent davantage. De plus, l'acceptation ou le refus d'un chemin peuvent être décidés indépendamment des autres une fois connue la longueur du plus court.

(1) Université de Paris IX — Direction Scientifique METRA.
(2) Université de Paris IX.

Rares semblent être les travaux abordant le sujet de cette seconde façon. Ceux dont nous avons connaissance impliquent quelques restrictions sur le graphe G et ne sont pas complètement dégagés de l'approche classique par les k plus courts chemins (notamment par le rôle qu'ils font jouer au concept de déviation). Un algorithme dû à Plowden (cf. [11]) repose sur une exploration des déviations successives des chemins déjà trouvés, combinée à un mode de pénalisation de certains arcs ayant pour effet de rallonger artificiellement certains chemins; dès qu'un chemin devient trop long, l'exploration est interrompue; il n'échappe pas à la répétition de calculs de plus courts chemins.

L'algorithme décrit dans le présent article est une Procédure par Séparation (branch and bound, implicit enumeration,...) dont le principe aurait déjà été expérimenté (cf. Sakarovitch [19]). De façon plus précise, c'est une P.S.E.S. (Procédure par Séparation et Evaluation en Séquence, cf. [6], [15]) qui présente en outre les particularités suivantes :

— aucune restriction sur le graphe;

— calcul préliminaire unique des longueurs des plus courts chemins;

— possibilité de n'explorer qu'un sous-ensemble des chemins solutions défini par des contraintes additionnelles de forme assez générale;

— possibilité de transformer l'énumération en une optimisation selon un critère de forme également assez générale.

Avant de décrire en détail les fondements de cet algorithme (section 2), il nous faut préciser les notations tout en rappelant quelques définitions, puis introduire le concept de chemin ε-minimum admissible (§ 1.2 et 1.3); nous passerons également brièvement en revue (§ 1.4) quelques applications de cet algorithme. Nous terminerons l'article avec un exemple, en décrivant un programme réalisé sur C.D.C. 6600 et en précisant les caractéristiques et performances.

1.2. Notations et rappels de définitions

Soit $X = \{ x_1, ..., x_n \}$ un ensemble fini dont les éléments seront appelés *sommets*, et $U \subset X^2$ un ensemble de couples de sommets appelés *arcs*. Le couple $G = (X, U)$ est par définition le graphe considéré. A chaque arc u de U est associé un nombre réel $a(u)$ (positif, négatif ou nul) appelé *longueur* de l'arc u. Pour $u = (x_i, x_j)$, on pose : $a(u) = a_{ij}$.

Un *chemin* μ_{ij} de G allant de x_i à x_j, est par définition une suite d'arcs :

$$\mu_{ij} = (u_1, ..., u_p, u_{p+1}, ..., u_q)$$

de la forme :

$$u_1 = (x_i, x_{k1}), ..., u_p = (x_{k_{p-1}}, x_{k_p}),$$

$$u_{p+1} = (x_{k_p}, x_{k_{p+1}}), ..., u_q = (x_{k_{q-1}}, x_j)$$

Souvent, il sera plus commode de faire apparaître un chemin non comme une suite d'arcs mais comme une suite de sommets, c'est pourquoi l'on écrira aussi :

$$\mu_{ij} = (x_i, x_{k_1}, \ldots, x_{k_{p-1}}, x_{k_p}, \ldots, x_{k_{q-1}}, x_j).$$

Lorsque cette suite de sommets est sans répétition, le chemin est dit *élémentaire*. On définit enfin la *longueur* $a(\mu_{ij})$ *d'un chemin* μ_{ij} par :

$$a(\mu_{ij}) = a(u_1) + \ldots + a(u_q).$$

X étant fini, le nombre des chemins élémentaires de G allant de x_i à x_j l'est aussi; il s'en suit que la longueur λ_{ij} du plus court chemin élémentaire joignant x_i à x_j est toujours définie. Le calcul des nombres λ_{ij} peut s'effectuer de très nombreuses manières et ne présente aucune difficulté (cf. par exemple : [16], chap. VII, sect. B) pourvu que le graphe G soit sans circuit absorbant. Par *circuit absorbant* on entend un chemin dont les deux sommets extrêmes coïncident (circuit) et dont la longueur est négative (absorbant). L'absence de tels circuits permet en effet de calculer λ_{ij} en raisonnant sur l'ensemble de tous les chemins joignant x_i à x_j sans faire intervenir la restriction « être élémentaire » (le minimum calculé de la sorte n'étant plus défini, pour certains des couples x_i, x_j, lorsque G admet des circuits absorbants).

Dans ce qui suit nous ne ferons aucune hypothèse sur la présence ou l'absence de circuits absorbants, mais nous supposerons connus les nombres λ_{in} pour $i = 1, \ldots, n-1$. Par convention, x_1 et x_n désigneront les deux sommets privilégiés entre lesquels on se propose d'étudier les chemins dont la longueur soit voisine de celle λ_{1n} du plus court (sous-entendu élémentaire s'il y a des circuits absorbants). On dira qu'un chemin (élémentaire ou non) μ_{1n} est ε-*minimum* s'il est tel que :

$$a(\mu_{1n}) \leqslant \lambda_{1n} + \frac{\varepsilon}{100} |\lambda_{1n}| \tag{1}$$

C'est précisément à l'ensemble C_ε de ces chemins que l'on s'intéresse. Faisons observer que C_ε est nécessairement fini si G est sans circuit absorbant et sans circuit de longueur nulle. Lorsqu'il en est autrement, on ne s'intéressera en fait qu'au sous-ensemble fini C_ε^e formé des seuls chemins élémentaires de C_ε.

1.3. Chemins ε-minimum admissibles

Comme nous le verrons au paragraphe suivant, il est bien des problèmes à propos desquels seuls les chemins ε-minimum vérifiant certaines contraintes méritent intérêt. Dans la mesure où de telles contraintes font apparaître comme inadmissibles de nombreux chemins ε-minimum, il importe d'en tirer parti pour accélérer la procédure.

Seules les contraintes dites de *dimension* et *d'exclusion* définies ci-après seront prises en considération dans le cadre de cet article. Précisons néanmoins que l'algorithme proposé pourra aisément être adapté pour tenir compte d'autres contraintes.

