

## RESOLUTION PAR DECOMPOSITION D'UN PROBLEME DE TRANSPORT SPECIAL

*Reçu le 21/07/1999– Accepté le 29/11/2003*

### Résumé

Dans cet article, on étudie un problème de transport spécial qu'on appelle problème de transport à destinations groupées. On montre qu'il est NP-dur. On propose ensuite une décomposition lagrangienne d'une partie des contraintes qui permettra de le réduire à un problème équivalent de flot entier à arcs homologues. On présente alors une méthode par séparation et évaluation pour résoudre ce dernier. La procédure d'évaluation est fondée sur la résolution du dual lagrangien par une méthode de sous-gradients. La fonction lagrangienne considérée est obtenue en relaxant les contraintes d'arcs homologues. Des résultats numériques obtenus sur plusieurs problèmes engendrés aléatoirement sont présentés.

**Mots clés:** *Décomposition lagrangienne, flot entier, séparation et évaluation.*

### Abstract

A special transportation problem is addressed. We show it is NP-hard. Using Lagrangian decomposition of part of the constraints, we transform it into an integer minimum cost flow with homologous arcs problem that we solve by a Branch-and-Bound technique. A Lagrangian dual is defined by relaxing the coupling constraints. A standard subgradient algorithm is employed to solve the latter, the value of which provides a lower bound on the value of the objective function value of the original problem. A least-lower-bound-next dichotomous branching strategy is utilized for generating the decision tree. Our algorithm is tested on a set of twenty-five randomly generated instances.

**Keywords:** *Lagrangian decomposition, integer flow, Branch-and-Bound.*

**S. HADDADI**

**A. BENCHETTAH**

Laboratoire de Mathématiques

Appliquées

Université Badji Mokhtar

BP 12, 23000 Annaba, Algérie

Dans cet article, on étudie un problème de transport spécial (PTDG en abrégé). Dans un problème de transport spécial (PTDG en abrégé). Dans un problème de transport classique, chaque destination exprime une demande de ressource. Dans notre modèle, certaines destinations sont groupées, à cause par exemple de leur proximité, et expriment une demande commune. Le modèle de (PTDG) peut également survenir dans des problèmes d'allocation de ressources et d'emploi du temps [7].

Désignons par  $I = \{i_1, i_2, \dots, i_m\}$  un ensemble de  $m$  dépôts où une ressource est entreposée, et pour chaque  $i \in I$ , soit  $a_i$  la disponibilité du dépôt  $i$ . Notons  $J = \{j_1, j_2, \dots, j_n\}$  un ensemble de  $n$  destinations et  $E_k, k \in K$  une famille de sous-ensembles de  $J$  avec  $K = \{k_1, k_2, \dots, k_p\}$ . Chaque ensemble  $E_k$ , appelé superdestination, requiert une quantité de ressource au moins égale à  $b_k$ . Soit  $c_{ij}$  le coût unitaire de transport, et  $x_{ij}$  la quantité de ressource transportée, du dépôt  $i$  à la destination  $j$ . Le modèle mathématique de (PTDG) est

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (1)$$

$$\sum_{j \in J} x_{ij} \leq a_i, \quad i \in I \quad (2)$$

$$\sum_{j \in E_k} \sum_{i \in I} x_{ij} \geq b_k, \quad k \in K \quad (3)$$

$$x_{ij} \in \mathbb{Z}^+, \quad i \in I, j \in J. \quad (4)$$

### 1- COMPLEXITE DE (PTDG)

**Lemme :** Si les coefficients  $c_j$  sont positifs, toute solution optimale du programme linéaire en nombres entiers (R)

$$\min \sum_{j \in J} c_j x_j \quad (5)$$

### ملخص

ندرس مشكلا خاصا للنقل. نبين بأنه NP-صعب فنقترح بالتالي تقسيما لقرنجا لجزء من الشروط. يتم الحصول على الدالة اللاقرنجية عن طريق تخفيف شروط الأفراس المتجانسة. أخيرا تحصلنا على نتائج رقمية من خلال معطيات عشوائية.

**الكلمات المفتاحية:** تقسيم لقرنجا, تدفق, تقسيم وتقييم.

$$\sum_{j \in E_k} x_j \geq 1, \quad k \in K \quad (6)$$

$$x_j \in \mathbb{Z}^+, \quad j \in J \quad (7)$$

a ses composantes binaires.

