RAIRO. RECHERCHE OPÉRATIONNELLE

M. HAOUARI

P. DEJAX

Plus court chemin avec dépendance horaire : résolution et application aux problèmes de tournées

RAIRO. Recherche opérationnelle, tome 31, n° 2 (1997), p. 117-131

<http://www.numdam.org/item?id=RO_1997__31_2_117_0>

Recherche opérationnelle/Operations Research
(vol. 31, n° 2, 1997, pp. 117-131)

PLUS COURT CHEMIN AVEC DÉPENDANCE HORAIRE : RÉSOLUTION ET APPLICATION AUX PROBLÈMES DE TOURNÉES (*)

par M. HAOUARI ([1]) et P. DEJAX ([1])

Résumé. – *Nous résolvons le problème du plus court chemin avec dépendance horaire. Il s'agit dans ce cas, de tenir compte des fenêtres de temps, et en plus du fait que le coût et la durée de parcours varient en fonction du temps. Deux versions de ce problème sont étudiées : avec ou sans contrainte supplémentaire de type « sac à dos ». Nous montrons une application de ce problème à la résolution optimale des problèmes de tournées de véhicules avec dépendance horaire par la méthode de génération de colonnes.*

Mots clés : Chemin, fenêtre de temps, génération de colonnes, graphe, recherche arborescente, relaxation lagrangienne, tournées de véhicules.

Abstract. – *We solve the time dependent shortest path problem. This problem consists in finding a minimum weight path through a network with respect to time windows constraints, and time varying costs and travelling times. Two issues of this problem are studied: without and with a knapsack type constraint. We show an important application of this problem for the design of an optimal column generation based algorithm for a time dependent vehicle routing problem.*

Keywords: Path, time window, column generation, graph, branch and bound, lagrangian relaxation, vehicle routing.

1. INTRODUCTION

Soit un graphe orienté $G(X, A)$ avec $X = N \cup \{s, t\}$ où $N = \{1, \ldots, n\}$. On associe à chaque nœud $i \in X$ une fenêtre de temps $[a_i, b_i]$ indiquant l'intervalle horaire pendant lequel ce nœud peut être traversé. A chaque arc $(i, j) \in A$ est associé un temps de parcours $t_{ij}(t_i)$ où t_i est l'instant de départ à partir de i, le coût de ce parcours est $c_{ij}(t_i)$.

(*) Manuscrit reçu en avril 1994.

([1]) École Centrale Paris, Laboratoire Économique Industriel et Social, 92295 Châtenay-Malabry Cedex, France. L'adresse actuelle du premier auteur est : École Nationale d'Ingénieurs de Tunis, Département de Génie Industriel, BP 37, Tunis, 1002 Tunisie.

Recherche opérationnelle/Operations Research, 0399-0559/97/02/$ 7.00

Le coût d'un chemin est défini comme étant la somme des coûts des arcs formant ce chemin. Le problème du plus court chemin avec dépendance horaire (PCCDH) consiste à déterminer dans le graphe G un chemin de coût minimal entre deux nœuds donnés, en respectant les contraintes de fenêtres de temps. Ce problème constitue une généralisation du plus court chemin avec fenêtre de temps (PCCFT) déjà traité dans la littérature scientifique, par le fait que les durées de parcours et les coûts des arcs varient en fonction du temps et ne sont pas fixes.

Cet article est organisé comme suit : dans la section 2, nous présentons une formulation mathématique de ce problème et nous montrons qu'il est possible de le résoudre en un temps pseudo-polynomial. Dans la section 3, nous étudions une extension du PCCDH avec prise en compte d'une contrainte supplémentaire de type « sac à dos ». Dans la section 4, nous étudions une importante application de ce dernier problème à la résolution optimale des problèmes de tournées avec dépendance horaire. Enfin, dans la section 5, nous présentons un ensemble de tests numériques permettant une évaluation pratique de l'efficacité des algorithmes présentés.

2. FORMULATION MATHÉMATIQUE ET RÉSOLUTION DU PCCDH

Une formulation du PCCDH où x_{ij} est la valeur flot sur l'arc $(i, j) \in A$ et t_i est l'instant de départ du nœud i est la suivante :

$$\text{Min } \Sigma_{(i,j)\in A}\, c_{ij}(t_i)\, x_{ij} \tag{1}$$

$$\Sigma_j x_{ij} - \Sigma_j x_{ji} = \begin{cases} 1 & i = s \\ 0 & i \in N \\ -1 & i = t \end{cases} \tag{2}$$

$$t_i + t_{ij}(t_i) - t_j \le M(1 - x_{ij}) \qquad (i, j) \in A \tag{3}$$

$$a_i \le t_i \le b_i \qquad i \in X \tag{4}$$

$$x_{ij} = 0 \text{ ou } 1 \qquad (i, j) \in A \tag{5}$$

Grâce aux contraintes (2) et (5) le vecteur $X_{st} = (x_{ij})$ représente le vecteur d'incidence d'un chemin élémentaire entre s et t, et les contraintes (3) et (4) imposent le respect des fenêtres de temps (M est un « grand » nombre). La contrainte (3) est équivalente à la contrainte suivante :

$$x_{ij} = 1 \quad \Rightarrow \quad t_i + t_{ij}(t_i) \le t_j \qquad (i, j) \in A \tag{3'}$$