Soient $d_1(u), ..., d_r(u)$ r fonctions à valeurs réelles positives ou nulles, définies sur U, et servant à apprécier ce que nous appellerons la *dimension* de l'arc u selon le 1er, ..., le $r^{\text{ième}}$ point de vue. Étant donné un chemin $\mu_{1n} = (u_1, ..., u_q)$ sa dimension selon le $\rho^{\text{ième}}$ point de vue sera définie par :

$$d_\rho(\mu_{1n}) = d_\rho(u_1) + ... + d_\rho(u_q) \qquad \rho = 1, ..., r$$

Soient enfin $d_1, ..., d_r$ r scalaires non négatifs, nous dirons que ce chemin satisfait aux *contraintes de dimensions* si :

$$d_\rho(\mu_{1n}) \leqslant d_\rho \qquad \rho = 1, ..., r \tag{2}$$

Supposons qu'à chaque sommet x on ait associé un sous-ensemble, éventuellement vide $E(x)$ de X. Étant donné un entier e, nous dirons qu'un chemin $\mu_{1n} = (x_{k_0}, ..., x_{k_q})(x_{k_0} = x_1$ et $x_{k_q} = x_n)$ satisfait aux *contraintes d'exclusion* définies par l'application E et de niveau e si :

avec

$$\left.\begin{aligned} &\sum_{j=0}^{p-1} \delta_j^p < e \qquad p = 1, ..., q \\ &\delta_j^p = 1 \qquad \text{si } x_{k_p} \in E(x_{k_j}) \\ &\delta_j^p = 0 \qquad \text{si } x_{k_p} \notin E(x_{k_j}) \end{aligned}\right\} \tag{3}$$

Dans certains cas on pourra être amené à introduire plusieurs applications E avec des niveaux différents.

Autrement dit, on interdit pour $p = 1, ..., q$ que le sommet x_{k_p} appartienne à au moins e ensembles $E(x_{k_j})$ pour $j = 0, ..., p-1$. Dans le cas où $e = 1$ la contrainte (3) revient à dire qu'aucun des sommets $x_{k_{j+1}}, ... x_{k_q}$ n'est élément de $E(x_{k_j})$ pour $j = 0, ..., q - 1$.

On appellera chemin ε-minimum admissible tout chemin de G allant de x_1 à x_n vérifiant les conditions (1), (2), (3). Nous noterons $\bar{C}_\varepsilon$ l'ensemble des chemins ε-minimum admissibles. Lorsque, pour des raisons données au § 1.2, on souhaite restreindre l'étude aux seuls chemins élémentaires, faisons observer que l'on pourra toujours le faire en raisonnant sur un ensemble de type $\bar{C}_\varepsilon$: il suffit de prendre la précaution de définir l'application d'exclusion E de telle sorte que $e = 1$ et que $x \in E(x)$ $\forall x \in X$, ou, si cela complique trop, d'en introduire une supplémentaire ayant ces caractéristiques.

Introduisons enfin la notion de chemin optimum, laquelle conduira à transformer l'algorithme d'énumération en algorithme d'optimisation. Sup-

posons pour cela définie sur U une fonction supplémentaire à valeurs non négatives notée $d_0(u)$ (dimension selon un point de vue référencé 0). Un *chemin optimum* sera par définition un chemin $\mu^*_{1n} \in \bar{C}_\varepsilon$ tel que :

$$d_0(\mu^*_{1n}) \leqslant d_0(\mu_{1n}) \; \forall \mu_{1n} \in \bar{C}_\varepsilon \tag{4}$$

1.4. Le problème et son intérêt

Que ce soit en gestion ou en économie appliquée, nombreux sont les problèmes concrets (renouvellement, trésorerie, ordonnancement, distribution, développement et croissance, transport et circulation...) qui font appel de façon explicite ou non, à l'étude de plus courts chemins. Dans ce genre de problème, il est assez rare que la longueur $a(u)$ affectée à chaque arc u du graphe G puisse être définie avec une grande précision. Cela tient soit à ce que l'évaluation précise de $a(u)$ serait trop coûteuse, soit à ce que la définition même de $a(u)$ prête à quelques ambiguïtés. C'est notamment le cas lorsque $a(u)$ représente un coût généralisé résultant d'une pondération artificielle de différents termes; de tels coûts sont couramment utilisés dans les modèles de circulation urbaine ou interurbaine (cf. Sakarovitch [18], R.A.T.P. [14]) pour expliquer la répartition des courants origine-destination entre divers itinéraires. Dans un cas comme dans l'autre, l'optimum (plus court chemin) défini sur de telles bases, n'a qu'une valeur illusoire et, pour être réaliste, il est opportun de faire intervenir dans l'étude tous les chemins dont la longueur n'est pas, compte tenu de la précision des données, significativement différente de celle du plus court. Ce sont précisément les chemins que nous avons appelés ε-minimum.

Toutefois, il n'y a pas que l'imprécision des données qui conduise à s'intéresser à de tels chemins. Dans certains problèmes (renouvellement, distribution, ordonnancement...), la longueur $a(u)$ sert à appréhender un critère (primordial) mais il en existe d'autres qu'il est essentiel de faire intervenir dès l'instant où la performance atteinte sur le critère longueur est suffisamment bonne. On pourra alors être conduit à rechercher l'optimum sur un critère secondaire (chemin ε-minimum optimum). En programmation dynamique, l'étude déjà fort ancienne (voir par exemple Kaufman et Cruon [8]) des politiques dites sous-optimales, relève précisément de telles préoccupations.

Sans trop entrer dans le détail des applications, nous voudrions insister sur le fait qu'une procédure d'énumération des chemins ε-minimum (optimum ou non) n'aura un réel intérêt pratique que dans la mesure où elle s'accomode facilement de contraintes additionnelles venant restreindre l'ensemble des chemins auxquels on s'intéresse. Les contraintes de type (2) et (3) introduites en 1.3 ne sont évidemment pas les seules intéressantes en pratique; mais, outre le fait que la procédure décrite ci-après peut aisément en intégrer d'autres, celles-ci sont susceptibles de nombreuses interprétations que nous allons brièvement esquisser.

Selon la définition attribuée à la longueur $a(u)$, les dimensions $d_0(u), \ldots, d_r(u)$ peuvent être utilisées pour comptabiliser des longueurs physiques, des durées de trajet, des coûts de transfert..., l'une de ces dimensions pouvant intervenir comme critère secondaire. Dans les problèmes d'ordonnancement, par exemple, la longueur $a(u)$ traduit la durée minimum qui doit séparer les instants auxquels prennent place les événements (débuts de tâches) symbolisés par les extrémités de l'arc u. L'étude des ordonnancements (calendriers de la famille d'événements symbolisés par X) est étroitement liée (cf. Roy [16], chap. VIII, sect. C) à celle des chemins joignant dans G les sommets correspondants à deux événements particuliers : « début » et « fin ». La durée globale (intervalle de temps qui s'écoule entre les deux événements) est fréquemment un critère primordial. Sa minimisation implique parfois, compte tenu de l'imprécision des données, l'examen de chemins ε-minimum. De plus, le critère durée globale n'est souvent pas le seul (cf. Roy [16], chap. VIII, sect. D, § 1 et 2) et l'on est amené dans certains cas à introduire des contraintes de dimension ayant trait à la trésorerie ou à l'emploi de la main-d'œuvre.

