Les Cahiers du GERAD ISSN : 0711–2440

Notes sur l'agrégation des contraintes de ressources en chaque nœud dans un problème de plus court chemin

Y. Miladi

G–2005–04

Janvier 2005

Notes sur l'agrégation des contraintes de ressources en chaque nœud dans un problème de plus court chemin

Yassir Miladi

GERAD et
Département de mathématiques et de génie industriel
École Polytechnique de Montréal
C.P. 6079, Succ. Centre-ville
Montréal (Québec) Canada H3C 3A7
yassir.miladi@gerad.ca

Janvier 2005

Les Cahiers du GERAD

G–2005–04

Résumé

Le problème de plus court chemin avec contraintes de ressources consiste à trouver un chemin d'un point origine à un point destination de coût minimum en respectant les contraintes sur les consommations de ressources. La complexité d'un algorithme de type programmation dynamique augmente avec le nombre de ressources. Une première heuristique couramment utilisée pour produire rapidement des solutions réalisables consiste à dominer uniquement sur un sous-ensemble de ressources. Nagih et Soumis (2005) se proposent l'amélioration de cet heuristique par agrégation des ressources en projetant, en chaque nœud, les ressources sur un vecteur de dimension inférieure en utilisant un algorithme de relaxation lagrangienne pour déterminer les coefficients de la projection.

Ce papier critique la méthode décrite par Nagih et Soumis en prouvant que la méthode de sélection des multiplicateurs de la projection par nœuds ne fournit pas des résultats aussi bons que l'heuristique qu'elle se propose d'améliorer et qu'elle est sensible à des paramètres aléatoires. De plus nous expliquons pourquoi les résultats obtenus sont très instables.

Abstract

The shortest path problem with resource constraints consists in finding a path from an origin point to a destination point at minimum cost while respecting constraints on the use of resources. The complexity of a dynamic programming type of algorithm increases with the number of resources. A first heuristic currently used to quickly produce feasible solutions consists in using dominance only on a subset of resources. Nagih and Soumis (2005) propose an improvement of this heuristic through aggregation of resources at each node by projecting on a vector of smaller dimension and use of a Lagrangian relaxation algorithm to determine the coefficients of the projection.

This paper evaluates the method described by Nagih and Soumis by proving that the selection method of the multipliers of the projection at the nodes does not give results as good as the heuristic that it proposes to improve and that it is sensitive to random parameters. Moreover, we explain why the results obtained are very unstable.

1 Problème de plus court chemin avec contraintes de ressources

Soit $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ un graphe orienté acyclique où $\mathcal{A}$ est l'ensemble des arcs (i, j) et $\mathcal{V} = \mathcal{N} \cup \{o, d\}$ l'ensemble des nœuds topologiquement ordonnés, o étant le nœud origine et d le nœud destination. Un arc $(i, j) \in \mathcal{A}$, consomme une quantité t^r_{ij} de la ressources r et a un coût c_{ij}. À chaque nœud $i \in \mathcal{V}$, on associe des intervalles de contraintes, appelé fenêtres de ressources $[a^r_i, b^r_i]$, $r \in \mathcal{R}$, restreignant la ressource consommée de l'origine o jusqu'au nœud i.

On associe à un chemin P_i, de l'origine o au nœud i, un coût C_i somme des coûts des arcs le composant, et un vecteur d'état $\mathbf{T_i}$, de dimension $|\mathcal{R}|$, telle que chacune de ses composantes T^r_i, $r \in \mathcal{R}$ correspond à la consommation de la ressource cumulée. Le vecteur (C_i, T_i) est appelé étiquette. Un état $\mathbf{T_i}$ est dit réalisable ou légal si :

$$a^r_i \leq T^r_i \leq b^r_i, \ \forall\, r \in \mathcal{R}.$$

Le problème de plus court chemin avec contraintes de ressources PCC-CR consiste à trouver un chemin réalisable de coût minimum allant de l'origine o à la destination d. Le PCC-CR a été introduit par Desrochers (1986) comme une généralisation du problème de plus court chemin avec fenêtres de temps. Il apparaît comme un sous-problème lors de la fabrication d'horaires de travail ou de routes de véhicules à l'intérieur d'un algorithme de génération de colonnes.