Malgré son intérêt pratique et théorique, il n'existe pas à notre connaissance de travaux de recherche traitant du PCCDH. Cependant, le seul aspect des durées et coûts variant en fonction du temps a été étudié par Cook et Halsey (1966), Dreyfus (1968), de Palma *et al.* (1987), et Hall (1986). D'autre part, le seul aspect des contraintes de fenêtres de temps a été étudié par Halpern et Priess (1974), Minoux (1976), et Desrosiers *et al.* (1983).

Dans ce qui suit nous allons montrer que le PCCDH peut être transformé et résolu comme un problème du plus court chemin classique (sans contraintes additionnelles).

PROPOSITION 1 : *Le PCCDH peut être résolu en un temps pseudo-polynomial.*

Preuve : Soit le PCCDH défini entre les nœuds s et t dans le graphe $G(X, A)$. Pour tout $i \in X$, il existe une fenêtre de temps $[a_i, b_i]$. Le temps de parcours entre deux nœuds quelconques i et j est $t_{ij}(t_i)$ où t_i est l'instant de départ à partir de i, le coût de ce parcours est $c_{ij}(t_i)$. Chaque intervalle $[a_i, b_i]$ peut être partitionné en $P_i = b_i - a_i + 1$ périodes de temps, une période p_i ($p_i = 1 \ldots P_i$) correspond à l'instant $t_i = a_i + p_i - 1$.

Construisons un graphe transformé $G'(X', A')$ où $X' = N' \cup S' \cup T'$ avec :

$$N' = \{(i, p_i), p_i = 1 \ldots P_i, i = 1 \ldots n\}$$

$$S' = \{(s, p_s), p_s = 1 \ldots P_s\}$$

$$T' = \{(t, p_t), p_t = 1 \ldots P_t\}$$

Pour tout i et $j \in X$, les nœuds (i, p_i) et (j, p_j) sont reliés si en partant du nœud i à la période p_i, il est possible d'atteindre le nœud j à la période p_j, donc si la relation suivante est vérifiée :

$$a_j + p_j - 1 = \mathrm{Max}(a_j, a_i + p_i - 1 + t_{ij}(a_i + p_i - 1)) \tag{6}$$

le coût d'un tel arc est $c_{ij}(a_i + p_i - 1)$.

Maintenant, incluons dans notre graphe G' deux nœuds s^* et t^*, que nous relions respectivement à tous les nœuds (s, p_s) et (t, p_t) et affectons ces arcs d'un coût nul. Alors le plus court chemin entre s^* et t^* dans G' à le même coût que le plus court chemin avec contraintes horaires entre s et t dans le graphe G. D'autre part, la succession des nœuds (i, p_i) nous indique l'instant d'entrée à chaque nœud.

Le cardinal de X' est de l'ordre de $O(nd^*)$ où d^* est la largeur moyenne des fenêtres de temps. La résolution du plus court chemin dans G' peut alors être faite avec une complexité en $O(n^2 d^{*2})$.

CQFD

D'après la proposition précédente, il apparaît que même si le PCCDH est un problème NP-dur (De Palma *et al.*, 1987), il est possible de le résoudre par la programmation dynamique en un temps polynomial, pour une largeur moyenne des fenêtres de temps fixée.

Le graphe G' est nécessairement sans circuit, puisqu'un chemin dans G' correspond à une succession de nœuds relatifs à des périodes horaires strictement croissantes.

Évaluons maintenant la complexité de la résolution du PCCDH. Cette résolution requiert deux étapes : la construction du graphe G', et la résolution du plus court chemin dans G'. Nous allons analyser ces deux étapes :

(i) Construction du graphe transformé :

Nous partons du graphe $G(X, A)$ avec $|X| = n+2$ et $|A| = M$. Deux nœuds i et j sont reliés si les contraintes de fenêtres de temps permettent d'atteindre j en partant de i pendant un instant donné compris dans la fenêtre de temps du nœud i. Ainsi, à chaque nœud i sont définis $\Gamma(i)$ successeurs.

Pour construire les arcs de G' plusieurs comparaisons sont nécessaires. Pour chaque période p_i d'un nœud i, $\Gamma(i)$ opérations sont nécessaires pour décider à quels successeurs il faudrait relier ce nœud. Si nous faisons la somme pour tous les nœuds il apparaît que $\sum_{i \in N} P_i \Gamma(i)$ opérations sont nécessaires, une borne supérieure de cette quantité est $P^* M$ (où $P^* = \max_{i \in N} P_i$). Cette étape nécessite donc au plus $O(P^* M)$ opérations.

