

JACQUES DESROSIERS

PAUL PELLETIER

FRANÇOIS SOUMIS

## **Plus court chemin avec contraintes d'horaires**

*RAIRO. Recherche opérationnelle*, tome 17, n° 4 (1983),  
p. 357-377

[http://www.numdam.org/item?id=RO\\_1983\\_\\_17\\_4\\_357\\_0](http://www.numdam.org/item?id=RO_1983__17_4_357_0)

© AFCET, 1983, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques  
<http://www.numdam.org/>

## PLUS COURT CHEMIN AVEC CONTRAINTES D'HORAIRES (\*)

par Jacques DESROSIER<sup>(1)</sup>, Paul PELLETIER et François SOUMIS

**Résumé.** — Nous déterminerons un chemin de longueur minimal dans un réseau où, à chaque nœud, est associé un intervalle de temps pendant lequel il est permis de le visiter. Ce problème survient naturellement lors de la fabrication de routes de véhicules qui respectent des contraintes d'horaires.

Ce problème est résolu par programmation dynamique, avec comme variables d'état, des couples (temps, longueur) associés aux nœuds. Pour le problème classique d'un plus court chemin, l'algorithme de Ford-Bellman-Moore associe à chaque nœud une seule étiquette (longueur) qui est progressivement améliorée. Pour le problème d'un plus court chemin avec contraintes d'horaires, il faut conserver à chaque nœud un ensemble de couples (temps, longueur). Nous avons développé une gestion efficace qui conserve à chaque nœud une liste ne contenant que des couples pertinents en réalisant de façon dynamique la création et l'élimination de couples.

**Mots clés :** chemin de longueur minimale; réseau, programmation dynamique; contraintes d'horaires.

**Abstract.** — We find the shortest path in a network where each node can be visited only during a certain specified time interval. This problem arises naturally in vehicle routing when scheduling constraints must be respected.

The problem is solved using dynamic programming, taking as state variables, the (time, length) pairs associated with the nodes. In the classical shortest path problem, the Ford-Bellman-Moore algorithm attaches a single label (length) to each node which is gradually improved. In the shortest path problem with scheduling constraints, a set of (time, length) pairs must be stored for each node. We have developed an efficient procedure which dynamically creates and eliminates pairs so that only pertinent pairs have to be stored for each node.

**Key words:** shortest path, network, dynamic programming, scheduling constraints.

### 1. INTRODUCTION

Le problème de la recherche d'un chemin de longueur minimale entre deux nœuds  $s$  et  $t$  d'un réseau est bien défini dans la mesure où il n'y a pas de circuit de longueur négative, et il existe alors de nombreux algorithmes efficaces pour le résoudre. Soit un réseau  $G=(N; A)$  où  $N=\{1, 2, \dots, n\}$  est l'ensemble des nœuds et  $A$  est celui des arcs orientés  $(i, j)$ . On associe à chaque arc  $(i, j) \in A$  une longueur  $c_{ij}$ . Un chemin de longueur minimale entre les nœuds  $s$  et  $t$  peut être

(\*) Reçu en avril 1982.

(1) H.E.C., 5255 avenue Decelles, Montréal, Québec, H3T 1V6 Canada.

obtenu à partir d'une solution optimale  $X^*$  du problème (1)-(3) ci-dessous; en effet,  $X_{st} = \{(i, j) \mid x_{ij}^* = 1\}$  est un tel chemin :

Minimiser :

$$(1) \quad \sum_{(i,j) \in A} c_{ij} x_{ij},$$

sujet à :

$$(2) \quad \sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1, & i = s, \\ 0, & i \neq s, \neq t \\ -1, & i = t, \end{cases}$$

$$(3) \quad x_{ij} \geq 0, \quad (i, j) \in A.$$

Nous étudions une extension de ce problème en introduisant sur chaque arc  $(i, j) \in A$  une durée  $t_{ij}$  non négative et sur chaque nœud  $i \in N$  un intervalle fermé de temps  $[a_i, b_i]$  pendant lequel il est permis de le visiter. Si des coûts  $c_i$  et des durées  $d_i$  sont associés aux nœuds, ces valeurs sont transférées sur les arcs en redéfinissant adéquatement les longueurs et les durées des arcs.

## 2. FORMULATION DU PROBLÈME

Soit le réseau  $G = (N; A)$  où l'ensemble  $A$  est constitué d'arcs  $(i, j)$  satisfaisant :

$$(4) \quad a_i + t_{ij} \leq b_j.$$

Soit  $t_i \in [a_i, b_i]$  le temps où le nœud  $i$  est visité par le chemin le plus court de  $s$  à  $t$ ; il faut préciser qu'on a le droit d'attendre en un nœud  $i$  si on arrive avant  $a_i$ . Une formulation du problème de chemin le plus court avec horaire sur les nœuds, où  $x_{ij}$  est la valeur du flot sur l'arc  $(i, j) \in A$  est la suivante :

Minimiser :

$$(5) \quad \sum_{(i,j) \in A} c_{ij} x_{ij}$$

sujet à :

$$(6) \quad \sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1, & i = s, \\ 0, & i \neq s, \neq t, \\ -1, & i = t, \end{cases}$$

$$(7) \quad x_{ij} \geq 0, \quad (i, j) \in A,$$

$$(8) \quad x_{ij} \neq 0 \Rightarrow t_i + t_{ij} \leq t_j, \quad (i, j) \in A,$$

$$(9) \quad a_i \leq t_i \leq b_i, \quad i \in N.$$

### 3. REVUE DE LA BIBLIOGRAPHIE

Le problème du plus court chemin avec la contrainte linéaire

$$(10) \quad \sum_{(i,j) \in A} t_{ij} x_{ij} \leq \lambda$$

a déjà retenu l'attention de certains auteurs. Jokschi [6] étudie le cas général où les valeurs  $t_{ij}$  et  $\lambda$  sont de signes quelconques. Il s'agit d'une étude théorique proposant de traiter ce problème avec une généralisation de l'algorithme de Bellman (pour le problème du plus court chemin classique). Le développement de cette méthode n'a pas été réalisé.

Par la suite, Minoux [8] puis Handler et Zang [5] reprennent le problème du plus court chemin avec la contrainte additionnelle du type knapsack :

$$(11) \quad \sum_{(i,j) \in A} t_{ij} x_{ij} \leq \lambda \quad \text{avec} \quad c_{ij} \geq 0 \quad \text{et} \quad t_{ij} \geq 0.$$

Compte tenu du fait que les coûts sont non négatifs, l'algorithme de Minoux peut être considéré comme une généralisation de l'algorithme de Dijkstra [3]. Il décrit aussi une résolution approchée par une méthode lagrangienne. D'autre part, Handler et Zang [5] développent un algorithme exact fondé sur la relaxation lagrangienne.

### 4. EXEMPLES D'APPLICATIONS

Considérons l'application suivante qui est à l'origine de la recherche sur le problème du chemin de longueur minimale avec contraintes d'horaires sur les nœuds [2].