Plusieurs méthodes de résolution et applications du PCC-CR ou ses variantes ont été décrites dans la littérature : Minoux (1975), Handler et Zang (1980), Aneja et al. (1983), Desrosiers et al. (1983), Jaffe (1984), Desrochers (1986), Desrochers et Soumis (1988a,b), Desrosiers et al. (1995).

Le problème de plus court chemin avec contraintes de ressources (PCC-CR) se formule comme suit :

$$\min \quad \sum_{i,j} c_{ij} x_{ij} \tag{1.1}$$

$$s.c. \quad \sum_i x_{ij} - \sum_i x_{ji} = \begin{cases} -1 & \text{si } j = o \\ \ \ 0 & \text{si } j \in \mathcal{N} \\ \ \ 1 & \text{si } j = d \end{cases} \tag{1.2}$$

$$x_{ij} \geq 0 \qquad \forall\, (i, j) \in \mathcal{A} \tag{1.3}$$

$$x_{ij}\left(T^r_i + t^r_{ij} - T^r_j\right) \leq 0 \qquad \forall\, (i, j) \in \mathcal{A},\ \forall\, r \in \mathcal{R} \tag{1.4}$$

$$T^r_j \in [a^r_j, b^r_j] \qquad \forall\, j \in \mathcal{V},\ \forall\, r \in \mathcal{R} \tag{1.5}$$

Pour résoudre ce problème, Desrochers et Soumis (1988a) proposent un algorithme de programmation dynamique du type pulling.

1.1 Algorithme de programmation dynamique

Définition 1 *Soient* $(C^{(1)}, T^{(1)})$ *et* $(C^{(2)}, T^{(2)})$ *deux étiquettes associées à deux chemins réalisables* $P^{(1)}$ *et* $P^{(2)}$ *de* o *à* i*. On dit que* $(C^{(1)}, T^{(1)})$ *domine* $(C^{(2)}, T^{(2)})$ *(ou aussi que* $P^{(1)}$ *domine* $P^{(2)}$*) et on note* $(C^{(1)}, T^{(1)}) \preceq (C^{(2)}, T^{(2)})$ *(ou aussi* $P^{(1)} \preceq P^{(2)}$*) si et seulement si*

$$C^{(1)} \leq C^{(2)} \qquad et \qquad T^{r(1)} \leq T^{r(2)}, \qquad \forall r \in \mathcal{R}$$

Définition 2 *Une étiquette associée à un chemin réalisable de* o *à* i*, est dite efficace si elle est minimale au sens de la relation d'ordre* $\preceq$*. Un chemin est dit efficace s'il est associé à une étiquette efficace.*

Dans la littérature, l'étude des solutions efficaces et la réduction de l'espace des états ont fait l'objet de recherches surtout dans le cadre de la programmation multicritère et également pour améliorer l'efficacité de certains algorithmes de programmation dynamique (Christofides et al. (1981), White (1982), Henig (1985), Warburton (1987), Kolen et al. (1987), Solomon (1987), Abdul-Razak et Potts (1988), Bianco et Ricciardelli (1997)).

L'algorithme de programmation dynamique (APD) procède en deux grandes étapes. En chaque nœud $j \in \mathcal{N}$, il effectue les opérations suivantes :

1. Prolongation des chemins (génération des étiquettes et test de réalisabilité),
2. Dominance (élimination des étiquettes non efficaces).

Pour un nœud j donné, des étiquettes sont créées en prolongeant celles présentes aux nœuds i, tels que $(i, j) \in \mathcal{A}$. Ainsi, une nouvelle étiquette $(C_j, T_j^{\mathcal{R}})$ est donnée par

$$\begin{array}{rcl} C_j & = & C_i + c_{ij} \\ T_j^r & = & \max\{a_j^r, T_i^r + t_{ij}^r\},\ r \in \mathcal{R} \end{array}$$

En considérant que tous les prédécesseurs du nœud $j \in \mathcal{N}$ sont déjà traités, la dominance au nœud j peut être interprétée comme la détermination des Pareto optimaux du problème multicritère à $|\mathcal{R}| + 1$ fonctions :

$$\left[\begin{array}{ll} \min\limits_{i,(i,j)\in\mathcal{A}} & \Big(C_i + c_{ij} \ ; \ \max\Big\{ a_j^r, \Big(T_i^r + t_{ij}^r \Big) \Big\},\ r \in \mathcal{R}) \Big) \\ \text{s.}c & T_i^r + t_{ij}^r \leq b_j^r,\ r \in \mathcal{R} \end{array} \right] \tag{1.6}$$

Soit $\mathcal{Z}_{Opt}$ sa valeur optimale.