(ii) Recherche du plus court chemin dans G'

Le graphe G' étant sans circuit, dans ce cas l'algorithme de Ford converge en une seule itération, si les nœuds sont examinés dans un ordre topologique croissant. La définition d'un ordre topologique peut être faite en $O(M')$, où M' est le nombre d'arcs de G' (*cf.* Gondran et Minoux, 1986, ch. 2). De même que la complexité de l'algorithme de Ford dans le cas d'un graphe sans circuit est $O(M')$. Une borne supérieure de M' est $P^* M$. Cette étape nécessite donc au plus $O(P^* M)$ opérations.

3. LE PCCDH AVEC UNE CONTRAINTE DE TYPE « SAC À DOS »

Considérons à présent une extension du PCCDH, en associant à chaque nœud $i \in N$ un nombre positif d_i (ce nombre représente la « demande » du nœud i), et en imposant la contrainte supplémentaire suivante :

$$\Sigma_{(i,j)\in A}\, d_i\, x_{ij} \leq D \qquad i \in N \tag{7}$$

où D est un paramètre représentant la « capacité » maximale d'un chemin. Nous désignons le nouveau problème ainsi obtenu par le problème du plus court chemin avec dépendance horaire et contrainte de capacité (PCCDHC).

Puique nous avons prouvé que le PCCDH peut être résolu comme un problème de plus court chemin, alors nous pouvons conclure que le PCCDHC peut être transformé et résolu comme un problème du plus court chemin avec une contrainte de capacité (PCCC). Handler et Zang (1980) ont montré que ce problème est NP-dur.

Deux types d'approches ont été proposées pour la résolution du PCCC. La première est basée sur la programmation dynamique par une généralisation des algorithmes classiques de Ford (Joksch, 1966) et Dijkstra (Minoux, 1975). La deuxième approche est basée sur la relaxation lagrangienne. Elle a été utilisée la première fois par Minoux (1975) pour l'obtention d'une solution approchée, ensuite par Handler et Zang (1980) qui ont montré comment il est possible de l'utiliser pour l'obtention de la solution optimale. Décrivons brièvement l'approche basée sur la relaxation lagrangienne.

Soit Q l'ensemble des chemins entre s' et t', associons un multiplicateur de Lagrange $\lambda \geq 0$ à la contrainte de capacité définie par (7), le problème dual lagrangien associé au PCCC revient à rechercher λ^* maximisant la fonction de Lagrange :

$$(\text{DPCCC}) \qquad \underset{\lambda \geq 0}{\text{Max}}\ L(\lambda) = \underset{x \in Q}{\text{Min}}\ (cx + \lambda\,(dx - D)) \tag{8}$$

où c et d représentent respectivement les vecteurs coût et demande.

Pour tout $\lambda \geq 0$, calculer $L(\lambda)$ revient donc à résoudre un problème du plus court chemin entre s et t avec les coûts modifiés $(c + \lambda\, d)$. Rappelons que la fonction de Lagrange $L(\lambda)$ est une fonction à une variable, concave et linéaire par morceaux. La concavité de $L(\lambda)$ assure que tout optimum local est un optimum global. D'autre part, soient $c(\lambda)$ et $d(\lambda)$ respectivement le coût et la demande totale du plus court chemin obtenu avec les coûts modifiés $(c + \lambda\, d)$. Il est facile de voir que $c(\lambda)$ et $d(\lambda)$ sont respectivement une fonction monotone croissante et monotone décroissante. Dans la méthode proposée par Minoux (1975), la fonction $L(\lambda)$ est optimisée à l'aide d'un principe dichotomique. Partant d'un intervalle $[\lambda_1, \lambda_2]$ dont on sait qu'il contient la solution optimale λ^*, on réduit progressivement cet intervalle en utilisant d'une part le fait que $c(\lambda)$ et $d(\lambda)$ sont monotones, et d'autre part, le fait que λ^* peut être défini par l'intersection de deux droites d'équations :

$$z = cx_1 + \lambda(dx_1 - D) \qquad \text{et} \qquad z = cx_2 + \lambda(dx_2 - D)$$

où x_1 et x_2 sont deux chemins distincts. L'algorithme est explicité de la manière suivante :

Étape 1 : Initialisation

Déterminer le plus court chemin avec les coûts modifiés $(c + \lambda d)$, en fixant λ successivement à $\lambda_1 = 0$ et $\lambda_2 = \infty$ (ce qui revient dans ce dernier cas à déterminer le chemin ayant une demande totale minimale).

Étape 2 :

Soit λ' le point d'intersection des droites d'équations :

$$z = c(\lambda_1) + \lambda(d(\lambda_1) - D) \quad \text{et} \quad z = c(\lambda_2) + \lambda(d(\lambda_2) - D)$$

nous avons donc : $\lambda' = (c(\lambda_1) - c(\lambda_2))/(d(\lambda_2) - d(\lambda_1))$

Étape 3 :

Déterminer le plus court chemin avec les coûts modifiés $(c + \lambda' d)$

si $L(\lambda') = c(\lambda_1) + \lambda'(d(\lambda_1) - D) = c(\lambda_2) + \lambda'(d(\lambda_2 - D)$

alors $\lambda' = \lambda^*$, FIN

sinon si $d(\lambda') > D$, faire $\lambda_1 = \lambda'$, aller à l'étape 2

si $d(\lambda') < D$, faire $\lambda_2 = \lambda'$, aller à l'étape 2

La procédure nécessite donc la résolution de problèmes de plus court chemin classique avec les coûts modifiés $(c + \lambda d)$, pour des valeurs de λ bien choisies. Après convergence on obtient une solution réalisable associée à la solution optimale duale, et un encadrement de la solution optimale (qui est ainsi comprise entre $L(\lambda^*)$ et $c(\lambda_2)$).