Soit un ensemble  $N$  de parcours; à chaque parcours  $i$  sont associés un lieu de départ et un lieu d'arrivée, une durée  $d_i$  et un temps variable de début  $t_i$  dans un intervalle  $[a_i, b_i]$ . Entre la fin du parcours  $i$  et le début du parcours  $j$  existe un transit d'une durée  $t_{ij}$  et d'un coût  $c_{ij}$  (incluant respectivement la durée et le coût du parcours  $i$ ). Le problème consiste à former un ensemble de routes (suite de parcours et de transits) qui satisfont les contraintes d'horaires.

Minimiser : Coût des routes.

Sujet à : Effectuer tous les parcours.

Horaire réalisable.

Sans les contraintes d'horaires, ce problème se formule comme un problème de transport ou d'affectation, ou encore comme un problème de recouvrement. Dans ce dernier cas, on procède par génération de colonnes où le sous-problème est un problème de chemin le plus court avec des coûts de signes quelconques sur les arcs. En incluant les contraintes d'horaires, le sous-problème devient un problème de chemin le plus court avec contraintes d'horaires sur des nœuds.

Voici une autre application : un artiste préparant sa tournée de spectacles. Soit un ensemble  $N$  de sites; à chaque site est associé un calendrier des dates adjacentes de représentations possibles. Entre les sites  $i$  et  $j$  existe un transit d'une durée  $t_{ij}$  (temps requis pour la représentation au site  $i$  plus temps de déplacement entre  $i$  et  $j$ ) et d'un coût  $c_{ij}$  (profit au site  $i$  moins frais de déplacement entre  $i$  et  $j$ ). Le problème consiste à choisir l'ensemble des représentations à donner pour maximiser le profit tout en respectant le calendrier. Il s'agit d'un problème de chemin le plus long avec contraintes de temps sur les nœuds.

Dans l'exemple précédent, les visites multiples à un site sont permises, tandis que dans le premier exemple, les circuits ne sont pas permis, chaque parcours ne devant être effectué qu'une seule fois.

Le problème du chemin de longueur minimale avec contrainte d'horaires sur les nœuds quand le temps d'attente n'est pas permis inclut le cas du problème du chemin de longueur minimale entre  $s$  et  $t$  avec la contrainte linéaire :

$$(12) \quad \lambda_1 \leq \sum_{(i,j) \in A} t_{ij} x_{ij} \leq \lambda_2, \quad t_{ij} \geq 0.$$

En effet, on associe au nœud  $t$  l'intervalle  $[\lambda_1, \lambda_2]$  et à tous les autres nœuds l'intervalle  $[0, \lambda_2]$ . Les applications suivantes étudiées dans [8] correspondent au problème avec une seule contrainte d'inégalité :

$$\sum_{(i,j) \in A} t_{ij} x_{ij} \leq \lambda_2,$$

chemin de longueur minimale avec contrainte de temps ou de capacité; chemin élémentaire de longueur minimale avec contrainte sur le nombre d'arcs; recherche d'un chemin critique sur un réseau PERT avec durée des tâches aléatoire avec borne supérieure sur la variance du chemin; chemin de longueur minimale dans un réseau téléphonique sous réserve que l'affaiblissement total le long de ce chemin soit inférieur à une limite donnée; chemin de longueur minimale où la longueur des arcs croît ou décroît selon des fonctions en escalier de l'investissement, tout en respectant une contrainte budgétaire. Nous renvoyons le lecteur intéressé par plus de détails sur ces différents problèmes à la référence [8] qui les décrit plus complètement.

Lorsque les intervalles de temps sur les nœuds ne sont pas contraignants, on retrouve le problème classique d'un plus court chemin, si le réseau ne présente pas de circuit négatif. Si les intervalles de temps se réduisent à un point, le graphe ne contient que les arcs satisfaisant  $a_i + t_{ij} = b_i + t_{ij} \leq a_j = b_j$  et l'on retrouve le problème classique, s'il existe un chemin de  $s$  à  $t$  dans ce graphe.

On obtient un problème du type voyageur de commerce si les intervalles de temps ne sont pas contraignants et si à chaque nœud est associé un coût très

négalif. En effet, avec la contrainte  $x_{ij}$  entier, la relation conditionnelle (8) peut s'écrire sous forme linéaire :

$$(13) \quad t_i + t_{ij} - t_j \leq (1 - x_{ij}) m_{ij}, \quad (i, j) \in A,$$

où  $m_{ij} \geq b_i + t_{ij} - a_j$ ; pour  $a_i = 0$ ,  $b_i = n - 1$  et  $t_{ij} = 1$ , on peut poser  $m_{ij} = n$  et les inéquations (13) correspondent aux contraintes d'élimination de sous-tours proposées par Miller, Tucker et Zemlin [7]

$$(14) \quad t_i - t_j + n x_{ij} \leq n - 1, \quad (i, j) \in A.$$

### 5. ÉTUDE DES SOLUTIONS OPTIMALES

Le problème formé des équations (5)-(7) est celui du chemin le plus court classique entre les nœuds  $s$  et  $t$ ; ce problème a toujours une solution optimale entière (dans la mesure où il n'existe pas de circuit négatif), la matrice des contraintes étant totalement unimodulaire.

La relation conditionnelle (8) qui impose les contraintes d'horaires empêche les visites multiples sur les nœuds, car chaque variable  $t_i$  ne peut prendre qu'une seule valeur. Si le chemin de longueur minimale de  $s$  à  $t$  passe par le nœud  $i$ , alors  $t_i$  est le temps de visite au nœud  $i$ . Si le nœud  $i$  ne fait pas partie du chemin de longueur minimale, alors  $t_i$  n'est pas contraint par (8) et peut prendre une valeur quelconque dans l'intervalle  $[a_i, b_i]$ .

S'il existe un chemin réalisable dans le temps de  $s$  à  $t$ , il est intéressant de constater que le problème du plus court chemin avec contraintes d'horaires sur les nœuds a toujours une solution optimale entière. En effet, la relation (8) signifie que si  $x_{ij} > 0$  alors  $t_i + t_{ij} \leq t_j$ , c'est-à-dire que si un flot positif passe sur l'arc  $(i, j)$ , le chemin est réalisable dans le temps de  $i$  à  $j$ . Si les valeurs optimales  $x_{ij}$  sont fractionnaires, la solution  $Z^*$  est composée de plusieurs chemins de longueur  $l_p$  ayant chacun un flot positif  $\lambda_p$  où  $\sum_p \lambda_p = 1$  :

$$Z^* = \sum_p l_p \lambda_p.$$

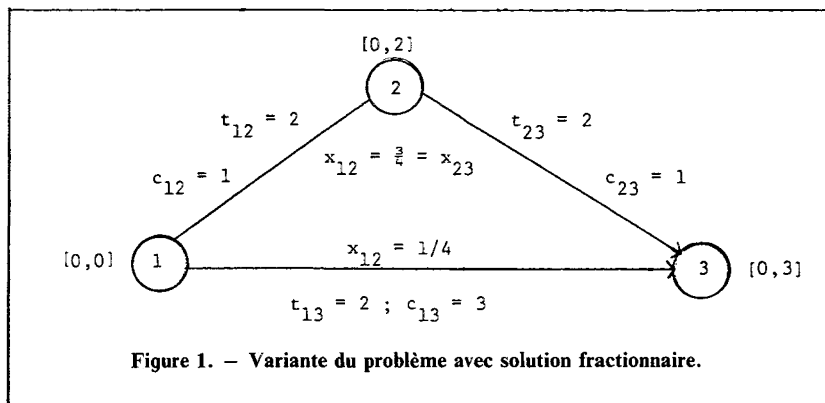
Assignons un flot unitaire sur les arcs du chemin de longueur minimale  $l_{\min}$ ; ce chemin est réalisable dans le temps et il constitue une solution optimale entière puisque :

$$(15) \quad Z^* \leq l_{\min} \cdot 1 = \sum_p l_{\min} \lambda_p \leq \sum_p l_p \lambda_p = Z^*.$$

De plus, chacun des autres chemins de la solution optimale affecté d'un flot unitaire est optimal.