Dans un récent travail Nagih et Soumis (2005) proposent une méthode d'agrégation des ressources pour les PCC-CR par projection, en chaque nœud en utilisant simultanément un algorithme de programmation dynamique et une relaxation lagrangienne.

1.2 Algorithme de programmation dynamique avec ressources non dominé

L'algorithme de programmation dynamique avec ressources non dominé APD-ND est une heuristique de APD, utilisé afin de réduire l'espace des états et produire des solutions réalisables. Dans cet algorithme la prolongation est effectuée comme pour APD. La dominance se fait sur un sous-ensemble de ressources $\mathcal{R}_1 \in \mathcal{R}$. Ceci est équivalent à la résolution des Pareto optimaux des problème multicritère à $|\mathcal{R}_1| + 1$ fonctions :

$$\left[\begin{array}{ll} \min\limits_{i,(i,j)\in\mathcal{A}} & \Big(C_i + c_{ij} \ \ ; \ \ \max\Big\{a_j^r, \Big(T_i^r + t_{ij}^r\Big)\Big\}\,,\ r \in \mathcal{R}_1)\Big) \\ \text{s.}c & T_i^r + t_{ij}^r \le b_j^r,\ r \in \mathcal{R}_1 \end{array} \right] \tag{1.7}$$

Désignons par $\mathcal{Z}_{ND}$ le coût du meilleur chemin de o à d généré par cet algorithme.

Proposition 1 *On a :*

$$\mathcal{Z}_{OPT} \le \mathcal{Z}_{ND}.$$

Preuve 1 *Si $\mathcal{P}$ est chemin optimal pour APD-ND alors $\mathcal{P}$ est réalisable pour APD.*

Remarque 1 *Dans certains cas on a $\mathcal{Z}_{OPT} < \mathcal{Z}_{ND}$ comme le montre l'exemple de la figure 1.2. En effet, si on applique APD-ND c'est l'étiquette $E_1 = (6,3)$ (de coût 3) qui est générée, alors que l'étiquette optimale est $E_2 = (5.5, 2.5)$ (de coût 5.5).*

2 La relaxation lagrangienne

Selon Nagih et Soumis (2005), le problème de plus court chemin avec contraintes de ressources (PCC-CR) peut se réécrire sous la forme équivalente suivante :

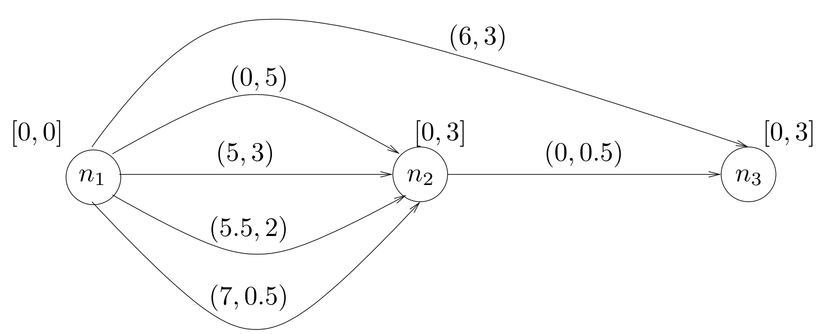

Figure 1

$$
\begin{array}{llrcll}
\min & & \sum_{i,j} c_{ij}x_{ij} & & & (2.1)\\
s.c. & & \sum_{i} x_{ij} - \sum_{i} x_{ji} & = & e_j \quad \forall j \in \mathcal{V} & (2.2)\\
& & x_{ij} & \geq & 0 \quad \forall\,(i,j) \in \mathcal{A} & (2.3)\\
& & x_{ij}\left(T_i^r + t_{ij}^r - T_j^r\right) & \leq & 0 \quad \forall\,(i,j) \in \mathcal{A},\ \forall r \in \mathcal{R} & (2.4)\\
& & T_j^r & \geq & a_j^r \quad \forall\, j \in \mathcal{V},\ \forall r \in \mathcal{R} & (2.5)\\
& & \sum_{i} x_{ij}\left(\max\{a_j^r, T_i^r + t_{ij}^r\} - b_j^r\right) & \leq & 0 \quad \forall\, j \in \mathcal{N} \cup \{d\},\ \forall r \in \mathcal{R} & (2.6)
\end{array}
$$