Si la solution réalisable obtenue n'est pas optimale, il est possible de trouver la solution optimale en utilisant une procédure de classement (Handler et Zang, 1980). En effet, soit B le coût de la meilleure solution primale réalisable obtenue dans l'étape 3 de l'algorithme dichotomique, et soit λ^* la solution duale optimale. Alors, on génère les chemins $x_2, x_3, \ldots, x_k$ [x_k correspond au k-ième plus court chemin avec les coûts modifiés $(c + \lambda^* d)$]. Chaque fois où cela s'avère possible, on améliore la valeur de la borne supérieure B. La fonction duale au point x_k étant :

$$L_k(\lambda^*) = cx_k + \lambda^*(dx_k - D)$$

Alors, on montre que dès qu'il existe un chemin x_k vérifiant $L_k(\lambda^*) > B$, alors la solution primale réalisable ayant servi à la dernière mise à jour de B est la solution optimale. La recherche du k ième plus court chemin peut être efficacement réalisée grâce à l'algorithme de Dreyfus (1968). Cet algorithme basé sur la programmation dynamique est le suivant :

– $P_l(j)$ = l-ième plus court chemin entre s et j, son coût étant noté $Z(P_l(j))$
– $n(i, j, l)$ = nombre de chemins parmi $P_1(j), \ldots, P_l(j)$ contenant l'arc (i, j)
Initialisation :
   – *Calcul de* $P_1(j)$ *et* $n(i, j, 1)$
étape courante
   – *faire pour* $l := 2$ *jusqu'à* k

$$Z(P_l(j) = \min_{i \in \Gamma^{-1}(j)} (Z(P_{n(i,j,l-1)}(i) + c_{ij})$$

$$n(i, j, l) = n(i, j, l-1) + 1 \qquad si \quad (i, j) \in P_l(j)$$
$$= n(i, j, l-1) \qquad sinon$$

fin faire

4. APPLICATION AUX PROBLÈMES DE TOURNÉES AVEC FENÊTRES DE TEMPS

4.1. Présentation du problème

L'application qui a été à l'origine de la recherche sur le problème du plus court chemin avec dépendance horaire et contrainte de capacité, concerne la résolution optimale d'un problème de tournées particulièrement complexe. Il s'agit du cas où un ensemble de clients ayant une demande connue, doivent être servis par une flotte de véhicules de capacité finie. Les visites aux clients ne sont autorisées que dans des intervalles horaires (fenêtres de temps) connus à l'avance. D'autre part, la durée d'un parcours et son coût varient en fonction du temps, et ne sont pas fixes comme il est de tradition dans ce type de problèmes. L'objectif consiste à déterminer la tournée effectuée par chaque véhicule de manière à servir l'ensemble des clients au moindre coût global. Le coût global étant formé par la somme des coûts de parcours et éventuellement des coûts fixes rattachés à l'utilisation des véhicules. Nous désignons ce problème par le problème de tournées de véhicules avec dépendance horaire (PTVDH).

La dépendance horaire rend compte des effets de congestion en distribution urbaine. En effet, si nous négligeons cet aspect, la solution obtenue peut être sous-optimale, ou même irréalisable en cas d'existence de fenêtres de temps relativement étroites. Malgré son intérêt pratique et théorique il n'existe pratiquement pas de travaux traitant du cas particulier des tournées de véhicules avec dépendance horaire. Le seul travail de recherche dont nous soyons au courant est celui de Malandraki (1989), qui a développé un ensemble de méthodes heuristiques.

4.2. Formulation mathématique du PTVDH

Nous utilisons une formulation de type problème de partitionnement. Nous introduisons la notation suivante :

T : ensemble des tournées réalisables (*i.e.* vérifiant les contraintes de fenêtres de temps et de capacité).

y_r : variable binaire qui vaut 1 si la tournée r est effectuée, sinon 0.

a_{ir} : constante binaire quit vaut 1 si le client i est visité dans la tournée r, sinon 0.

c_r : coût de la tournée r, c'est la somme du coût fixe et du coût de roulage.