La forme linéaire (13) est équivalente à la relation conditionnelle (8) seulement si on impose la condition  $x_{ij}$  binaire; pour  $x_{ij} \in (0, 1)$ , on peut avoir

$0 < t_i + t_{ij} - t_j \leq (1 - x_{ij})m_{ij}$  qui ne satisfait pas (8). De plus, l'intégrité de la solution optimale n'est plus garantie (fig. 1).



Le chemin optimal du nœud 1 au nœud 3 est  $1 \rightarrow 3$  de longueur 3 et c'est le seul chemin réalisable dans le temps. Avec  $t_3 = 3$  et  $m_{23} = b_2 + t_{23} - a_3 = 4$ , des flots fractionnaires de  $3/4$  sur le chemin  $1 \rightarrow 2 \rightarrow 3$  et de  $1/4$  sur le chemin  $1 \rightarrow 3$  satisfont les contraintes linéaires, pour une solution optimale de longueur  $2 + 1/4$ . En général, avec  $t_3 = 3$  et  $m_{23} \geq 4$ , un flot fractionnaire de  $(1 - 1/m_{23})$  sur le chemin  $1 \rightarrow 2 \rightarrow 3$  et de  $1/m_{23}$  sur le chemin  $1 \rightarrow 3$  est aussi une solution optimale de longueur  $(2 + 1/m_{23})$ .

La relation conditionnelle (8) s'écrit aussi sous une forme non linéaire équivalente :

$$(16) \quad x_{ij}(t_i + t_{ij} - t_j) \leq 0, \quad (i, j) \in A.$$

Cette forme non linéaire (16) tout comme la relation conditionnelle assure l'existence d'une solution optimale binaire.

Il est intéressant de noter l'analogie entre l'équation (16) et l'équation de complémentarité du problème classique d'un chemin de longueur minimale :

$$(17) \quad x_{ij}(\pi_i + c_{ij} - \pi_j) = 0, \quad (i, j) \in A,$$

où  $\pi_i$  est la variable duale associée au nœud  $i$ . L'équation (16) peut aussi s'écrire avec l'égalité si le temps d'attente n'est pas permis. Dans ce cas, même si les coûts  $c_{ij}$  étaient égaux aux  $t_{ij}$ , les variables  $\pi_i$  et  $t_i$  ne seraient pas équivalentes puisque les variables  $t_i$  sont contraintes par les intervalles de temps, alors que les variables  $\pi_i$  sont libres.

## 6. UN ALGORITHME POUR LE PROBLÈME DU PLUS COURT CHEMIN AVEC CONTRAINTES D'HORAIRES

### 6.1. Arbre des plus courts chemins

Pour résoudre par programmation dynamique le problème du calcul d'un plus court chemin de  $s$  à  $t$ , on le plonge dans le problème du calcul de l'arbre des plus courts chemins de  $s$  vers tous les autres nœuds  $N' = N - \{s\}$  :

Minimiser :

$$\sum_{k \in N'} \sum_{(i,j) \in A} c_{ij} x_{ij}^k,$$

sujet à :

$$(19) \quad \sum_j x_{ij}^k - \sum_j x_{ji}^k = \begin{cases} 1 & k \in N'; \quad i = s, \\ 0 & k \in N'; \quad i \neq k, \\ -1 & k \in N'; \quad i = k, \end{cases}$$

$$(20) \quad x_{ij}^k \geq 0, \quad (i,j) \in A; \quad k \in N',$$

$$(21) \quad a_i \leq t_i^k \leq b_i, \quad i \in N; \quad k \in N',$$

$$(22) \quad x_{ij}^k > 0 \Rightarrow t_i^k + t_{ij} \leq t_j^k \quad (i,j) \in A; \quad k \in N'.$$

Un chemin de longueur minimale de  $s$  à  $k$  est donné par l'ensemble  $\{(i,j) \mid x_{ij}^k = 1\}$ . L'arbre contiendra jusqu'à  $n-1$  chemins pouvant passer à des temps distincts sur un nœud particulier (fig. 2).

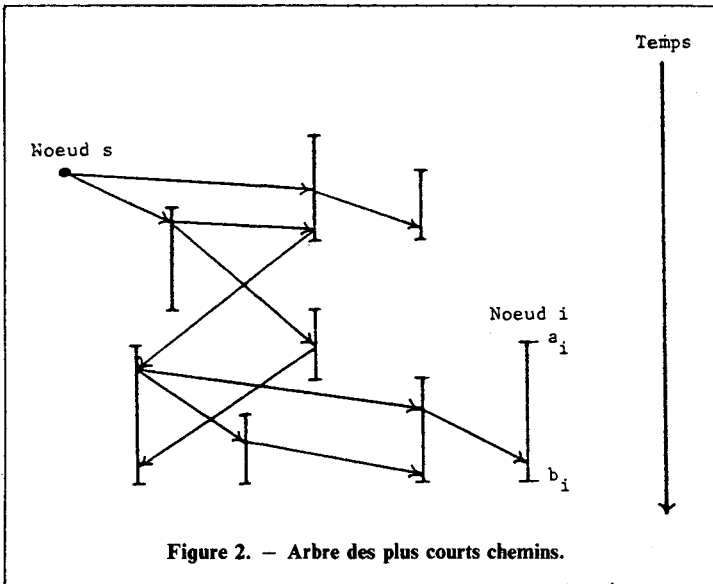


Figure 2. — Arbre des plus courts chemins.



## 6.2. Algorithme de Ford-Bellman-Moore [4, 1, 9]

Rappelons d'abord l'algorithme de Ford-Bellman-Moore avant de discuter notre généralisation. Notons par  $F(i) = \{j \mid (i,j) \in A, j \in N\}$  l'ensemble des successeurs de  $i$ . L'arbre des chemins les plus courts de la source  $s$  vers tous les autres nœuds du réseau s'obtient en associant à chaque nœud une étiquette qui est progressivement améliorée. Soit  $L$  l'ensemble des nœuds n'ayant pas été explorés après une amélioration : il contient au départ  $\{s\}$  avec l'étiquette  $\pi_s = 0$ . La méthode d'étiquetage consiste à choisir un nœud  $i \in L$ , l'enlever de l'ensemble  $L$ , et tenter d'améliorer tous ses successeurs  $F(i)$ . Les nouveaux nœuds améliorés sont insérés dans  $L$ . Lorsque l'ensemble  $L$  est épuisé, l'étiquette  $\pi_i$  du nœud  $i$  représente la longueur du plus court chemin à partir de la source  $s$ .