Dans des problèmes d'une autre nature (routage de marchandises, de flux postaux ou téléphoniques,...) on peut vouloir dénombrer, afin d'en limiter le nombre sur chaque chemin, les obstacles inhérents au franchissement d'arcs particuliers de diverses catégories. Il suffit pour cela d'introduire autant de dimensions $d_\rho(u)$ qu'il y a de catégories, $d_\rho(u)$ désignant le nombre d'obstacles portés par l'arc u dans la catégorie correspondante. C'est ainsi que dans l'étude d'un réseau de transport en commun, il sera facile de faire intervenir les obstacles que sont les correspondances (cf. exemple 3.3). De même on peut, en définissant des nombres $d_\rho(u)$ appropriés, dénombrer (pour en limiter le nombre) les sommets rencontrés dans diverses classes de sommets introduites *a priori* (sommets impliquant un transfert, un tri, une commutation, une transformation, ...).

Les contraintes d'exclusion interviennent chaque fois que le passage par au moins e sommets interdit le passage ultérieur par d'autres sommets (ceux communs aux ensembles d'exclusions correspondants). Comme exemple de cette contrainte, nous avons déjà évoqué les cas particuliers $e = 1$ et $x \in E(x)\ \forall\ x$. Citons un autre exemple : les nécessités de la modélisation conduisent fréquemment à « éclater un carrefour physique » en plusieurs sommets dans le graphe G (cf. exemple 3.3); un tel éclatement introduit généralement de nombreux chemins parasites qu'il faut éliminer. On parvient aisément, avec des contraintes d'exclusion de niveau 2, à ne s'intéresser qu'aux chemins qui, pénétrant dans un carrefour physique par un sommet x, en sortent obligatoirement après le sommet suivant x, et ce pour ne plus y revenir; il suffit de poser : $E(x) =$ ensemble des sommets appartenant au même carrefour physique que x.

Signalons encore que moyennant l'introduction d'un sommet supplémentaire sur chacun des arcs concernés on peut exprimer, grâce aux contraintes

d'exclusion, le fait que passer par un arc u exclut le passage ultérieur par certains arcs. Précisons enfin que, si le mot ultérieur employé à plusieurs reprises dans ces exemples permet d'exclure x suivi de y tout en tolérant y suivi de x, il est tout à fait facile d'interdire la présence sur un même chemin de x et de y quel que soit l'ordre d'apparition : on posera $y \in E(x)$ et $x \in E(y)$.

2. LES FONDEMENTS DE L'ALGORITHME

Nous commencerons par introduire l'arborescence des chemins de G issus du sommet x_1, nous dirons plus simplement l'arborescence des chemins, et par rappeler comment, une fois défini un ordre dit transverse, il en résulte un ordre complet sur l'ensemble des sommets de l'arborescence (ordre de Tarry). L'intérêt de cette arborescence apparaîtra au paragraphe 2.2 où l'on montrera qu'elle traduit un principe de séparation qui, appliqué à $\bar{C}_\varepsilon$ conformément à l'ordre de Tarry permet d'en énumérer tous les éléments sans omission ni répétition. Des tests d'évaluation sont toutefois nécessaires (cf. § 2.3) pour éviter d'explorer des branches de l'arborescence (autrement dit de poursuivre la séparation) alors que c'est inutile pour l'énumération ou la recherche d'un chemin optimum.

2.1. L'arborescence des chemins

Commençons par quelques rappels. Étant donné un graphe, on peut associer à chaque sommet x l'ensemble $S(x)$ de ses *suivants*, c'est-à-dire des sommets y tels que (x, y) soit un arc. La donnée de l'ensemble des sommets et de l'application S suffit à définir le graphe.

Soient Ω un ensemble (fini ou infini dénombrable) et Δ une application de Ω dans l'ensemble des parties finies de Ω. On définit ainsi un graphe χ ayant pour sommets les éléments ω de Ω, $\Delta(\omega)$ définissant l'ensemble des suivants de ω. χ est une *arborescence* de racine ω_1 si (cf. par exemple [16], tome 1, p. 339, th. V. 25, prop. (4)) : $\forall\, \omega \in \Omega$ il existe dans χ un chemin ([1]) et un seul allant de ω_1 à ω. Ce chemin sera désigné par $\gamma(\omega)$.

Considérons une règle (exemple : être plus à droite sur la figure) permettant de ranger selon un ordre complet les sommets de $\Delta(\omega)$ et ce $\forall\, \omega \in \Omega$. Nous parlerons alors d'*ordre transverse* sur l'arborescence χ. Étant donné un ordre transverse, Ω peut être doté d'un ordre complet τ, compatible avec lui et appelé *ordre de Tarry*. Par définition, τ prescrit de ranger le sommet ω' avant le sommet ω'' si (et seulement si), ω_d désignant le dernier sommet commun à $\gamma(\omega')$ et $\gamma(\omega'')$, l'une des deux conditions suivantes est remplie :

— $\omega_d = \omega'$;

(1) Même lorsque Ω est infini dénombrable un chemin reste ici défini par une suite nécessairement finie comme il a été dit plus haut à propos du cas fini.

— $\omega_d \neq \omega'$, $\omega_d \neq \omega''$ et la comparaison dans l'ordre transverse des deux éléments de $\Delta(\omega_d)$ que sont ω'_d suivant de ω_d sur $\gamma(\omega')$ et ω''_d suivant de ω_d sur $\gamma(\omega'')$, place ω'_d avant ω''_d.