Le problème peut alors être formulé de la manière suivante :

$$\text{Min} \sum_{r \in T} c_r \, y_r \tag{9}$$

(PP)

$$\sum_{r \in T} a_{ir} \, y_r = 1, \qquad i = 1 \dots n \tag{10}$$

$$y_r = 0 \text{ ou } 1, \qquad r \in T, \tag{11}$$

La fonction objectif (9) consiste à minimiser le coût total d'exploitation (coûts fixes et coûts variables). La contrainte (10) impose que chaque client doit être visité exactement une fois, et la contrainte (11) est la contrainte d'intégrité des variables. Nous avons ainsi formulé le PTVDH comme un problème de partitionnement de très grande taille, puisqu'il contient autant de colonnes que de tournées réalisables. Nous résolvons le PTVDH par un algorithme de recherche arborescente, où l'obtention d'une borne inférieure est faite par la résolution de la relaxation linéaire de (PP).

4.2.2. *Obtention d'une borne inférieure*

Si dans ce modèle (PP) nous relaxons la contrainte d'intégrité des variables, alors, la solution du problème relaxé (PP_L), constitue une borne inférieure de la solution optimale entière, et peut être par conséquent utilisée dans une procédure de recherche arborescente. Cette borne inférieure est obtenue après la résolution de (PP_L) à l'aide d'un algorithme de génération de colonnes [pour un exposé relatif à cette technique le lecteur pourra se reporter à

Minoux (1983)]. La recherche d'une colonne Y_k de coût réduit minimal revient à la résolution du sous-problème suivant :

Trouver une colonne Y_k telle que :

$$C_k - \pi^t Y_k = \underset{r \in T}{\mathrm{Min}} (C_r - \pi^t Y_r) \qquad \text{(PGC)}$$

où π est la solution duale optimale du problème restreint.

Si nous créons à partir du dépôt deux nœuds s et t représentant respectivement la sortie et l'entrée du dépôt, alors une tournée est équivalente à un chemin entre s et t. Résoudre le (PGC) revient donc à trouver le chemin de coût réduit minimal. Les contraintes du problème de tournée sont transférées au sous-problème. Ainsi, dans le cas du PTVDH, le sous-problème qui consiste à fournir la colonne de coût marginal minimal, est le problème du plus court chemin avec fenêtres de temps, coûts et durée de parcours dépendant du temps, et contrainte de capacité. Il s'agit bien dans ce cas de résoudre le PCCDHC étudié dans la section précédente. Les arcs du graphe étant affectés des coûts réduits $c_{ij}^* (t_i) = c_{ij} (t_i) - \pi_i$. Les coûts fixes des véhicules peuvent être simplement pris en compte en les ajoutant à tous les arcs issus du nœud s (représentant le dépôt).

4.3. Choix de la variable à séparer, règle de séparation, et stratégie de branchement

Si la solution obtenue par la relaxation linéaire n'est pas entière, un processus de recherche arborescente est alors utilisé. Il s'agit de sélectionner une variable de base fractionnaire et de générer à partir du problème initial deux sous-problèmes en fixant la variable sélectionnée à 0 ou à 1. Une règle de sélection consiste à choisir la variable de base fractionnaire ayant le nombre minimal de coefficients nuls dans la colonne correspondante, dans ce cas cette tournée partage le maximum de clients avec les autres tournées, et les deux sous-problèmes générés seraient fortement différenciés. La résolution de ces deux sous-problèmes est faite de la manière suivante :

(i) Si nous fixons la variable sélectionnée à 1, nous obtenons un problème de taille plus réduite que le problème initial, puisqu'on élimine de notre graphe tous les clients servis dans la tournée sélectionnée. L'algorithme permettant d'obtenir la colonne de coût marginal minimal reste ainsi inchangé.

(ii) Si nous fixons la variable sélectionnée à 0, la colonne correspondante doit être retirée du tableau du simplexe, et il faut s'assurer que cette tournée ne sera plus générée. Nous pouvons alors utiliser une idée similaire à celle de Ribeiro *et al.* (1989). Cette idée consiste à utiliser une procédure de classement de la manière suivante : si à une itération donnée la colonne correspondante à un chemin le plus court entre s et t appartient à l'ensemble des tournées interdites (fixées à 0), nous devons chercher le 2-ième plus court chemin, 3-ième, ..., k-ième jusqu'à ce que le chemin obtenu n'ait pas de tournée correspondante interdite. Ce processus est convergent, puisque le nombre de chemins différents est fini, et le nombre de tournées interdites est fini. Il est facile de voir que l'algorithme basé sur la relaxation lagrangienne présenté dans la section 3 peut être étendu pour générer une séquence de chemins réalisables de coûts croissants. En effet, supposons que pendant le processus de génération des k plus courts chemins $\{x_0, x_1, \cdots, x_{k-1}\}$, un sous-ensemble $R = \{y_1, y_2, \ldots, y_r\}$ de chemins réalisables ont été obtenus, ayant pour coûts respectifs $\{c_1, c_2, \ldots, c_r\}$. Ces chemins sont classés par coûts croissants. Notons que :

$$L_k(\lambda^*) \leq cx + \lambda^*(dx - D) \quad \text{pour tout } x \in Q - \{x_0, x_1, \ldots, x_{k-1}\}$$

$L_k(\lambda^*)$ constitue donc une borne inférieure pour tous les chemins réalisables non encore générés. Donc si $L_k(\lambda^*) > c_q$ (avec $1 \leq q \leq r$), alors les q chemins réalisables $y_1, y_2, \ldots, y_q$ sont respectivement le chemin réalisable le plus court, le second chemin réalisable le plus court, ..., et le q-ième chemin réalisable le plus court.