### 0: Initialisation

$$\pi_s = 0$$

$$\pi_i = \infty \quad i \neq s$$

$$L = \{s\}$$

### 1: Exploration de $F(i)$

Choisir un nœud  $i \in L$

$$\forall j \in F(i) \text{ tel que } \pi_j > \pi_i + c_{ij}$$

$$\text{poser } \pi_j = \pi_i + c_{ij}$$

$$L = L \cup \{j\}$$

### 2: Réduction de $L$

$$L = L - \{i\}$$

Si  $L = \emptyset$  alors arrêter  
sinon retourner à 1.

Algorithme pour le problème classique d'un plus court chemin.

Cette approche du type programmation dynamique a été adaptée pour traiter le problème du plus court chemin avec contraintes d'horaires sur les nœuds. Nous verrons successivement la nature des étiquettes, le principe d'optimalité et la gestion des étiquettes.

### 6.3. Nature des étiquettes

Pour traiter simultanément les variables de temps  $t_i$  et les variables duales  $\pi_i$  associées à chaque nœud  $i$ , nous utilisons comme étiquette des couples  $(t_i, \pi_i)$ . Pour préciser le sens d'une telle étiquette, introduisons la notion de chemin réalisable, c'est-à-dire un chemin pour lequel il est possible d'associer des temps aux nœuds de façon à ce que les contraintes d'horaires soient satisfaites.

Notons qu'il existe généralement pour un chemin réalisable un nombre infini de choix de temps satisfaisant les contraintes d'horaires, mais nous considérons qu'il s'agit d'un seul et même chemin. Ainsi l'étiquette  $(t_i, \pi_i)$  est attribué au nœud  $i$  si et seulement si il existe un chemin réalisable de coût  $\pi_i$  partant de  $s$  et arrivant en  $i$  au plus tôt au temps  $t_i$ . Un tel chemin  $X_{si}$  associé à l'étiquette  $(t_i, \pi_i)$  a le même coût  $\pi_i$  pour tous les temps d'arrivée en  $i$  dans l'intervalle  $[t_i, b_i]$  car il n'y a pas de coût associé à l'attente.

L'algorithme de Ford-Bellman-More, pour le problème du plus court chemin classique, associe à chaque nœud une seule étiquette qui est progressivement améliorée. Avec des couples  $(t_i, \pi_i)$ , il n'est pas possible de conserver seulement la meilleure étiquette pour chaque nœud car il n'y a pas d'ordre total dans  $\mathbf{R}^2$  : il faut conserver un ensemble d'étiquettes  $(t_i^\alpha, \pi_i^\alpha)$ ,  $\alpha \geq 1$ . Cet ensemble est évidemment fini puisque le nombre fini de nœuds assure un nombre fini de chemins réalisables de  $s$  à  $i$  et chacun de ces chemins réalisables ne peut produire qu'une seule étiquette.

### 6.4. Principe d'optimalité

Le succès des méthodes de marquage pour le problème classique du plus court chemin réside dans l'application du principe d'optimalité qui s'énonce comme suit : si  $X_{sj}$  est un chemin optimal entre  $s$  et  $j$  d'une longueur  $\pi_j$ , et si l'arc  $(i, j)$  est le dernier arc de ce chemin, alors le sous-chemin de  $s$  à  $i$  d'une longueur  $\pi_i$  est optimal entre  $s$  et  $i$ .

Dans le cas du plus court chemin avec contraintes d'horaires sur les nœuds, le principe d'optimalité doit être modifié. En effet, soit  $(t_j^\alpha, \pi_j^\alpha)$  une étiquette au nœud  $j$  et  $X_{sj}^\alpha$  le chemin associé. Si ce chemin est optimal parmi les chemins arrivant au nœud  $j$  à des temps  $t_j$  respectant  $t_j \leq t_j^\alpha$  et si  $(i, j)$  est le dernier arc du chemin  $X_{sj}^\alpha$ , alors le sous-chemin de  $s$  à  $i$  est un chemin optimal parmi les chemins arrivant au nœud  $i$  à des temps  $t_i$  respectant  $t_i \leq t_j^\alpha - t_{ij}$ .

### 6.5. Relation d'ordre sur les étiquettes

Nous allons construire un ensemble d'étiquettes à chaque nœud  $i$  qui donne les meilleures distances à différents temps de l'intervalle  $[a_i, b_i]$ . Définissons la

relation d'ordre partiel  $<$  au nœud  $i$  :

$$(23) \quad (t_i^1, \pi_i^1) < (t_i^2, \pi_i^2) \Leftrightarrow (t_i^1 \leq t_i^2) \quad \text{et} \quad (\pi_i^1 \leq \pi_i^2).$$

Une étiquette est dite minimale dans un ensemble s'il n'y existe pas d'étiquette inférieure suivant cette relation d'ordre. Les étiquettes le long du chemin optimal de  $s$  à  $i$  sont des étiquettes minimales dans tout ensemble, la preuve se faisant par induction. En effet, l'étiquette optimale  $(t_i^*, \pi_i^*)$  est minimale et en appliquant le principe d'optimalité, les étiquettes intermédiaires d'un plus court chemin satisfaisant des contraintes d'horaires sont minimales. Ce résultat permet de réduire l'ensemble des étiquettes de chaque nœud en enlevant celles qui ne sont pas minimales et de conserver un sous-ensemble irréductible dans le cas où le temps d'attente est permis. (Notons que, dans le cas où le temps d'attente n'est pas permis, il faudrait définir une autre relation d'ordre plus complexe, et le nombre d'étiquettes à conserver serait généralement beaucoup plus élevé.) L'ensemble irréductible associé au nœud  $i$  est ordonné suivant l'ordre strictement croissant des temps; remarquons que les longueurs sont alors en ordre strictement décroissant comme l'illustre la figure 3.

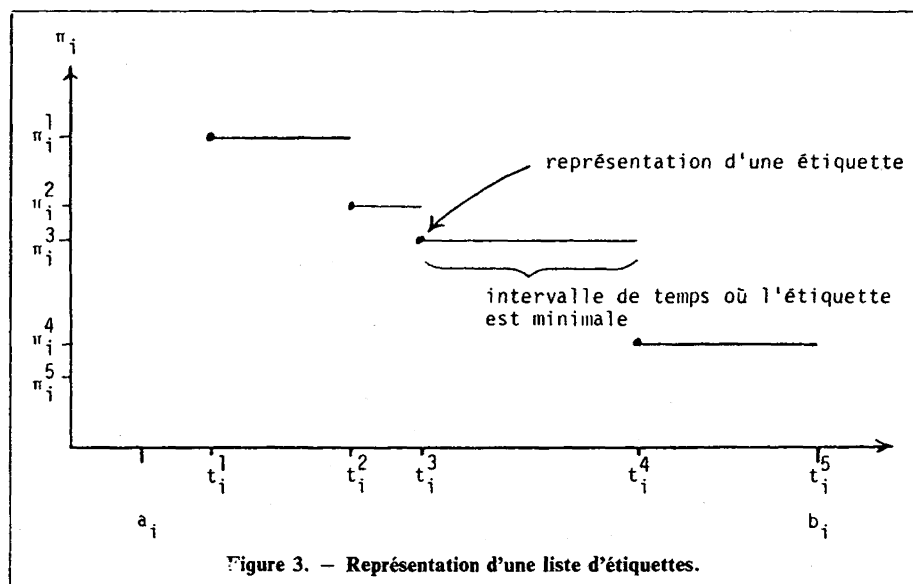


Figure 3. — Représentation d'une liste d'étiquettes.