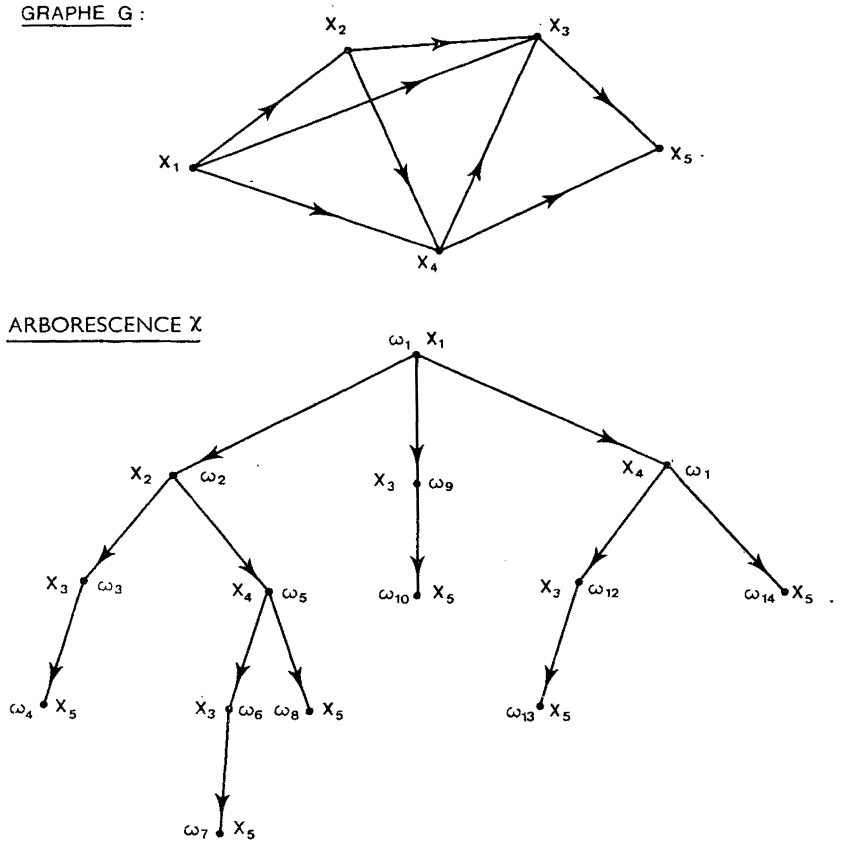


Figure 1
Arborescence des chemins

Revenons maintenant au graphe G du § 1.2 et à l'étude des chemins issus de x_1. Introduisons pour cela une arborescence $\chi = (\Omega, \Delta)$ dont chaque sommet ω est étiqueté par un sommet $x_{i_\omega} \in X$, celle-ci étant définie de façon récursive comme suit :

1° la racine de l'arborescence ω_1 est étiquetée x_1;

2° si $\omega \in \Omega$ étiqueté x_{i_ω},

alors : $\Delta(\omega)$ contient autant d'éléments que x_{i_ω} a de suivants dans G et leurs étiquettes définissent une bijection entre ces deux ensembles.

L'arborescence étiquetée χ ainsi définie est évidemment finie si G est sans circuit (cf. par exemple figure 1). Qu'elle soit finie ou non, il est clair que, à tout chemin μ issu de x_1 dans G correspond un sommet ω unique de Ω tel que la suite d'étiquettes portée par le chemin $\gamma(\omega)$ (joignant dans χ ω_1 à ω) soit précisément la suite de sommets définissant μ. Réciproquement, quel que soit $\omega \in \Omega$ la suite d'étiquettes portées par le chemin $\gamma(\omega)$ définit dans G un chemin issu de x_1. Il existe donc une bijection entre Ω et l'ensemble des chemins issus de x_1 dans G. Nous noterons $\mu(\omega)$ le chemin de G correspondant dans cette bijection au sommet ω de Ω.

L'indexation des sommets de G définit un ordre transverse sur Ω dans lequel, $\forall\, \omega \in \Omega$, les sommets de $\Delta(\omega)$ sont ordonnés conformément à l'ordre croissant des indices de leurs étiquettes. Faisons observer que l'ordre de Tarry associé à l'ordre transverse définit sur Ω un ordre complet lié de façon simple et évidente à l'ordre lexicographique des chemins de G issus de x_1.

2.2. La séparation en séquence

Soit $x_1, x_{k_1}, \ldots, x_{k_p}$ la suite des étiquettes associées à un chemin quelconque $\gamma(\omega)$ de χ. Un chemin de G sera dit *compatible* avec le chemin $\gamma(\omega)$ s'il commence par $x_1, x_{k_1}, \ldots, x_{k_p}$ (d'autres sommets pouvant ou non venir ensuite). Cette relation de compatibilité permet d'associer à chaque $\omega \in \Omega$ le sous-ensemble $\bar{C}_\varepsilon(\omega)$ de $\bar{C}_\varepsilon$ formé de tous les chemins ε-minimum admissibles de G compatibles avec $\gamma(\omega)$ ($\bar{C}_\varepsilon(\omega) = \varnothing$ lorsque de tels chemins n'existent pas). Revenons alors à l'application Δ définissant χ, et considérons un sommet quelconque ω^0 (de Ω) portant l'étiquette x^0. Soit :

$$\Delta[\bar{C}_\varepsilon(\omega^0)] = \{\, \bar{C}_\varepsilon(\omega)/\omega \in \Delta(\omega^0) \,\}$$

la famille de parties de $\bar{C}_\varepsilon(\omega^0)$ obtenue en *séparant* cet ensemble en autant de sous-ensembles (éventuellement vides) — dont la réunion reconstitue l'ensemble séparé — que x^0 a de suivants dans G.

Pour l'essentiel, la procédure consiste en une séquence de séparations du type ci-dessus conforme à l'ordre de Tarry introduit à la fin du paragraphe précédent. A l'initial (phase I de l'algorithme décrit au 3.1), c'est l'ensemble $\bar{C}_\varepsilon = \bar{C}_\varepsilon(\omega_1)$ qui est séparé et qui donne ainsi naissance à une suite ordonnée :

$$\bar{C}_\varepsilon(\omega_2),\ \bar{C}_\varepsilon(\omega_3),\ \ldots,\ \bar{C}_\varepsilon(\omega_{h_1})$$

de sous-ensembles rangés — nous dirons empilés (1) — conformément à l'ordre transverse : $\bar{C}_\varepsilon(\omega_2)$ étant sur le haut de la pile.

A ce stade, comme à l'issue de n'importe quelle séparation ultérieure, on ne sait rien (si ce n'est leur définition) des divers sous-ensembles engendrés contenus dans cette pile. Chaque fois qu'un sous-ensemble de la pile sera

(1) Par référence au concept informatique de *pile* utilisé dans l'algorithme.

sélectionné en vue d'être à son tour séparé, il fera, au préalable, l'objet d'une série d'évaluations à l'aide de tests (décrits dans le paragraphe suivant). Ces tests sont conçus de telle sorte que, dès l'instant où l'un d'eux est négatif, c'est la preuve que le sous-ensemble $\bar{C}(\omega^0)$ en cours d'évaluation peut être qualifié de *sondable* (cf. Hansen [6]), c'est-à-dire n'a pas lieu d'être séparé :

— soit parce qu'il est vide (ou, dans la variante d'optimisation, qu'il ne renferme aucun chemin optimum),

— soit parce que l'on a le moyen d'énumérer directement tous les chemins ε-minimum admissibles (ou optimum) qu'il renferme (cf. 3.1, phase T).