En dualisant un sous-ensemble $\mathcal{R}_1 \subset \mathcal{R}$ des contraintes (2.6), on obtient la fonction de Lagrange suivante :

$$
\mathcal{Z}_L(u) \quad = \min_{(2.2)-(2.6 \text{ avec } \mathcal{R}\backslash\mathcal{R}_1)} \quad \sum_{i,j} \left(c_{ij} + \sum_{r\in\mathcal{R}_1} u_j^r(\max\{a_j, T_i^r + t_{ij}^r\} - b_j^r) \right) x_{ij}
$$

où $u_j^r \geq 0, j \in \mathcal{N} \cup \{d\},\ r \in \mathcal{R}_1$, sont les multiplicateurs de Lagrange associés. Le dual lagrangien est :

$$
\begin{cases}
\max_u & \mathcal{Z}_L(u) \\
 & \\
s.c. & u_j^{r_1} \geq 0, \quad \forall\, j \in \mathcal{N} \cup \{d\},\ \forall\, r_1 \in \mathcal{R}_1
\end{cases}
$$

En utilisant la formulation (2.1)-(2.6) et en procédant de la même façon qu'à la section 1.1, lors de la résolution par un algorithme de programmation dynamique, la prolongation se fait comme en 1.1 la dominance en chaque nœud j correspond alors à la détermination des Pareto optimaux du problème multicritère à $|\mathcal{R}_2| + 1$ fonctions, où $\mathcal{R}_2 = \mathcal{R} \setminus \mathcal{R}_1$:

$$
\left[
\begin{array}{ll}
\min_{i;(i,j)\in\mathcal{A}} & \left(C_i + c_{ij} + \sum_{r_1\in\mathcal{R}_1} u_j^{r_1}(\max\{a_j^{r_1}, T_i^{r_1} + t_{ij}^{r_1}\} - b_j^{r_1}); \max\left\{a_j^{r_2}, T_i^{r_2} + t_{ij}^{r_2}\right\} \right) r_2 \in \mathcal{R}_2 \\
\text{s.}c & T_i^{r_2} + t_{ij}^{r_2} \leq b_j^{r_2}
\end{array}
\right]
\tag{2.7}
$$

Désignons par APD-L un tel algorithme.

3 APD-L à l'intérieur d'un algorithme de génération de colonnes

Dans un algorithme de génération de colonnes, les PCC-RC apparaissent comme sous-problèmes. À chaque itération de génération de colonnes une colonne réalisable représente un chemin réalisable de coût réduit négatif. Ainsi APD et APD-ND peuvent être utilisés pour résoudre les sous-problèmes dans un problème de génération de colonnes. Par contre APD-L ne peut être utilisé étant donné que les chemins générés par ce dernier peuvent être non réalisables. D'où l'introduction par Nagih et Soumis (2005) de l'algorithme APD-LND, qui n'est autre qu'un APD-ND avec des coûts lagrangiens sur les arcs. Ainsi, pour u fixé, en considérant la résolution par programmation dynamique, la prolongation est effectuée comme présenté en 1.1 et la dominance dans APD-LND est équivalente à la résolution des Pareto optimaux des problèmes multicritères à $|\mathcal{R}_2| + 1$ fonctions :

$$\left[\begin{array}{l} \min\limits_{i;(i,j)\in\mathcal{A}} \left(C_i + c_{ij} + \sum\limits_{r_1\in\mathcal{R}_1} u_j^{r_1}(\max\{a_j^{r_1}, T_i^{r_1} + t_{ij}^{r_1}\} - b_j^{r_1}); \max\left\{a_j^{r_2}, T_i^{r_2} + t_{ij}^{r_2}\right\}\right) r_2 \in \mathcal{R}_2 \\ \text{s.}c \qquad T_i^r + t_{ij}^r \leq b_j^r \end{array}\right] \tag{3.1}$$

Désignons par $\mathcal{Z}_{LND}(u)$ le coût du meilleur chemin généré par cet algorithme.