L'ensemble irréductible associé au nœud  $i$  contient  $\alpha(i)$  paires d'étiquettes :

$$(t_i^\alpha, \pi_i^\alpha), \quad \alpha = 1, 2, \dots, \alpha(i),$$

avec :

$$(24) \quad a_i \leq t_i^1 < t_i^2 < \dots < t_i^{\alpha(i)} \leq b_i,$$

et :

$$\pi_i^1 > \pi_i^2 > \dots > \pi_i^{\alpha(i)}.$$

Pour l'étiquette  $\alpha$ ,  $\pi_i^\alpha$  est la plus courte distance de  $s$  à  $i$  dans l'intervalle de temps :

$$(25) \quad \begin{array}{ll} [t_i^\alpha, t_i^{\alpha+1}) & \text{si } \alpha = 1, 2, \dots, \alpha(i) - 1, \\ [t_i^{\alpha(i)}, b_i] & \text{sinon.} \end{array}$$

La représentation graphique de cet ensemble prend la forme d'une fonction en escalier (fig. 3).

## 6.6. Description de l'algorithme

Nous allons décrire l'algorithme du chemin de longueur minimale sur un réseau avec contraintes d'horaires sur les nœuds de façon similaire à l'algorithme de Ford-Bellman-Moore.

Soit  $Q_i = (t_i^\alpha, \pi_i^\alpha)$ ,  $\alpha = 1, 2, \dots, \alpha(i)$  un ensemble irréductible d'étiquettes associé au nœud  $i$ . En ajoutant l'arc  $(i, j)$  aux chemins de  $s$  à  $i$ , on obtient un nouvel ensemble d'étiquettes au nœud  $j$  par l'application :

$$(26) \quad F_{ij}(Q_i) = \bigcup_{\alpha} f_{ij} \{ (t_i^\alpha, \pi_i^\alpha) \},$$

où :

$$(27) \quad f_{ij} \{ (t_i^\alpha, \pi_i^\alpha) \} = \begin{cases} \{ (\max[a_j, t_i^\alpha + t_{ij}], \pi_i^\alpha + c_{ij}) \} & \text{si } t_i^\alpha + t_{ij} \leq b_j \\ \emptyset & \text{sinon.} \end{cases}$$

La définition de  $f_{ij}$  permet un temps d'attente au nœud  $j$  et ce nouvel ensemble  $F_{ij}(Q_i)$  n'est pas nécessairement irréductible puisque les valeurs du temps  $t_j^\alpha$  peuvent s'empiler sur la valeur  $a_j$  pour les premières étiquettes :

$$t_j^\alpha = a_j, \quad \forall \alpha \text{ tel que } t_i^\alpha + t_{ij} < a_j.$$

Comme dans [8] on notera  $\overline{F_{ij}(Q_i) \cup Q_j}$  la réduction des deux ensembles en ne conservant que les étiquettes minimales.

*Remarque.* — Si l'attente au nœud  $j$  n'était pas permise, il faudrait conserver trois paramètres pour chaque étiquette  $a$  : l'intervalle pendant lequel le chemin est valide, soit  $[t_i^a, T_i^a]$ ; la longueur  $\pi_i^a$ . On définirait alors  $f_{ij}$  par :

$$(28) \quad f_{ij} \{([t_i^a, T_i^a]; \pi_i^a)\} = \begin{cases} \{([t_i^a + t_{ij}, T_i^a + t_{ij}] \cap [a_j, b_j]; \pi_i^a + c_{ij})\} \\ \quad \text{si } [t_i^a + t_{ij}, T_i^a + t_{ij}] \cap [a_j, b_j] \text{ non vide,} \\ \emptyset \quad \text{sinon} \end{cases} \dots]$$

0: Initialisation

$$Q_s = \{(t_s^1 = a_s, \pi_s^1 = 0)\}$$

$$Q_i = \{(t_i^1 = a_i, \pi_i^1 = \infty)\} \quad i \neq s$$

$$L = \{s\}$$

1: Exploration de  $F(i)$

Choisir en nœud  $i \in L$

$$\forall j \in F(i) \text{ tel que } \overline{F_{ij}(Q_i) \cup Q_j} \neq Q_j$$

$$\text{poser } Q_j = \overline{F_{ij}(Q_i) \cup Q_j}$$

$$L = L \cup \{j\}$$

2: Réduction de  $L$

$$L = L - \{i\}$$

Si  $L = \emptyset$  alors arrêter  
sinon retourner à 1.

**Algorithme pour le problème d'un plus court chemin  
avec contraintes d'horaires sur les nœuds.**

$L$  est l'ensemble des nœuds n'ayant pas été explorés après une modification. Le nœud pour exploration est choisi selon la stratégie FIFO et peut l'être selon l'implantation de D'Esopo-Pape [10].

Note : Le crochet indique la fin de la remarque.

Les étiquettes associées à un nœud sont placées dans une liste avec les temps en ordre croissant. Le calcul de  $\overline{F_{ij}(Q_i)} \cup Q_j$  au nœud  $j$  se fait par une mise à jour séquentielle de l'ensemble  $Q_j$  à partir de  $\Delta_i$ , un sous-ensemble de  $Q_i$ .  $\Delta_i$  contient les étiquettes modifiées depuis que  $i$  est dans  $L$ ; ce sont les seules susceptibles d'engendrer des modifications au nœud  $j$ . La mise à jour s'effectue en progressant dans les ensembles  $\Delta_i$  et  $Q_j$  et en comparant les longueurs des chemins de  $s$  à  $j$  utilisables au même instant.

La figure 4 illustre l'information à considérer dans un exemple de mise à jour. Les étiquettes de  $\Delta_i$ , représentées par des traits pleins, sont translatées :