*Preuve :*

Soit  $x^*$  une solution optimale de (R). Supposons que certaines de ses composantes sont  $\geq 2$ . En posant  $y_j = \min\{1, x_j^*\}$ , il est facile de voir que  $y$  est une solution réalisable de (R). Puisque les coefficients  $c_j$  sont positifs, on a

$$\sum_{j \in J} c_j y_j < \sum_{j \in J} c_j x_j^*. \quad (8)$$

Ce qui contredit l'optimalité de  $x^*$ .  $\square$

**Proposition :** *PTDG est NP-dur.*

*Preuve :*

On montre qu'il est NP-dur même pour des instances particulières. Posons :  $I = \{1\}$ ,  $a_1 = M$ , où  $M$  est une valeur entière arbitrairement grande,  $b_k = 1$ , pour tout  $k$ , et  $c_j > 0$ , pour tout  $j$ . Avec ces données, (PTDG) s'écrit

$$\min \sum_{j \in J} c_j x_j \quad (9)$$

$$\sum_{j \in J} x_j \leq M \quad (10)$$

$$\sum_{j \in E_k} x_j \geq 1, \quad k \in K \quad (11)$$

$$x_j \in \mathbb{Z}^+, \quad j \in J. \quad (12)$$

On voit que la contrainte (10) est valide pour tout  $x$ , et peut par conséquent être omise. D'autre part, l'indice 1 est superflu. Le problème précédent devient alors

$$\min \sum_{j \in J} c_j x_j \quad (13)$$

$$\sum_{j \in E_k} x_j \geq 1, \quad k \in K \quad (14)$$

$$x_j \in \mathbb{Z}^+, \quad j \in J, \quad (15)$$

qui n'est autre, en vertu du lemme, que le problème de recouvrement, connu comme étant NP-dur [2]. Par conséquent, (PTDG) est au moins aussi dur.  $\square$

Cependant, (PTDG) peut être polynomialement résolu dans les cas suivants :

- Les superdestinations sont deux à deux disjointes.

- Chaque superdestination contient deux destinations au plus (par des techniques de couplage [5]).

## 2- REDUCTION DE (PTDG) A UN PROGRAMME LINEAIRE MIXTE

Associons dans les contraintes (3), à chaque quantité

$\sum_{i \in I} x_{ij}$  une variable  $y_j$  et réécrivons (PTDG)

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (16)$$

$$\sum_{j \in J} x_{ij} \leq a_i, \quad i \in I \quad (17)$$

$$\sum_{i \in I} x_{ij} = y_j, \quad j \in J \quad (18)$$

$$\sum_{j \in E_k} y_j \geq b_k, \quad k \in K \quad (19)$$

$$x_{ij} \geq 0 \quad i \in I, j \in J \quad (20)$$

$$y_j \in \mathbb{Z}^+ \quad j \in J. \quad (21)$$

Les contraintes (17) et (18) définissent, pour  $y$  fixé, un problème classique de transport, dont la matrice est totalement unimodulaire. De ce fait, il suffit que les valeurs des variables  $y_j$  soient entières pour que les valeurs des variables  $x_{ij}$  le soient. Pour ces raisons, les contraintes d'intégrité portant sur les variables  $x_{ij}$  ont pu être relâchées. On voit qu'à l'aide de ce simple artifice, on est arrivé à transférer l'intégrité de  $m \times n$  variables à  $n$  variables. Le gain est considérable, même en aboutissant à un programme linéaire mixte.

Une approche classique, qui aurait pu être adoptée, consiste en une décomposition de Benders. Itérativement, un programme-maître formé initialement des contraintes (19) (programme linéaire en nombres entiers) fournit des valeurs  $y_j$  au sous-programme constitué des contraintes (17) et (18) (problème de transport) dont les variables duales serviront, en retour, à construire des coupes pour le programme-maître. Cette approche a l'inconvénient de nécessiter une routine de résolution de programmes linéaires en nombres entiers.

La matrice binaire des contraintes (19), dites de recouvrement généralisé, est la matrice d'incidence hyperarêtes-sommets d'un hypergraphe dont les sommets sont les destinations  $j \in J$  et les hyperarêtes, les superdestinations  $E_k$ ,  $k \in K$ . L'efficacité de tout algorithme de résolution de ce problème dépendra essentiellement de la prise en compte de ces contraintes.