3.1 Algorithme de Nagih et Soumis

Nagih et Soumis (2005) présentent un algorithme N-S utilisant deux étapes, à chaque itération k de l'algorithme de génération de colonnes, afin de produire des colonnes de coûts marginaux négatifs. La première utilise APD-L afin de trouver des multiplicateurs de Lagrange efficaces u_k, la deuxième consiste à appliquer APD-LND en utilisant les multiplicateurs de Lagrange trouvés dans la première étape afin de générer des colonnes réalisables.

Étant donné, qu'à une itération k de l'algorithme de génération de colonnes, trouver les multiplicateurs optimaux u_k^* nécessiterait plusieurs itérations successives de APD-L, cette méthode peut s'avérer coûteuse. La méthode utilisée par Nagih et Soumis N-S consiste, à appliquer une seule fois APD-L puis à appliquer APD-LND en utilisant les multiplicateurs u_k trouvés afin de produire des colonnes réalisable et de coût marginal négatif. Plus précisément, on choisit d'abord une suite de pas (a_k) telle que le série $(\sum a_k)$ est divergente et $\lim\limits_{k\to\infty} a_k = 0$, autrement dit les conditions qui assurent la convergence de l'algorithme du sous-gradient aussi appelées conditions de Polyak (1967). On applique en premier lieu APD-L en utilisant les multiplicateurs u_{k-1} de l'itération précédente, on trouve les sous-gradients $Sg_k(n_i) = T_{n_i} - b_{n_i}$ correspondants au nœud i, ensuite on calcule les nouveaux multiplicateurs de Lagrange $u_k(n_i) = u_{k-1}(n_i) + a_k \times Sg_k(n_i)$. Cette heuristique est certainement basée sur le fait que lorsque k est grand, le vecteur C_k des coûts réduits sur les arcs du réseau ne change pas beaucoup d'une itération à une autre de l'algorithme de génération de colonnes. Ainsi pour k grand, on peut espérer voir u_k converger vers une valeur optimale.

Désignons par $\mathcal{Z}_{NS}^{Rec}(a_k)$, la valeur optimale donnée par l'algorithme de génération de colonnes pour un problème de recouvrement en utilisant cette méthode pour résoudre les sous-problèmes pour une suite de pas (a_k) donnée, par $\mathcal{Z}_{ND}^{Rec}$ celle trouvée par l'utilisation de l'algorithme APD-ND pour résoudre les sous-problèmes et $\mathcal{Z}_{Opt}^{Rec}$ la valeur optimale de la solution du problème.

Proposition 2 *Il existe un problème de couverture de tâches et une suite de pas* $(a_k)_{k\geq 1}$ *satisfaisant les conditions de Polyak, pour lesquels on a :*

$$\mathcal{Z}_{ND}^{Rec} < \mathcal{Z}_{NS}^{Rec}(a_k)$$

Preuve 2 *Un tel problème est présenté dans l'exemple suivant :*
Considérons le problème de couverture avec des chemins du nœud n_1 *au nœud* n_3 *sur le graphe orienté (figure 1) avec une tâche au nœud destination. Les solutions réalisables de ce problème sont les chemins de* n_1 *à* n_3*. De plus ce problème est équivalent au PCC-CR sous-jacent. On a au total 5 chemins entre* n_1 *et* n_3 *(non nécessairement tous réalisables) d'étiquettes respectives* $E_1 = (6, 1)$*,* $E_2 = (0, 5.5)$*,* $E_3 = (5, 3.5)$*,* $E_4 = (5.5, 2.5)$ *et* $E_5 = (7, 1)$*. Notons que si on applique un algorithme de génération de colonnes sur ce problème en résolvant les PCC-CR avec APD, l'étiquette optimale est* E_4 *de coût 5.5 alors que l'algorithme de génération de colonnes utilisant APD-ND donne l'étiquette* E_1 *de coût 6.*