$$(29) \quad (t_i, \pi_i) \rightarrow (t_i + t_{ij}, \pi_i + c_{ij}),$$

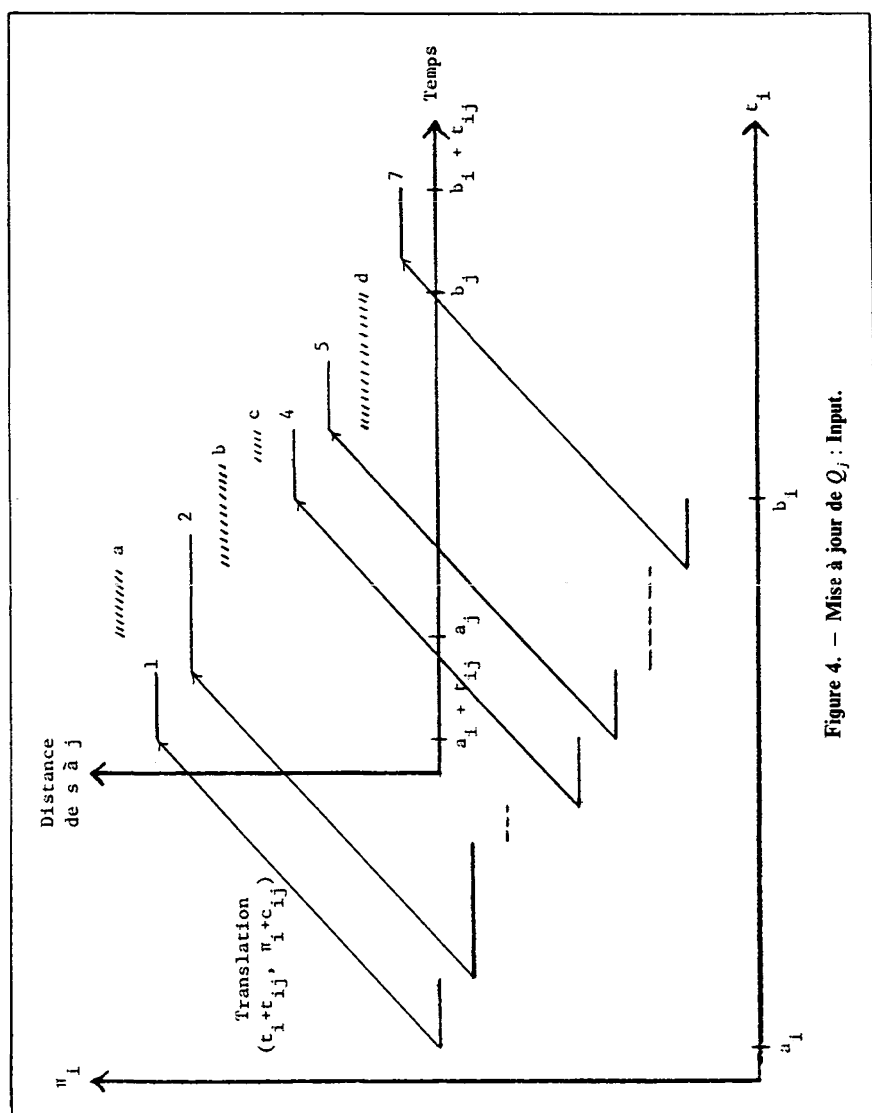
en vue d'ajouter l'arc  $(i, j)$  aux chemins de  $s$  à  $i$ . Les étiquettes de  $Q_i - \Delta_i$  et de  $Q_j$  sont respectivement représentées par des traits pointillés et hachurés.

Les comparaisons et modifications suivantes sont réalisées. Parmi les translations arrivant avant  $a_j$ , il suffit de comparer avec l'étiquette  $(a)$  seulement la dernière : elle est la plus courte. Dans l'exemple, la translation 2 *améliore* la composante longueur de l'étiquette  $(a)$ . Le chemin de  $s$  à  $j$  représenté par la translation 4 est meilleur, sur une portion du temps, que celui représenté par l'étiquette  $(b)$ ; l'étiquette  $(b)$  demeure et 4 *s'insère*. Cette nouvelle étiquette *élimine* l'étiquette  $(c)$  qu'elle améliore. Les translations 5 et 7 n'ont *pas d'effet* sur la liste  $Q_j$  puisque 5 n'améliore pas la longueur de  $(c)$  et que 7 excède l'intervalle de temps permis.

Cet exemple illustre les quatre résultats possibles de la comparaison d'une translation provenant de  $i$  et d'une étiquette de  $j$  : amélioration, insertion, élimination et pas d'effet. La figure 5 décrit le nouvel ensemble  $Q_j$ ; les étiquettes de  $\Delta_j$  et de  $Q_j - \Delta_j$  sont respectivement en traits pleins et pointillés, en supposant que  $\Delta_j$  était vide avant cette mise à jour.

### 6.7. Visites multiples à un nœud

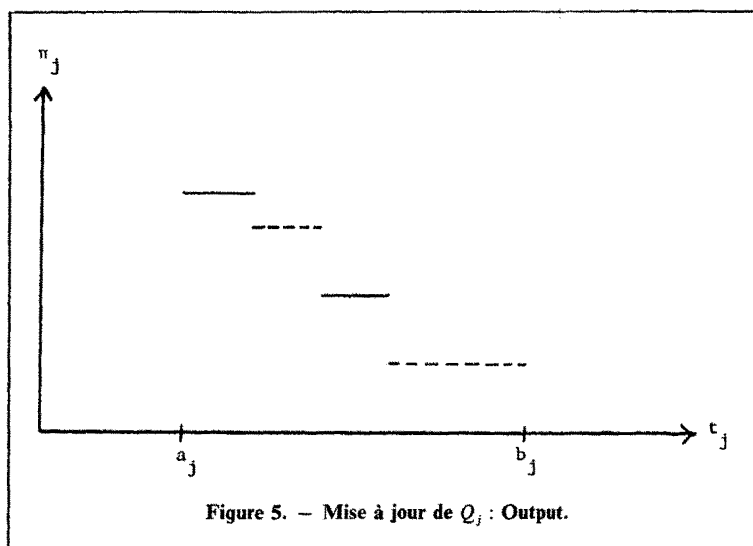
Lorsque le réseau présente des circuits négatifs, la méthode de résolution telle qu'elle est décrite n'assure pas des chemins sans circuit. Un chemin peut passer plusieurs fois à un nœud  $j$  à des temps différents en empruntant des étiquettes distinctes de l'ensemble  $Q_j$ . Le nombre de passages est cependant limité car tous les circuits s'effectuent dans l'intervalle de temps borné  $[a_j, b_j]$ . En présence de circuits négatifs, la complexité du problème atteint celle du problème du voyageur de commerce. Par contre, pour certaines variantes comme le problème de la tournée artistique, on accepte les solutions avec visites multiples : il est permis de donner plusieurs représentations au même site, à des temps différents.

Figure 4. — Mise à jour de  $Q_j$  : Input.

## 7. STRATÉGIE D'IMPLANTATION

Ce paragraphe a pour but de montrer l'importance de l'ordre d'exploration des nœuds. En effet, dans un réseau acyclique correspondant à un ensemble de tâches partiellement ordonnées (réseau PERT ou réseau de tâches avec horaire fixe), on peut numéroté les nœuds de telle sorte que si :

$$(30) \quad (i, j) \in A \quad \text{alors} \quad i < j,$$



et leur exploration dans cet ordre assure que chaque nœud est *inséré* dans la liste  $L$  une et une seule fois. Dans nos exemples réels où la largeur des fenêtres de temps ne dépasse pas 30 % de l'horizon, la numérotation des nœuds par ordre chronologique  $a_1 \leq a_2 \leq \dots \leq a_n$  assure que (30) est satisfaite par la très grande majorité des arcs et l'exploration des nœuds dans cet ordre réduit de façon importante les réinsertions et le temps de calcul. Le tableau I illustre ce fait sur trois problèmes réels de 273 nœuds où la numérotation initiale des nœuds est

TABLEAU I

*Effet de la numérotation chronologique des nœuds ( $n=273$ )*

Nombre d'arcs $ A $	Numérotation initiale des nœuds		$a_1 \leq a_2 \leq \dots \leq a_n$	
	Réinsertions I	Temps (sec) T	Réinsertions I	Temps (sec) T
14066	1549	4,40	0	1,66
15205	2109	8,36	0	2,07
16098	2580	12,64	3	2,61



celle fournie par l'organisme de transport. Pour chaque problème sont indiqués le nombre d'arcs  $|A|$ , le nombre  $I$  de réinsertions et le temps de calcul en secondes sur un ordinateur Cyber-173 de CDC.