Faisons observer que, si $\bar{C}_\varepsilon(\omega^0)$ est sondable, il devient inutile d'examiner les sous-ensembles associés aux sommets de la sous-arborescence de χ ayant pour racine ω^0. Ces sommets, appelés *descendants* de ω^0 forment un ensemble $\Delta(\omega^0)$ dont les éléments sont par définition les sommets de χ extrémités d'un chemin issu de ω^0; le sous-ensembles de $\bar{C}_\varepsilon$ associé à chacun est donc nécessairement inclus dans $\bar{C}_\varepsilon(\omega^0)$.

Chaque fois qu'un sous-ensemble $\bar{C}_\varepsilon(\omega^0)$ donne lieu à évaluation (phase E de l'algorithme) il est retiré de la pile. Si tous les tests sont positifs, il est à son tour séparé (phase S de l'algorithme). Il faut entendre par là :

— la définition des sous-ensembles formant $\Delta[\bar{C}_\varepsilon(\omega^0)]$;

— lorsque $\Delta[\bar{C}_\varepsilon(\omega^0)] = \varnothing$ le renvoie en phase T pour énumération de tous les chemins ε-minimum admissibles (ou optimum) que renferme $\bar{C}_\varepsilon(\omega^0)$; nous verrons en 3.1 que, comme il est de règle dans toute Procédure par Séparation et Evaluation (P.S.E.), l'impossibilité de séparer un sous-ensemble implique effectivement la possibilité d'en énumérer les éléments « intéressants »;

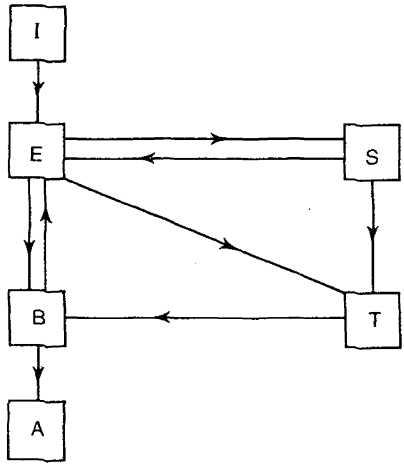
— lorsque $\Delta[\bar{C}_\varepsilon(\omega^0)] \neq \varnothing$ l'insertion sur la pile des sous-ensembles qu'il contient rangés conformément à l'ordre transverse, le premier selon cet ordre étant sur le dessus de la pile.

Comme dans toute P.S.E.S., du fait que la réunion des sous-ensembles qui sont substitués sur la pile à $\bar{C}_\varepsilon(\omega^0)$ n'est autre que $\bar{C}_\varepsilon(\omega^0)$ lui-même, et que cette substitution a toujours lieu sauf si $\bar{C}_\varepsilon(\omega^0)$ ne nécessite plus d'être examiné, la réunion des sous-ensembles formant la pile à un stade quelconque de la procédure n'est autre que la partie de l'ensemble des solutions initiales (ici $\bar{C}_\varepsilon$) qui, à ce stade, nécessite encore examen (autrement dit Evaluation et Séparation pour les sous-ensembles qui s'y trouvent).

Il ne reste plus maintenant qu'à préciser selon quelle règle se construit la séquence de séparation. Comme dans toute P.S.E.S. (cf. [15]) c'est chaque fois le sous-ensemble figurant sur le dessus de la pile qui doit être sélectionné comme sous-ensemble $\bar{C}_\varepsilon(\omega^0)$ à évaluer et le cas échéant à séparer. En effet $\bar{C}_\varepsilon(\omega^0)$ ainsi défini n'est autre (d'après le mode de constitution de la pile) que le premier sous-ensemble $\bar{C}_\varepsilon(\omega)$ qui vient, dans l'ordre de Tarry, après ceux

déjà évalués et parmi ceux qui nécessitent encore de l'être. Il s'en suit qu'en procédant de la sorte jusqu'à épuisement de la pile on est assuré d'exhiber au cours des phases T successives tous les chemins ε-minimum admissibles (ou optimum dans la variante d'optimisation). La procédure peut d'ailleurs être interrompue avant épuisement de la pile dès l'instant où le nombre de solutions déjà trouvées est jugé suffisant.

Faisons enfin observer que, dans l'algorithme 3.1 comme dans beaucoup d'autres du même type (cf. par exemple [1], [6], [17]) il est commode de suivre l'ordre de Tarry en introduisant une phase B (Back tracking) qui réduit considérablement le nombre des « échanges » avec la pile. Après chaque séparation, le premier élément de l'ordre transverse n'est jamais effectivement placé sur la pile (puisque c'est lui qui est immédiatement sélectionné pour évaluation); le recours effectif à la pile pour sélection n'a donc lieu qu'après un test négatif ou après exécution de la phase T.



I : Initialisation B : Backtracking
E : Evaluation T : Terminal
S : Séparation A : Arrêt

Figure 2

L'enchaînement des phases est résumé dans l'organigramme de la figure 2 (lequel est, à des détails secondaires près, commun à toutes les P.S.E.S.). On remarquera que le diagnostic des sous-ensembles dits *terminaux* c'est-à-dire ceux susceptibles d'être traités en phase T afin d'exhiber toutes les solutions « intéressantes » qu'ils contiennent, peut *a priori*, se faire en phase E et/ou en phase S. Ce sont des considérations d'ordre essentiellement informatique qui font opter pour l'une, l'autre ou les deux possibilités.

2.3. Les tests d'évaluation

Dans le paragraphe précédent nous avons situé l'algorithme dans le cadre d'une P.S.E.S. sans préciser le détail de la phase d'évaluation. Les tests qu'il convient d'y inclure sont étroitement liés à la notion de chemin ε-minimum admissible. Leur efficacité est d'une grande importance pratique car c'est d'eux que dépend la poursuite du développement de l'arborescence χ en chaque sommet ω lorsque l'on envisage de séparer l'ensemble $\bar{C}_\varepsilon(\omega)$.

Soient un ensemble $\bar{C}_\varepsilon(\omega^0)$, x^0 l'étiquette de ω^0 et $x_{k_0}, x_{k_1}, \ldots, x_{k_p}$ la suite des étiquettes associée au chemin $\gamma(\omega^0)$ de $\chi(x_{k_0} = x_1$ et $x_{k_p} = x^0)$. L'ensemble $\bar{C}_\varepsilon(\omega^0)$ sera soumis en phase E à trois types de tests correspondant respectivement aux contraintes (1) (2) et (3) introduites en 1.2 et 1.3.