## 3- REDUCTION DE (PTDG) A UN PROBLEME DE FLOT COMPLIQUE PAR DES CONTRAINTES SUPPLEMENTAIRES

Nous pouvons supposer que toute destination appartient à au moins une superdestination et définir le degré d'une destination comme le nombre de superdestinations qui la contiennent :  $d_j = |\{k \in K | E_k \ni j\}|$ ,  $j \in J$ .

Le but de la décomposition lagrangienne est de construire une structure simple des contraintes, en créant  $d_j - 1$  copie(s) de chaque variable  $y_j$ , quitte à ajouter des contraintes couplantes au modèle, exprimant l'égalité de chaque variable et ses copies. Soit  $k_j = \min\{k \in K | E_k \ni j\}$  l'indice de la première superdestination qui contient la destination  $j$ . Posons  $F_k = \{j \in J | k = k_j\}$ ,  $k \in K$  et  $G_k = E_k \setminus F_k$ ,  $k \in K$ .

Ces ensembles, qui expriment la décomposition proprement dite, doivent être interprétés ainsi:  $j \in F_k$  signifie que la variable  $y_j$  est impliquée dans la contrainte numéro  $k$ , tandis que  $j \in G_k$  indique qu'une copie  $z_j^k$  de la variable  $y_j$  est impliquée (à la place de  $y_j$ ) dans la contrainte numéro  $k$ . Puisque chaque variable  $y_j$  est impliquée dans une seule contrainte, on peut lui adjoindre, dans un souci purement didactique, un subscript qui indiquera la contrainte dans laquelle elle est impliquée. Avec ces notations, considérons le problème (P)

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (22)$$

$$\sum_{j \in J} x_{ij} = a_i, \quad i \in I \quad (23)$$

$$\sum_{i \in I} x_{ij} = y_j^k, \quad j \in J \quad (24)$$

$$\sum_{j \in F_k} y_j^k + \sum_{j \in G_k} z_j^k \geq b_k, \quad k \in K \quad (25)$$

$$y_j^k = z_j^k, \quad j \in G_k, k \in K \quad (26)$$

$$x_{ij} \in \mathbb{Z}^+, \quad i \in I, j \in J \quad (27)$$

$$y_j^k \in \mathbb{Z}^+, \quad j \in J \quad (28)$$

$$z_j^k \in \mathbb{Z}^+, \quad j \in G_k, k \in K. \quad (29)$$

Les problèmes (P) et (PTDG) sont équivalents dans la mesure où si  $(x^*, y^*, z^*)$  est une solution optimale de (P) alors  $x^*$  est une solution optimale de (PTDG). De plus, le problème obtenu à partir de (P) par la relaxation des  $r = \sum_{k \in K} |G_k|$  contraintes (26) est un problème de flot ordinaire de coût minimum, et (PTDG) se réduit ainsi à un problème de flot de coût minimum avec la contrainte que les arcs correspondant à la variable  $y_j^k$  et ses copies doivent porter la même valeur du flot. La décomposition lagrangienne proposée se justifie donc par le fait que la relaxation des contraintes couplantes aboutit à un problème de flot ordinaire polynomialement résoluble.

#### Exemple :

Puisque la décomposition suggérée ne concerne que les relations d'inclusion des superdestinations, c'est-à-dire la famille  $E_k$ ,  $k \in K$ , restreignons (PTDG) aux données suivantes:  $n = 4$ ,  $p = 3$ ,  $E_1 = \{1, 2, 4\}$ ,  $E_2 = \{2, 3\}$  et  $E_3 = \{1, 2\}$ . Les contraintes (19) correspondant à ces données sont

$$\begin{cases} y_1 + y_2 + y_4 \geq b_1 \\ y_2 + y_3 \geq b_2 \\ y_1 + y_2 \geq b_3. \end{cases}$$

Les contraintes (25) et (26) s'écrivent alors pour cet exemple

$$\begin{cases} y_1^1 + y_2^1 + y_4^1 \geq b_1 \\ z_2^2 + y_3^2 \geq b_2 \\ z_1^3 + z_2^3 \geq b_3 \\ y_1^1 = z_1^3 \\ y_2^1 = z_2^2 \\ y_2^1 = z_2^3. \end{cases}$$

## 4- RESOLUTION DE (P) PAR SEPARATION ET EVALUATION

Cette méthode utilise une stratégie de branchement dichotomique en largeur. La procédure d'évaluation est fondée sur la relaxation lagrangienne des contraintes (26) et une méthode de sous-gradients pour résoudre le dual lagrangien.