Ainsi, en poursuivant le processus de génération des plus courts chemins, il est possible de déterminer la séquence des plus courts chemins réalisables par coûts croissants.

En ce qui concerne la stratégie de branchement, nous appliquons le critère classique qui consiste à traiter en priorité le nœud ayant la plus faible borne inférieure.

5. TESTS NUMÉRIQUES

Les tests numériques qui suivent ont pour but d'évaluer la performance de l'algorithme de résolution du problème du plus court chemin avec dépendance horaire et contrainte de capacité, et également de tester l'efficacité de la procédure de classement des chemins réalisables. Nous nous sommes en particulier intéressé à l'examen de la variation du temps de calcul en fonction du nombre de nœuds, et du nombre de chemins à réaliser.

Construction des problèmes tests

Nous générons aléatoirement le graphe correspondant au graphe initial transformé. Le graphe à construire doit être connexe et sans circuit. Afin d'éviter d'avoir à chaque fois à déterminer la fonction rang du graphe, nous le construisons de telle manière que les nœuds soient indexés, d'après leur ordre topologique croissant.

Un problème est défini par la donnée des deux paramètres n et δ, représentant respectivement le nombre total de nœuds (y compris les nœuds « origine » et « destination » respectivement désignées par 1 et n), et la densité du graphe, mesurée comme étant le rapport entre le nombre total d'arcs du graphe et le nombre maximal d'arcs qu'un 1-graphe de n nœuds peut contenir [soit $n(n-1)$]. Nous procédons en deux étapes de la manière suivante :

(i) Nous commençons par relier par un arc toutes les paires de nœuds de la forme $(i, i+1)$, pour tout nœud i tel que $1 \leq i \leq n-1$. A partir de ce moment, l'indice d'un nœud est aussi son ordre topologique, en effet l'ordre du nœud i est égal au nombre d'arcs d'un chemin de cardinalité maximale entre les nœuds 1 et i.

(ii) Afin d'assurer une structure sans circuit et préserver l'ordre topologique déjà défini, nous ne relions un nœud i $(1 \leq i \leq n-2)$ qu'à un nœud j ayant un indice plus élevé $(i+2 \leq j \leq n)$. Cette liaison est faite avec une probabilité p, que nous calculons en fonction de n et δ.

Le nœud i $(1 \leq i \leq n-2)$ peut avoir au maximum $(n-i-1)$ successeurs. En effectuant la liaison avec une probabilité p, l'espérance mathématique du nombre de successeurs de i est $p(n-i-1)$. Ainsi, en sommant sur i variant de 1 à $n-2$, nous obtenons l'espérance mathématique du nombre total d'arcs formés de cette manière, ce nombre est égal à $p(n-1)(n-2)/2$. En ajoutant les $(n-1)$ arcs construits à l'étape (i), nous obtenons un nombre total d'arcs égal à $(p(n-2)(n-1)/2+(n+1))$. Or le nombre total d'arcs du graphe est égal à $\delta n(n-1)$. La probabilité p vaut donc :

$$p = 2(\delta n - 1)/(n-2)$$

(avec $\delta \leq 1/2$ puisque le graphe est sans circuit).

Une fois le graphe construit, les arcs sont affectés d'un coût compris entre 1 et 100, pris aléatoirement suivant une répartition uniforme. De même, les demandes sont toutes comprises entre 1 et 20, et sont également distribuées aléatoirement suivant une répartition uniforme.

La limite de capacité D est prise égale à la partie entière de la moyenne arithmétique de la demande totale du plus court chemin sans contraintes, et de celle du chemin de capacité totale minimale. Cette limite de capacité nous garantit d'une part que notre problème est réalisable et d'autre part que le plus court chemin recherché à une demande totale sensiblement inférieure à celle du plus court chemin sans contraintes. Remarquons que les deux quantités nécessaires au calcul de la restriction de capacité sont nécessairement calculées aux deux premières itérations de l'algorithme dual et sont obtenues en fixant λ à une valeur nulle ensuite à une valeur infinie.

Résultats numériques

Nous avons testé l'efficacité de notre algorithme sur 50 problèmes générés aléatoirement. La taille des graphes utilisés variait entre 100 nœuds et 982 arcs et 1 000 nœuds et 28 614 arcs. Nous avons reporté dans les tableaux 5.1 et 5.2 une synthèse des principaux résultats obtenus. Les colonnes des tableaux ont la signification suivante :

N : Nombre total de nœuds (dans le graphe transformé)

A : Nombre total d'arcs

It : Nombre total d'itérations de la méthode dichotomique (*cf.* étape 3)

B Borne supérieure de la solution optimale

b : Borne inférieure de la solution optimale

Z^* : Coût de solution optimale

Ti : Temps de calcul nécessaire pour prouver l'optimalité de la i-ième solution réalisable, ce temps est donné en secondes et a été obtenu sur un micro-ordinateur équipé du microprocesseur 80286 d'Intel.

ki : nombre de chemins générés afin de prouver l'optimalité de la i-ième solution réalisable.