Nous dirons que $\bar{C}_\varepsilon(\omega^0)$ satisfait au *test de longueur* si (cf. contrainte (1)) :

$$a(x_{k_0}, \ldots, x_{k_p}) + \lambda_{k_p n} \leqslant \lambda_{1n} + \frac{\varepsilon}{100} |\lambda_{1n}|$$

Nous disons que $\bar{C}_\varepsilon(\omega^0)$ satisfait aux *tests d'exclusion* (cf. contrainte (3) si :

$$\sum_{j=0}^{p-1} \delta_j < e \qquad \text{avec} \qquad \begin{cases} \delta_j = 0 & \text{si} \quad x_{k_p} \notin E(x_{k_j}) \\ \delta_j = 1 & \text{si} \quad x_{k_p} \in E(x_{k_j}) \end{cases}$$

et si :

$$\sum_{j=0}^{p} \beta_j < e \qquad \text{avec} \qquad \begin{cases} \beta_j = 0 & \text{si} \quad x_n \notin E(x_{k_j}) \\ \beta_j = 1 & \text{si} \quad x_n \in E(x_{k_j}) \end{cases}$$

Autrement dit, si le sommet x_{k_p} n'est pas exclu par au moins e des sommets $x_{k_0}, \ldots, x_{k_{p-1}}$ et si le sommet x_n n'est pas exclu par au moins e des sommets $x_{k_0}, \ldots, x_{k_p}$.

Enfin, nous dirons que $\bar{C}_\varepsilon(\omega^0)$ satisfait aux *tests de dimension* (1) suivant les r points de vue si (cf. contrainte (2)) :

$$d_\rho(x_{k_0}, \ldots, x_{k_p}) \leqslant d_\rho \qquad \rho = 0, \ldots, r$$

Dans le cas où l'on minimise la dimension suivant le point de vue référencé 0 (algorithme d'optimisation), d_0 n'est pas défini *a priori*. Pour appliquer le test on prendra comme valeur de d_0, une valeur très grande au départ, puis au cours des itérations, toujours la plus petite dimension (suivant ce point de vue 0) des chemins solutions déjà trouvés. Notons que dans ce cas s'il y a plusieurs chemins optimaux on n'en trouvera qu'un seul.

(1) Tout comme pour le test de longueur, on pourrait ajouter au premier membre du présent test la dimension minimum $\Delta_{k_p n}$ du chemin allant de x_{k_p} à x_n. Cela n'a d'intérêt que si ce minimum est généralement significativement différent de zéro.

L'évaluation d'un ensemble $\bar{C}_\varepsilon(\omega^0)$ sera positive s'il satisfait à tous les tests introduits, elle sera négative s'il ne satisfait pas à au moins un d'entre eux.

Cette évaluation appelle deux remarques :

— d'une part, si l'on a pris soin d'attribuer à λ_{in} une valeur très grande quand il n'y a pas de chemin du sommet x_i au sommet x_n, alors une évaluation positive pour un ensemble $\bar{C}_\varepsilon(\omega^0)$ garantit effectivement que $C_\varepsilon(\omega^0)$ n'est pas vide;

— d'autre part, si $x^0 = x_n$ une évaluation positive implique que le chemin $(x_{k0}, ..., x_n)$ satisfait aux contraintes (1) parce que $\lambda_{nn} = 0$, (2) et (3); c'est donc un chemin ε-minimum admissible appartenant à $\bar{C}_\varepsilon(\omega^0)$. Mais ce ne sera pas forcément le seul appartenant à cet ensemble car le sommet x_n peut appartenir à un ou plusieurs circuits.

3. L'ALGORITHME ET SA PROGRAMMATION

3.1. L'algorithme

La description qui suit explicite l'organigramme de la figure 2.

Phase I : Faire $\omega^0 = \omega_1$ (cf. 2.1) et aller à la phase *E*.

Phase E : Évaluer l'ensemble $\bar{C}_\varepsilon(\omega^0)$, c'est-à-dire effectuer sur lui les tests du 2.3. Si l'évaluation est positive aller à la phase *S* sinon aller à la phase *B*.

Phase S : Définir (cf. 2.2) les ensembles éléments de $\Delta(\bar{C}_\varepsilon(\omega^0))$, puis :

— si $x^0 = x_n$ mettre tous les ensembles ainsi définis sur la pile conformément à l'ordre de Tarry et aller à la phase *T*;

— sinon mettre ces ensembles sur la pile conformément à l'ordre de Tarry sauf le premier qui devient le nouvel ensemble $\bar{C}_\varepsilon(\omega^0)$ et aller à la phase *E*.

Phase T : Le chemin $\gamma(\omega^0)$ est ε-minimum admissible. Imprimer ce chemin sauf dans la variante d'optimisation où il suffit de conserver parmi les chemins déjà obtenus en phase *T* celui qui a la plus petite dimension d_0, cette dimension définissant le second membre du test de dimension correspondant dans la phase *E*. Aller à la phase *B*.

Phase B : Si la pile est épuisée aller à la phase *A*, sinon retirer le premier ensemble de la pile, l'appeler $\bar{C}_\varepsilon(\omega^0)$ et aller à la phase *E*.

Phase A : Fin de l'algorithme. Dans le cas de l'optimisation imprimer le dernier chemin conservé en phase *T*, c'est l'optimum.

3.2. Le programme MULTICHEMIN

L'algorithme décrit dans les paragraphes précédents a donné lieu à la réalisation d'un programme nommé MULTICHEMIN écrit en FORTRAN pour C.D.C. 6600. On a cherché à obtenir une grande souplesse d'utilisation et à

réduire l'occupation de mémoire, ce qui devrait permettre de résoudre des problèmes variés faisant intervenir des graphes assez importants.

Citons quelques caractéristiques de ce programme :

— Entrée très souple du graphe par le dictionnaire des suivants. Possibilité, quand le graphe est symétrique, de ne mentionner que la moitié des arcs.

— Possibilité de vérification des données sur le nombre de précédents de tout ou partie des sommets du graphe.

— Possibilité de suppression de sommets, de libellés pour les sommets et d'impression du dictionnaire.

— Calcul de λ_{in} par l'algorithme de Dantzig, Blattner et Rao (cf. [4] ou [16], tome II, VII B.3) ou entrée directe de ces nombres.

— Possibilité de prendre en compte un grand nombre de contraintes de dimension ainsi que des sommets excluant beaucoup d'autres sommets. Grâce à la codification de ces contraintes, seuls le nombre total d'arcs dimensionnés et le nombre total de sommets exclus sont limités.

— Existence d'une option ELEMENTAIRE assurant automatiquement les exclusions nécessaires.

— Possibilité de réaliser une ou plusieurs énumérations en indiquant à chaque fois les contraintes d'exclusion, les contraintes de dimension que l'on veut prendre en compte avec leurs seconds membres, l'origine, l'extrémité et la marge en pourcentage (ε) ou en valeur absolue.

— Possibilité de minimisation sur une dimension ou sur la longueur elle-même.

— Possibilité d'arrêt à n solutions et d'élimination des chemins ayant plus de k sommets.