Appliquons N-S à ce problème avec $a_k = \frac{0.7}{k}$ *qui est une suite vérifiant les conditions de Polyak. Quelques itérations de l'algorithme sont illustrées dans le tableau qui suit. Les étapes sont présentées en ordre, de gauche à droite. Par exemple pour la deuxième itération* $(k = 2)$ *on applique APD-L avec* $u_1(n_2) = 1.4$ *et* $u_1(n_3) = 1$ *trouvés dans la première itération ce qui donne l'étiquette optimale* E_2 *et un sous-gradient au nœud* n_2 *de* $Sg_2(n_2) = 2$*,* $Sg_2(n_3) = 1.5$*,* $u_2(n_2) = u_1(n_3)+a_2\times Sg_2(n_2) = 1.4+\frac{0.7}{2}\times(2) = 2.1$ *et* $u_2(n_3) = u_1(n_3)+a_2 \times Sg_2(n_3) = 1 + \frac{0.7}{2} \times (2) = 1.8$*. Les coûts lagrangiens des étiquettes pour la deuxième itération, qui sont donnés par* $C_{n_3}^L(E_i) = C_{n_3}(E_i) + u_2 \times (T_{n_2} - b_{n_2}) + u_3 \times (T_{n_3} - b_{n_3})$*, sont alors :* $C^L(E_1) = 6$*,* $C^L(E_2) = 8$*,* $C^L(E_3) = 5.9$*,* $C^L(E_4) = 2.5$*,* $C^L(E_5) = -1.85$ *la colonne générée par l'algorithme APD-LND est alors le chemin d'étiquette* E_5*.*

k	$\mathcal{Z}_L(u_{k-1})$	$Sg_k(n2)$	$Sg_k(n_3)$	$u_k(n_2)$	$u_k(n_3)$	$\mathcal{Z}_{LND}$	*Étiq Opt*
0	E_2	2	2.5	0	0	-	-
1	E_2	2	1.5	1.4	1	7	E_5
2	E_2	2	1.5	2.1	1.8	7	E_5
3	E_5	-2.5	-2	1.5	1.6	7	E_5
4	E_5	-2.5	-2	1.4	1.5	7	E_5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

À l'intérieur d'un algorithme de génération de colonnes, lors de la première itération, la solution donnée par N-S est E_5 *de coût 7. La variable duale de la contrainte de couverture devient alors 7. À la deuxième itération, le coût réduit de tous les chemins est ainsi égal à* $C_{n_3}^r(E_i) = C_{n_3}(E_i) - 7$ *donc* $C_{n_3}^r(E_5) = 0$ *ainsi si le critère d'arrêt d'algorithme est le signe des coûts réduits, on voit que pour* $k = 1$ *l'optimalité est atteinte et le chemin optimal pour N-S est donc* E_5*. D'après la figure 3.1 représentant les coûts lagrangiens des 4 chemins*

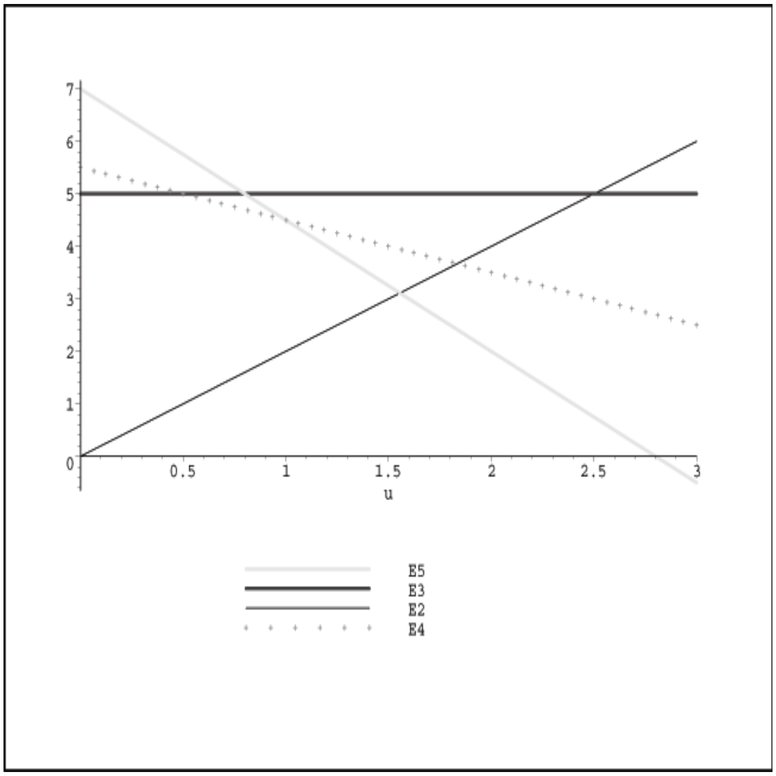

Figure 2

passants par le nœud n_2, on observe que la solution de APD-L sera E_2 ou E_5 et que $u_k(n_2)$ converge vers 1.5 qui représente la valeur du point d'intersection entre les deux droites, par la suite même si on poursuivait les itérations de l'algorithme de génération de colonnes, afin de laisser converger les multiplicateurs, le chemin généré va toujours être E_5. Ainsi quel que soit le critère d'arrêt, avec cette suite de pas, on obtient : $6 = \mathcal{Z}_{ND}^{Rec} < \mathcal{Z}_{NS}^{Rec}(a_k) = 7$.