La première constatation que nous pouvons faire est que l'algorithme s'exécute rapidement. En effet, la modestie de l'équipement informatique utilisé n'a pas empêché la résolution de problèmes de très grande taille (1 000 nœuds et environ 30 000 arcs) en moins de 120 secondes pour trouver la solution de coût minimal, et moins de 450 secondes pour prouver l'optimalité de la troisième solution réalisable. Nous remarquons que la résolution du problème dual n'a nécessité qu'un très faible nombre d'itérations (4 itérations dans 26 % des cas, 3 itérations dans 72 % des cas, et 2 itérations dans 2 % des cas), ce qui indique que la méthode dichotomique permet de converger très rapidement vers la solution duale optimale. Nous remarquons également que la borne supérieure s'est révélée être dans 90 % des cas égale à la solution duale. En pratique, chaque fois qu'une nouvelle colonne doit être générée, il serait judicieux de se contenter de la meilleure solution réalisable associée à la solution duale optimale, sauf aux dernières itérations du simplexe quand il deviendra de plus en plus difficile de pouvoir trouver une solution de coût marginal négatif.

TABLEAU 5.1

Résultats de l'algorithme obtenus pour des problèmes ayant entre 100 et 500 nœuds.

N	A	It	B	b	Z^*	$T1$ ($k1$)	$T2$ ($k2$)	$T3$ ($k3$)
100	982	3	64	47	64	2,00 (2)	2,40 (3)	2,92 (4)
100	1 467	2	67	35	67	2,39 (2)	4,19 (5)	8,00 (11)
100	1 987	3	55	38	55	4,05 (2)	7,29 (6)	11,60 (11)
100	2 911	4	16	16	16	4,75 (0)	8,31 (3)	10,10 (4)
100	4 035	3	18	15	18	6,58 (1)	9,87 (3)	13,81 (5)
200	1 932	3	142	118	142	5,20 (2)	9,36 (6)	13,74 (10)
200	3 892	3	69	29	69	12,57 (3)	16,76 (5)	30,07 (11)
200	7 998	3	24	23	24	21,53 (2)	25,83 (3)	50,64 (8)
200	12 132	3	74	39	74	39,19 (3)	65,32 (7)	113,82 (13)
200	16 011	3	13	13	13	25,86 (0)	34,48 (1)	62,52 (4)
300	2 743	3	77	71	77	10,42 (3)	12,15 (4)	16,61 (6)
300	4 565	3	35	32	34	14,45 (2)	20,23 (4)	36,59 (9)
300	9 051	3	58	37	58	22,92 (1)	34,39 (3)	70,84 (9)
300	17 873	3	30	24	30	45,27 (1)	56,59 (2)	105,93 (6)
300	27 157	3	13	13	13	51,59 (0)	103,18 (3)	130,85 (4)
400	3 185	3	95	89	95	11,31 (2)	13,57 (3)	21,58 (6)
400	4 908	3	85	82	85	17,43 (2)	27,89 (5)	51,50 (11)
400	7 929	3	48	40	48	28,16 (2)	33,80 (3)	72,53 (9)
400	16 137	4	30	29	30	57,32 (1)	68,79 (2)	107,93 (5)
400	24 133	4	20	18	20	102,87 (2)	120,02 (3)	212,13 (8)
500	2 442	3	117	96	117	7,58 (1)	12,27 (4)	24,99 (9)
500	4 939	3	122	84	114	19,18 (2)	26,85 (4)	32,87 (5)
500	7 543	3	62	61	62	23,43 (1)	29,29 (2)	36,56 (3)
500	12 548	3	50	50	50	29,24 (0)	38,99 (2)	100,40 (7)
500	24 087	4	47	38	47	134,89 (3)	173,43 (5)	217,06 (7)

TABLEAU 5.2

Résultats de l'algorithme obtenus pour des problèmes ayant entre 600 et 1 000 nœuds.