Pour une description plus détaillée nous renvoyons au manuel de références (cf. [2 *bis*]). Deux aspects méritent encore d'être précisés avant de décrire un exemple d'application.

Afin de pouvoir traiter des graphes de grande taille, il était important de réduire l'occupation de mémoire et de faciliter la description de l'arborescence conformément à un ordre de Tarry. La codification retenue répond à ce double objectif. Les EXTrémités et les Longueurs des ARCs sont rangées, groupées par même origine, dans deux tableaux EXT et LARC. On prend comme ordre transverse sur les suivants d'un même sommet celui des indices dans le tableau EXT. Pour localiser les suivants d'un même sommet dans EXT, on construit deux tableaux NSUI et ISUI qui indiquent respectivement pour chaque sommet le Nombre de SUIvants et l'indice diminué de 1 du premier de ces SUIvants dans EXT. Ainsi, dans EXT, les extrémités des arcs ayant le sommet i comme origine seront logées à partir de la position ISUI(i) + 1 jusqu'à la position ISUI(i) + NSUI(i), comme le montre la figure 3. De même, les lon-

gueurs de ces arcs seront logées dans les positions correspondantes du tableau LARC. Pour les contraintes de dimension et d'exclusion, on a adopté des codifications analogues.

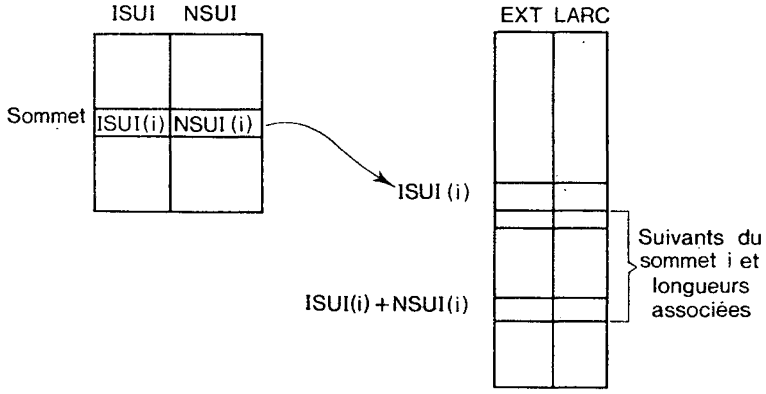


Figure 3
Codification du graphe

La pile introduite au § 2.2 est bien adaptée à la description de l'algorithme mais serait lourde à représenter telle quelle. Soit un ensemble $\bar{C}_\varepsilon(\omega^0)$; nous lui associons deux tableaux :

— un tableau CHEM contenant la liste des étiquettes de $\gamma(\omega^0)$.

— un tableau IBR tel que IBR(i) désigne le numéro d'ordre du sommet CHEM($i + 1$) dans l'ordre transverse des suivants de CHEM(i).

La description, suivant l'ordre de Tarry des ensembles $\bar{C}_\varepsilon(\omega)$ se ramène alors à une énumération lexicographique de chemins de G (cf. arborescence de chemins du 2.1).

Avec ces conventions l'occupation de mémoire, évaluée en nombre de variables entières, est approximativement donnée par la formule suivante :

$10x$ (nombre de sommets) $+ 7x$ (nombre d'arcs) + nombre total de sommets exclus + (nombre de dimensions) $\times$ (nombre total d'arcs ayant au moins une dimension non nulle).

Ainsi le programme peut traiter des graphes ayant 1 000 sommets, 5 000 arcs et 3 contraintes de dimension.

3.3. **Un exemple d'application** (pour plus de détails, cf. [5 *bis*])

Nous avons appliqué ce programme au « graphe de base » du réseau métropolitain de la R.A.T.P. (1). Les sommets sont les couples (n° de ligne, station

(1) Régie Autonome des Transports Parisiens.

de correspondance). Le graphe est symétrique et comporte 141 sommets. Les arcs au nombre de 428, sont de deux types :

— les arcs de parcours, correspondant à un trajet en train, dont la longueur est égale au temps de parcours,

— les arcs de passage, correspondant à un trajet effectué à pied entre deux lignes à l'intérieur d'une même station de correspondance et dont la longueur est égale au temps de parcours affecté d'un coefficient de pénibilité. On a affecté à ces derniers arcs une dimension égale à 1.

D'autre part, on est conduit à ne s'intéresser qu'aux chemins élémentaires. Mais, l'éclatement de chaque station de correspondance en autant de sommets qu'il y passe de lignes nécessite l'exclusion des chemins empruntant deux arcs de passage d'une même station. Ceci s'obtient aisément au moyen d'une contrainte d'exclusion de niveau 2 : chaque sommet excluant tous ceux de la même station de correspondance. Ainsi, le passage par deux sommets de la station interdit le passage par un troisième.

Le programme permet alors d'énumérer entre deux sommets déterminés quelconques :

— les chemins ε-minimum,

— les chemins ε-minimum sans plus de n correspondances,

— les chemins ε-minimum minimisant le nombre de correspondances,

— les chemins minimisant la longueur sans plus de n correspondances.

On considère, à titre d'exemple, les chemins ε-minimum entre les sommets 95 (Pasteur ligne 6) et 98 (Père Lachaise ligne 3) du graphe de base du réseau métropolitain. Chaque sommet exclut tous ceux de la même station de correspondance de sorte que, avec un niveau d'exclusion égal à deux, le passage par deux sommets de la même station sera interdit. A l'aide de la contrainte de dimension, on exclut les chemins qui empruntent plus de trois arcs de passage du graphe. On a fixé à 10 % la marge d'écart toléré. On trouvera en figure 4 les sorties sur listing des résultats obtenus sur ordinateur CDC 6600. Le programme a fourni six chemins ε-minimum admissibles (1). Le temps d'exécution du programme est d'environ 2 secondes. Le temps se répartit comme suit :

— temps de lecture des données, de constitution des tableaux et d'impression du graphe	1 798 ms
— temps de calcul des longueurs des plus courts chemins	127 ms
— temps d'exploration du graphe et d'énumération des chemins solutions	76 ms