Proposition 3 *Il existe un problème de couverture de tâches et deux suites de pas semblables, $(a_k)_{k\geq 1}$ et $(b_k)_{k\geq 1}$ satisfaisant les conditions de Polyak, pour lesquels :*

$$\mathcal{Z}_{Opt}^{Rec} = \mathcal{Z}_{NS}^{Rec}(b_k) < \mathcal{Z}_{ND}^{Rec} < \mathcal{Z}_{NS}^{Rec}(a_k)$$

Preuve 3 *Les inégalités : $\mathcal{Z}_{ND}^{Rec} < \mathcal{Z}_{NS}^{Rec}(a_k)$ et $\mathcal{Z}_{Opt}^{Rec} < \mathcal{Z}_{ND}^{Rec}$ ont été déjà prouvées. Pour l'inégalité $\mathcal{Z}_{NS}^{Rec}(b_k) < \mathcal{Z}_{NS}^{Rec}(a_k)$, si on considère le même exemple que ci-dessous, d'après la figure 3.1 on voit que pour $b_k = \frac{0.8}{k}$, l'algorithme N-S donne comme solution l'étiquette E_4 optimale. En effet si on considère les 4 premières itérations représentées dans le tableau qui suit :*

k	$\mathcal{Z}_L(u_{k-1})$	$Sg_k(n2)$	$Sg_k(n_3)$	$u_k(n_2)$	$u_k(n_3)$	$\mathcal{Z}_{LND}$	*Étiq Opt*
0	E_2	*2*	*2.5*	*0*	*0*	-	-
1	E_2	*2*	*2.5*	*1.6*	*1.4*	*7*	E_5
2	E_5	*-2.5*	*-2*	*0.6*	*0.5*	*5.5*	E_4
3	E_2	*2*	*1.5*	*1.1*	*1.02*	*7*	E_5
4	E_2	*2*	*1.5*	*1.5*	*1.4*	*7*	E_5
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

À la deuxième itération l'algorithme génère le chemin d'étiquette E_4 qui est la solution optimale de coût 5.5. Or d'après l'exemple précédent, pour $a_k = \frac{0.7}{k}$ l'algorithme génère la solution correspondante à l'étiquette E_5 de coût 7. Ce qui prouve la proposition.

Remarque 2

1. *Dans les deux exemples précédents, bien que les deux suites $a_k = \frac{0.7}{k}$ et $b_k = \frac{0.8}{k}$ soient assez semblables, les solutions sont différentes (différence > 1). De plus les différences entre les valeurs de $\mathcal{Z}_{NS}^{Rec}$, $\mathcal{Z}_{ND}^{Rec}$ et $\mathcal{Z}_{Opt}^{Rec}$ peuvent être aussi grandes que l'on veut. En effet, il suffit de multiplier les coûts sur les arcs dans l'exemple précédent pas M pour obtenir des différences M fois plus grandes.*
2. *Si à chaque itération de génération de colonnes, on faisait plusieurs itérations de APD-L afin d'obtenir le multiplicateur optimal $u^*(n_2) = 1.5$, on génèrerait toujours le chemin d'étiquette E_5. Ceci montre que N-S peut générer de meilleures solutions avec une suite de pas qui produit des multiplicateurs variés plutôt qu'en utilisant des multiplicateurs lagrangiens optimaux.*
3. *Pour cet exemple, il existe une valeur des multiplicateurs u telle que APD-LND génère la colonne optimale, toutefois ce n'est pas le cas pour tous les problèmes.*

3.2 Commentaires sur les résultats numériques de l'article de Nagih et Soumis (2005)

Comme les meilleures colonnes ne sont pas générées quand u_k s'approche de la valeur optimale u_k^* mais plutôt quand les multiplicateurs tombent sur des "bonnes" valeurs au cours des oscillations, le comportement de l'algorithme peut être assez aléatoire. Ceci explique la grande variabilité des améliorations obtenues avec APD-LND par rapport à APD-ND. Des tests qu'on a effectués sur le problème (Pb 3) de Nagih et Soumis montre que pour des pas "mal" choisis, on peut même avoir des valeurs où $\mathcal{Z}_{ND}^{Rec} < \mathcal{Z}_{NS}^{Rec}$ pour un problème réel (rotation d'équipages d'avions).