N	A	It	B	b	Z^*	$T1\ (k1)$	$T2\ (k2)$	$T3\ (k3)$
600	3 676	3	107	97	107	12,28 (1)	15,35 (2)	19,57 (3)
600	7 352	3	89	81	89	24,57 (1)	49,14 (5)	94,26 (11)
600	10 836	3	124	71	124	63,38 (4)	90,54 (7)	147,38 (13)
600	14 523	3	52	47	52	60,67 (2)	97,08 (5)	111,94 (6)
600	17 987	4	44	33	44	90,17 (2)	135,26 (5)	151,05 (6)
700	4 795	3	137	125	137	17,04 (1)	25,56 (3)	44,27 (7)
700	12 190	3	98	77	98	54,16 (2)	65,00 (3)	102,76 (6)
700	17 281	3	37	31	37	76,79 (2)	122,86 (5)	194,80 (9)
700	19 481	4	57	51	57	121,19 (3)	173,13 (6)	263,96 (10)
700	24 431	3	52	34	51	217,12 (7)	260,55 (9)	331,03 (11)
800	6 578	3	137	125	137	24,66 (1)	49,33 (5)	58,61 (6)
800	13 146	4	37	34	37	61,62 (1)	86,27 (3)	171,59 (9)
800	19 454	4	35	33	35	91,19 (1)	145,90 (4)	193,68 (6)
800	22 836	3	49	40	49	85,63 (1)	128,45 (3)	250,09 (8)
800	23 907	4	26	26	26	89,05 (0)	179,30 (4)	255,91 (7)
900	8 159	4	109	109	109	32,07 (0)	62,04 (4)	100,83 (8)
900	16 270	4	75	61	74	127,91 (4)	191,86 (8)	232,79 (10)
900	20 451	3	38	37	38	80,39 (1)	160,78 (5)	271,19 (10)
900	22 796	3	37	37	37	67,20 (0)	156,81 (4)	204,24 (6)
900	23 801	4	63	52	61	163,72 (3)	257,28 (7)	313,79 (9)
1 000	10 011	3	76	74	76	41,04 (1)	61,56 (3)	87,18 (5)
1 000	15 404	4	108	85	108	110,52 (3)	126,31 (4)	194,02 (8)
1 000	20 092	3	56	45	56	102,97 (2)	144,16 (4)	290,49 (10)
1 000	28 587	3	39	37	39	117,21 (1)	234,42 (5)	449,20 (11)
1 000	28 614	3	53	53	53	87,99 (0)	117,32 (1)	313,53(7)

6. CONCLUSION

Cet article a été consacré à l'étude du problème du plus court chemin avec dépendance horaire. Nous avons montré l'intérêt de ce problème pour la résolution optimale du problème de tournées avec contraintes de capacité, fenêtres de temps, et coûts et durées des parcours dépendant du temps. Les résultats numériques obtenus sont encourageants, puisque des problèmes relativement grands ont pu être résolus en des temps assez courts. L'algorithme de génération de colonnes peut aisément être généralisé et traiter des problèmes avec dépendance horaire incluant en particulier les cas multi-dépôts et avec flotte de véhicules hétérogène.

BIBLIOGRAPHIE

1. K. L. Cook et E. Halsey, The Shortest Route Through a Network with Time Dependent Internodal Transit Times, *Journal of Mathematical Analysis and Applications*, *vol. 14*, 1966, p. 493-498.
2. A. De Palma, P. Hansen et M. Labbé, Commuter's Paths with Penalties for Early or Late Arrival Time, *CORE Discussion Paper* n° 8712, 1987.
3. J. P. Desrosiers et F. Pelletier, F. Soumis, Plus court chemin avec contraintes d'Horaires, *RAIRO, Recherche Opérationnelle, vol. 17*, 1983., p. 1-21.
4. S. E. Dreyfus, An Appraisal of Some Shortest Path Algorithms, *Operations Research, vol. 17*, 1968, p. 395-412.
5. M. Gondran et M. Minoux, *Graphes et Algorithmes*, Eyrolles, Paris, 1986.
6. R. W. Hall, The Fastest Path Through a Network with Random Time-Dependent Travel Times, *Transportation Science*, *vol. 20*, 1986, p. 182-188.
7. J. Halpern et I. Priess, Shortest Path with Time Constraints on Mouvement and Parking, *Network*, *vol. 4*, 1974, p. 241-253.
8. G. Y. Handler et I. Zang, A Dual Algorithm for the Constrained Shortest Path Problem, *Networks*, *vol. 10*, 1980, p. 293-310.
9. H. C. Joksch, The Shortest Route with Constraints, *Journal of Mathematical. Analysis and Applications*, *vol. 14*, 1966, p. 191-197.
10. C. Malandraki, *Time Dependent Vehicle Routing Problems: Formulations, Solution Algorithms and Computational Experiments*, Thèse de Ph. D., Northwestern University, Evanston, Illinois, 1989.
11. M. Minoux, Plus court chemin avec contraintes : algorithmes et applications, *Annales des Télécommunications*, *vol. 30*, 1975, p. 383-394.
12. M. Minoux, Structures algébriques généralisées des problèmes de cheminement dans les graphes : théorèmes, algorithmes, et applications, *RAIRO, Recherche Opérationnelle*, *vol. 10*, 1976, p. 33-62.
13. M. Minoux, Résolution des problèmes de grandes dimensions : programmation linéaire généralisée et techniques de décomposition, in *Programmation Mathématique*, Tome 2, Dunod, Paris, 1983, p. 55-105.
14. C. Ribeiro, M. Minoux et M. Penna, An Optimal Column Generation with Ranking Algorithm for Very Large Scale Set Partitionning Problems in Traffic Assignement, *European Journal of Operation Research, vol. 41*, 1989, p. 232-239.