(1) Les longueurs sont exprimées en 1/10 de minutes.

```
CHEMINS E-MINIMUM  ENTRE LES SOMMETS PASTEUR6
                              ET  PERE LACHAISE3
**********************************************************

LA PLUS COURTE DISTANCE DE  95 A   98 EST        397

LA LONGUEUR CRITIQUE EST      436

OPTION ELEMENTAIRE
**********************************************************

CONTRAINTE   1   SECOND MEMBRE       3
**********************************************************

CHEMIN ACCEPTE LONGUEUR      432
    CONTRAINTE    1   VALEUR        3
PASTEUR6         PASTEUR12        MONTPARNASSE12  MONTPARNASSE4   ODEON4          CHATELET4
REAUMUR4         REAUMUR3         ARTS-METIERS3   REPUBLIQUE3     PERE LACHAISE3

CHEMIN ACCEPTE LONGUEUR      431
    CONTRAINTE    1   VALEUR        2
PASTEUR6         LAMOTTE PICQUET6LAMOTTE PICQUET8INVALIDES8      CONCORDE8       MADELEINE8
OPERA8           RICH DROUOT8     MONTMARTRE8     BONNE NOUVELLE8 STRAS ST DENIS8 REPUBLIQUE8
REPUBLIQUE3      PERE LACHAISE3

CHEMIN ACCEPTE LONGUEUR      427
    CONTRAINTE    1   VALEUR        3
PASTEUR6         MONTPARNASSE6    RASPAIL6        RASPAIL4        MONTPARNASSE4   ODEON4
CHATELET4        CHATELET11       HOTEL DE VILLE11ARTS-METIERS11  ARTS-METIERS3   REPUBLIQUE3
PERE LACHAISE3

CHEMIN ACCEPTE LONGUEUR      397
    CONTRAINTE    1   VALEUR        2
PASTEUR6         MONTPARNASSE6    RASPAIL6        RASPAIL4        MONTPARNASSE4   ODEON4
CHATELET4        REAUMUR4         REAUMUR3        ARTS-METIERS3   REPUBLIQUE3     PERE LACHAISE3

CHEMIN ACCEPTE LONGUEUR      428
    CONTRAINTE    1   VALEUR        2
PASTEUR6         MONTPARNASSE6    RASPAIL6        DENFERT6        PCE D'ITALIE6   PCE D'ITALIE5
AUSTERLITZ5      BASTILLE5        OBERKAMPF5      REPUBLIQUE5     REPUBLIQUE3     PERE LACHAISE3

CHEMIN ACCEPTE LONGUEUR      439
    CONTRAINTE    1   VALEUR        2
PASTEUR6         MONTPARNASSE6    RASPAIL6        DENFERT6        PCE D'ITALIE6   DAUMESNIL6
NATION6          NATION2          PERE LACHAISE2  PERE LACHAISE3

FIN DE L EXPLORATION

TEMPS DE RECHERCHE DES CHEMINS E-MINIMAUX=        76
```

Figure 4

Exemple de listing de sortie du programme Multichemin

L'importance des temps de lecture et d'impression est justifiée par la nature des données. De plus, en cas de répétition du programme, ce temps disparaît.

Si on avait imposé aux chemins de ne pas emprunter plus de deux arcs de passage du graphe, deux des chemins auraient été éliminés et si on avait imposé aux chemins de ne pas emprunter plus d'un arc de passage, il n'y aurait pas eu de solution.

Au lieu de limiter le nombre d'arcs de passage empruntés, on aurait pu chercher à obtenir un chemin ε-minimum qui minimise ce nombre d'arcs. On aurait obtenu un chemin optimum empruntant deux arcs de passage.

Enfin, la minimisation de la longueur donne comme résultat un chemin de longueur 397, compte tenu des contraintes imposées.

BIBLIOGRAPHIE

[1] BALAS E., *An additive algorithm for solving linear programs with zero-one variables*, Operations Research, 1965, 13, 517-549.

[2] BELLMAN R. and KALABA R., *On the Kth best policies*, Journal of the Society for Industrial and Applied Mathematics, 1960, 8, 582-588.

[2 *bis*] BRAGARD L. et GALLAND D., *Le programme Multichemin : manuel de références*, Document Technique n° 27, mars 1972, Direction Scientifique, Groupe METRA.

[3] CLARKE S., KRIKORIAN A. and RAUSEN J., *Computing the N best loopless paths in a network*, Journal of the Society for Industrial and Applied Mathematics, 1963, 11, 1096-1102.

[4] DANTZIG G. B., BLATTNER W. O. and RAO M. R., *All shortest routes from a fixed origin in an graph*, Théorie des graphes, Journées Internationales d'Études Rome, 1966 (Dunod, Paris, 1967).

[5] DREYFUS S. E., *An appraisal of some shortest path algorithms*, Operations Research, vol. 17, n° 3, 1969.

[5 *bis*] FAYEIN V., *Chemins-ɛminimaux dans un graphe valué*, Colloque de Recherche Opérationnelle du Comité International des Métros, Milan, mai 1972.

[6] HANSEN P., *Les procédures d'optimisation par séparation : présentation générale*, Revue de Statistique, Tijdschrift voor Statistiek, 11 (3), 1971.

[7] HOFFMAN W. and PAVLEY R., *A method for the solution of the Nth best path problem.* Journal of the Association for Computing Machinery, 1959, 6, 506-514.

[8] KAUFMAN A. et CRUON R., *Etude de la sensibilité en programme dynamique ; Politiques k-optimales en avenir certain*, Revue Française de Recherche Opérationnelle n° 32, 1964.

[8 *bis*] LAWLER E. L., *A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem*, Management Science, vol. 18, n° 7, March 1972.

[9] PAIR C. et DERNIAME J. C., *Problèmes de cheminement dans les graphes*, Monographies d'informatique AFCET (Dunod, 1971).

[10] PICHAT E., *Contribution à l'algorithmique non numérique dans les ensembles ordonnés*, Thèse Université de Grenoble (1970).

[11] PLOWDEN S., Note interne Metra Consulting Group Ltd, 1969.

[12] POLLACK M., *Solutions of the Kth best route through a network*, Journal of Mathematical Analysis and Applications, 1961, 3, 547-559.

[13] POLLACK M., *The Kth best route through a network*, Operations Research, 1961, 9, 578-580.

[14] REGIE AUTONOME DES TRANSPORTS PARISIENS, *Programme de recherche des chemins les meilleurs entre deux stations du réseau métropolitain*, Service de l'Informatique, 1971.

[15] ROY B., *Procédures d'Exploration par Séparation et Evaluation (PSEP, PSES)*, RIRO, n° V-1, 1969.

[16] ROY B., *Algèbre Moderne et Théorie des Graphes orientées vers les sciences économiques et sociales*, Tomes 1 et 2 (Dunod, Paris, 1969-1970).

[17] ROY B., *An algorithm for a general constrained set covering problem*, Computing and Graph Theory, Ronald C. Read, Academic Press Inc., New York, 1972.

[18] SAKAROVITCH M., *The K shortest chains in a graph*, Transportation Research, 1968, 2, 1-11.

[19] SAKAROVITCH M., *Les k plus courts chemins élémentaires dans un graphe*, Note interne Institut de Recherche des Transports.

[20] YEN J. Y., *Finding the K shortest loopless paths in a network*, Management Science, vol. 17, n° 11, 1971, 712-716.