Dom	RL	ItrGC	Col	T(spp)	Obj
1	0	190	1992	147	71942.7
1	3	192	1461	238.1	72965.0

Pour les problèmes de grande taille, résolus par génération de colonnes, où il faut des centaines d'itérations avant que les variables duales s'approchent de leurs valeurs optimales, si la suite de pas (a_k) décroit rapidement vers 0, on se retrouve avec des multiplicateurs de Lagrange qui oscillent dans une région proche des multiplicateurs u^* optimaux. Si APD-LND ne produit pas de bonnes colonnes avec les multiplicateurs u^* optimaux la qualité des solutions est très faible. Il est préférable d'opter pour une suite de pas qui tend vers 0 assez lentement pour augmenter la probabilité d'obtenir de meilleures colonnes lors des dernières itérations de génération de colonnes.

Références

[1] T.S. Abdul-Razak et C.N. Potts (1988), Dynamic Programming State-Space Relaxation for Single-Machine Scheduling, *Journal of Operational Research Society*, 39(2), 141–152.

[2] Y.P. Aneja, V. Aggarwal et K.P.K. Nair (1983), Shortest Chain Subject to Side Constraints, *Networks*, 13, 295–302.

[3] L. Bianco et S. Ricciardelli (1997), Dynamic Programming Strategies for the Traveling Salesman Problem with Time Window and Precedence Constraints, *Operations Research*, 45(3), 365–377.

[4] N. Christofides, A. Mingozzi et P. Toth (1981), Exact Algorithms for the Vehicle Routing Problem Based on Spanning Tree and Shortest Path Relaxations. *Mathematical Programming*, 20, 255–282.

[5] M. Desrochers (1986), *La fabrication d'horaires de travail pour les conducteurs d'autobus par une méthode de génération de colonnes*, thèse de doctorat, Université de Montréal, Montréal, Canada.

[6] M. Desrochers et F. Soumis (1988a), A Generalized Permanent Labelling Algorithm for the Shortest Path Problem with Time Windows, *INFOR* 26, 191–212.

[7] M. Desrochers et F. Soumis (1988b), A Reoptimization Algorithm for the Shortest Path Problem with Time Windows, *European Journal of Operational Research*, 35, 242–254.

[8] J. Desrosiers, P. Pelletier et F. Soumis (1983), Plus court chemin avec contraintes d'horaires, *RAIRO Recherche Opérationnelle*, 17(4), 357–377.

[9] G.Y. Handler et I. Zang (1980), A Dual Algorithm for the Constrained Shortest Path Problem, *Networks*, 10, 293–310.

[10] M.I. Henig (1985), The Shortest Path Problem with Two Objective Functions, *European Journal of Operational Research*, 25, 281–291.

[11] J.M. Jaffe (1984), Algorithms for Finding Paths with Multiple Constraints, *Networks*, 14, 95–116.

[12] A.W. Kolen, A.H.G. Rinnooy Kan et H.W.J.M. Trienekens (1987), Vehicle Routing with Time Windows, *Operations Research*, 35(2), 266–273.

[13] M. Minoux (1975), Plus court chemin avec contraintes : algorithmes et applications, *Annales des télécommunications*, tome 30, 326–329.

[14] A. Nagih et F. Soumis (1999), L'agrégation des contraintes de ressources dans un problème de plus court chemin avec contraintes de ressources, *Les Cahiers du GERAD*, G-99-02, février 1999.

[15] A. Nagih et F. Soumis (2005), Nodal Aggregation of Resource Constraints in a Shortest Path Problem, *European Journal of Operational Research*.

[16] A. Warburton (1987), Approximation of Pareto Optima in Multiple-Objective Shortest-Path Problems, *Operations Research*, 35(1), 70–79.

[17] B.T. Polyak (1967), A General Method of Solving Extremum Problems, *Soviet Math. Doklady*, 8(3), 593–597.

[18] D.J. White (1982), The Set of Efficient Solutions for Multiple Objective Shortest Path Problems, *Comput. and Operations Research*, 9(2), 101–107.