## 8. RÉSULTATS NUMÉRIQUES

L'évaluation de la performance de l'algorithme s'est faite dans le cadre de l'application principale, c'est-à-dire le routage de véhicules satisfaisant des contraintes d'horaires et avec la possibilité d'attente en chaque nœud. Le réseau consiste en un ensemble de  $n$  tâches ou parcours à effectuer reliés à un dépôt de sortie  $s$  et d'entrée  $t$  des véhicules.

Nous avons reproduit aléatoirement des valeurs semblables aux données réelles; les tâches à effectuer sont relativement bien ordonnées dans le temps, et les contraintes d'horaires sont assez serrées. La structure fixe des données aléatoires comprend :

1° la dispersion physique des tâches : loi uniforme sur un carré de  $[0,75]$  de côté;

2° la durée  $d_i$  d'une tâche : loi uniforme sur  $[5,20]$ ;

3°  $t_{ij}, (i, j) \in A$  : distance euclidienne à laquelle est ajoutée la durée  $d_i$  de la tâche  $i$ ;  $c_{ij}, (i, j) \in A$  : distance euclidienne de laquelle est retranchée la constante 100 000;

4° la dispersion du début des fenêtres de temps  $a_i$  : loi uniforme sur  $[0, 100]$ .

La structure de coût négative correspond à l'effet du transfert des variables duales du problème de recouvrement au sous-problème de plus court chemin avec contraintes d'horaires dans une décomposition de Dantzig-Wolfe pour le routage des véhicules [2]. Les paramètres utilisés sont les suivants :

1° le nombre de nœuds  $n$  du réseau : 100, 200, 300;

2° le pourcentage de nœuds avec fenêtre de largeur non nulle : 50, 75 et 90 %;

3° la largeur  $b_i - a_i$  des fenêtres de temps, en pourcentage de l'horizon : 10, 20 et 30 %.

Le facteur principal du temps de calcul est sans contredit le nombre d'arcs  $|A|$  du réseau, auquel viennent s'ajouter les effets multiplicatifs de la division des étiquettes et du nombre de réinsertions  $I$  dans la liste de nœuds modifiés. En général, le temps de calcul  $T$  pour la solution d'un chemin le plus court est proportionnel au nombre d'arcs  $|A|$  plus le nombre d'explorations nouvelles de ces arcs :

$$(31) \quad T \propto |A| + I \star \frac{|A|}{n}.$$

Dans le cas du plus court chemin avec contraintes d'horaires sur les nœuds, on se doit d'introduire l'effet de la division des nœuds. Si on note par  $E/n$  le nombre moyen d'étiquettes par nœud lors de la résolution, la dernière relation devient :

$$(32) \quad T \propto \frac{E}{n} \star |A| + I \star \frac{E}{n} \star \frac{|A|}{n}.$$

Le tableau II présente quelques résultats numériques. Pour des horaires fixes ( $b_i = a_i, \forall i$ ), le réseau serait acyclique; dans ces tests, la croissance du pourcentage de nœuds avec fenêtre de largeur non nulle (50, 75 et 90 %)

TABLEAU II  
Tests paramétriques

Noeuds n	Largeur des fenêtres %	Noeuds avec fenêtre %	Arcs  A	Réinsertions I	Etiquettes par noeud E/n	Temps (sec) T
100	10	50	1541	0	1,33	0,16
		75	1625	1	1,45	0,17
		90	1740	1	1,69	0,20
	20	50	1865	5	1,85	0,22
		75	2052	7	2,30	0,29
		90	2272	10	3,03	0,35
	30	50	2271	35	2,48	0,42
		75	2584	36	3,52	0,58
		90	2817	48	4,54	0,77
200	10	50	6684	0	1,39	0,67
		75	7217	0	1,64	0,77
		90	7564	0	1,78	0,86
	20	50	7879	23	2,77	1,08
		75	9051	36	3,17	1,56
		90	9819	35	3,47	1,82
	30	50	9313	80	3,06	1,97
		75	11137	122	4,26	3,09
		90	12365	148	5,18	3,97
300	10	50	14635	3	1,66	1,54
		75	15809	4	2,05	1,84
		90	16519	4	2,54	2,18
	20	50	17684	52	2,69	2,78
		75	20166	63	3,52	3,84
		90	21784	72	4,58	5,02
	30	50	21215	161	3,60	5,00
		75	25280	209	5,05	7,90
		90	27930	débordement de mémoire (130 000 <sub>g</sub> )		

augmente la difficulté de résolution d'un problème. Pour les données générées, les fenêtres très étroites ( $b_i - a_i = 10\%$ ) réduisent le nombre de réinsertions à au plus 4, car les réseaux construits sont presque acycliques. Lorsque la largeur des fenêtres augmente, le nombre d'arcs du réseau satisfaisant l'équation (4) augmente, la subdivision des nœuds est plus fine et la structure acyclique du réseau est peu à peu détruite. Même si l'effort de calcul augmente avec la largeur des fenêtres, l'algorithme peut traiter de grands problèmes avec des fenêtres jusqu'à 30 minutes pour un horizon de 2 heures. Les réseaux traités sont alors contraints et presque acycliques, ce qui évite la prolifération d'étiquettes par nœud et réduit le nombre de mise à jour des étiquettes d'un nœud.

TABLEAU III  
*Effet de la largeur des fenêtres de temps ( $n=100$ )*

Largeur des fenêtres %	Arcs	Réinsertions	Etiquettes par noeud	Temps (sec)
	A	I	E/100	T
0	1250	0	1,00	0,11
10	1789	1	1,70	0,18
20	2340	16	2,90	0,32
30	3019	62	4,04	0,61
40	3868	161	6,16	1,34
50	4834	300	8,51	2,66
60	6054	535	10,86	5,53
70	7447	858	12,20	10,48
80	8762	1148	14,46	17,30
90	9633	1457	15,13	25,27
100	9874	1694	16,00	32,22

La seconde série de tests étudie la croissance du temps de calcul en fonction de la largeur des intervalles de temps pour un problème de  $n = 100$  tâches où il y a un pourcentage nul de tâches avec horaire fixe (tableau III). Le nombre d'arcs du réseau satisfaisant l'équation (4) est limité à 9 900, valeur qui n'est pas atteinte puisque certaines distances euclidiennes excèdent 100 unités. La croissance du temps de calcul (0,11 à 32,22 secondes) est exponentielle, et la proportionnalité de la relation (32) est vérifiée dans un modèle de régression linéaire avec un  $R^2 = 0,9988$ . Le nombre moyen d'étiquettes par nœud est assez restreint, même pour le réseau où la fenêtre de flexibilité associée à chaque nœud est l'horizon total (16 étiquettes/nœud). Ceci s'explique en grande partie par le fait que l'autorisation d'attente permet d'éliminer beaucoup d'étiquettes non minimales.

Le nombre de réinsertions (0 à 1964) est très lié à la structure acyclique ou non du réseau. En faisant l'examen de ces temps de calculs, on doit se rappeler que dans l'application du routage de véhicules, la largeur des intervalles de temps est limitée à environ 30 % de l'horizon.

Finalement, nous avons évalué la performance de l'algorithme sur le réseau de 100 nœuds et 2 340 arcs du tableau III. Le but de ces tests est de mesurer l'effet de la largeur des fenêtres de temps sur le nombre de réinsertions et sur le nombre d'étiquettes par nœud, tout en maintenant fixe le nombre d'arcs. Le tableau IV présente les résultats. Le réseau est presque sans circuit d'où un nombre de réinsertions très réduit. Le nombre moyen d'étiquettes par nœud reste en deçà de 8 et les temps de calculs sont inférieurs à une seconde. Cet essai démontre clairement que l'augmentation du temps de calcul lorsqu'on élargit les fenêtres de temps est principalement due à l'augmentation de la taille du réseau.

TABLEAU IV

*Effet de la largeur des fenêtres de temps lorsque le nombre d'arcs est fixe.*

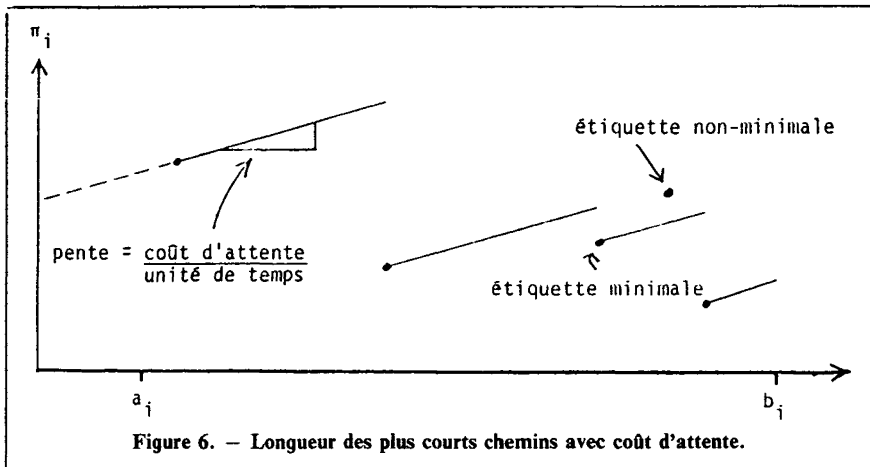
Largeur des fenêtres %	Réinsertions I	Etiquettes par nœud E/n	Temps (sec) T
0	0	1	0,15
10	1	1,66	0,20
20	16	2,90	0,32
30	19	3,78	0,45
40	21	4,56	0,54
50	23	5,09	0,60
60	23	5,66	0,66
70	25	6,36	0,74
80	26	7,13	0,79
90	32	7,74	0,86
100	37	7,80	0,87

## 9. EXTENSIONS

Il est facile de généraliser l'algorithme pour le problème de plusieurs intervalles de temps disjoints à chaque nœud. La méthode proposée s'applique aussi pour traiter l'approximation du problème où le temps permis de visite à un nœud est une suite de valeurs discrètes au lieu d'être une variable continue. Elle s'applique directement si les données sont déjà discrétisées de façon à ce

qu'apparaissent uniquement les valeurs discrètes de temps de passage (ex. : valeurs entières, multiples de 5, 10 ou 15 minutes). Si les données ne sont pas convenablement discrétisées, il faut ajouter à l'algorithme des procédures d'arrondi pour que les temps de passage aux nœuds ne prennent que les valeurs discrètes permises. La discrétisation du problème peut être utilisée pour obtenir une solution approximative tout en limitant la longueur des listes d'étiquettes. En effet, la longueur de la liste d'étiquettes à un nœud est inférieure ou égale au nombre de valeurs permises pour le temps, et est généralement beaucoup moindre avec la création et l'élimination dynamique des étiquettes.

Lorsque le problème comporte un coût proportionnel au temps d'attente avant la visite d'un nœud, l'algorithme s'adapte facilement en modifiant la relation d'ordre. La figure 6 illustre cette nouvelle relation et les propriétés qui en découlent; cet exemple présente le coût d'un plus court chemin en fonction de l'heure de visite.



## 10. CONCLUSION

Le traitement efficace du problème de plus court chemin avec horaires ouvre une nouvelle voie pour résoudre les problèmes de fabrication de routes de véhicules avec contraintes d'horaires, de capacités ou de durées [2]. Ces problèmes lorsqu'ils sont de grande taille se traitent par décomposition pour profiter des algorithmes efficaces sur les réseaux. Jusqu'ici, les approches par décomposition séparent les contraintes additionnelles et le problème de réseau. Il est alors très difficile, à partir des solutions entières du problème de réseau, de construire des solutions entières satisfaisant en plus les contraintes

additionnelles. Avec le problème du plus court chemin avec contraintes d'horaires, on obtient une solution entière satisfaisant une partie des contraintes additionnelles en plus de celles du réseau. Il est alors plus facile d'obtenir une solution entière satisfaisant l'ensemble des contraintes.

## BIBLIOGRAPHIE

1. R. E. BELLMAN, *On a Routing Problem*, Quart. Appl. Math., 16, 1958, p. 87-90.
2. J. DESROSIERS, F. SOUMIS et M. DESROCHERS, *Routing with Time Windows by Column Generation*, Rapport de recherche G83-15, École des Hautes Études Commerciales, Université de Montréal, 1983 (soumis pour publication à NETWORKS).
3. E. W. DIJKSTRA, *A Note on Two Problems in Connexion with Graph*, Numer. Math., vol. 1, 1959, p. 269-271.
4. L. R. FORD, Jr : *Network Flow Theory*, The Rand Corporation, vol. 293, 1956.
5. G. Y. HANDLER et I. ZANG, *A Dual Algorithm for the Constrained Shortest Path Problem*, Networks, vol. 10, 1980, p. 293-310.
6. H. C. JOKSCH, *The Shortest Route Problem with Constraints*, J. Math. Anal. Appl., vol. 14, 1966, p. 191-197.
7. C. E. MILLER, A. W. TUCKER et R. A. ZEMLIN, *Integer Programming Formulation of Travelling Salesman Problems*, ACM, vol. 7, 1960, p. 326-329.
8. M. MINOUX, *Plus court chemin avec contraintes : algorithmes et applications*, Annales des télécommunications, tome 30, 1975, p. 383-394.
9. E. F. MOORE, *The Shortest Path Through a Maze* (Proc. of international Symposium on the Theory of Switching, Part II, Apr. 2-5, 1957, Harvard University Press, Cambridge, Mass., 1959).
10. S. PALLOTTINO, *Adaptation de l'algorithme de D'Esopo-Pape pour la détermination de tous les chemins les plus courts : améliorations et simplifications*, publication n° 136, Centre de recherche sur les transports, Université de Montréal, 1979.