### 4.1- Procédure d'évaluation et sa complexité

Associés un  $r$ -vecteur  $\pi$  de multiplicateurs de Lagrange aux contraintes couplantes (26). Le dual lagrangien de (P) est alors le problème (D)

$$\max_{\pi \in \mathbb{R}^r} W(\pi) \quad (30)$$

avec  $W(\pi)$  valeur optimale du problème (R( $\pi$ ))

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{k \in K} \sum_{j \in G_k} \pi_j^k (y_j^k - z_j^k) \quad (31)$$

sous les contraintes (23) à (25) et (27) à (29).

(R( $\pi$ )) est un problème de flot ordinaire sur le réseau  $(X, U, \alpha, \beta, \gamma)$  avec

$$X = I \cup J \cup K \cup \{s_1, s_2\}$$

$$U = U_1 \cup U_2 \cup U_3 \cup U_4 \cup U_5 \cup \{(s_2, s_1)\}$$

où

$$U_1 = \{(s_1, i), i \in I\},$$

$$U_2 = \{(i, j), i \in I, j \in J\},$$

$$U_3 = \{(j, k), j \in F_k, k \in K\},$$

$$U_4 = \{(s_1, k), j \in G_k, k \in K\},$$

$$U_5 = \{(k, s_2), k \in K\}$$

$$\alpha_u = \begin{cases} b_k & u \in U_5 \\ 0 & \text{sinon} \end{cases}; \quad \beta_u = \begin{cases} a_i & u \in U_1 \\ +\infty & \text{sinon} \end{cases}$$

$$\gamma_u = \begin{cases} c_{ij} & u \in U_2 \\ \pi_j^k & u \in U_4 \\ \sum_{G_k \ni j} \pi_j^k & u \in U_3 \\ 0 & \text{sinon.} \end{cases}$$

Chaque arc de  $U_3$  correspond à une variable  $y_j^k$ . Les variables  $y_j^k$  sont liées aux variables  $x_{ij}$  par les équations (24). Les variables  $z_j^k$  n'ont de lien qu'avec les variables  $y_j^k$  par (25) et (26). Les arcs correspondant aux variables  $z_j^k$  ont donc pu être reliés directement à la source (ensemble  $U_4$ ). Ce réseau comporte  $m+n+p+2$  sommets,  $mn+m+n+p+r+1$  arcs.

Notons que (D) a même valeur que la PL-relaxation de (PTDG) puisque le problème (R( $\pi$ )) a la propriété d'intégralité [3].

#### Exemple (suite) :

Le réseau associé est montré à la figure 1 (avec  $m = 2$ ). On a  $r = 3$  contraintes. Les deux arcs  $(j_1, k_1)$  et  $(s_1, k_3)$ , qui correspondent respectivement à la variable  $y_1^1$  et sa copie  $z_1^3$ , doivent porter la même valeur du flot. Il en va de même pour les trois arcs  $(j_2, k_1)$ ,  $(s_1, k_2)$  et  $(s_1, k_3)$  qui correspondent respectivement à la variable  $y_2^1$  et ses copies  $z_2^2$  et  $z_2^3$ .

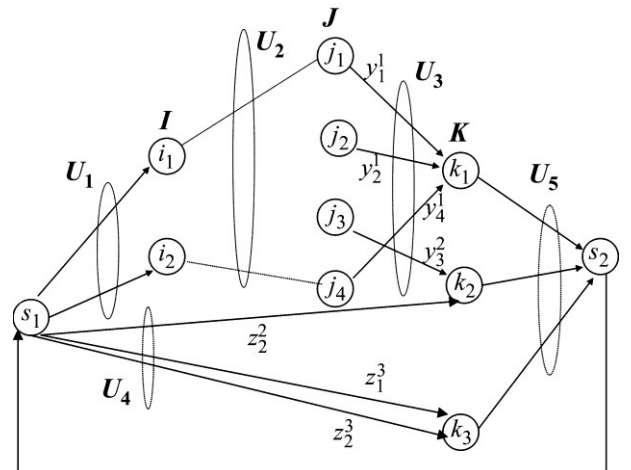


Figure 1: Réseau de l'exemple.

**procédure** évaluation;

**début**

$t \leftarrow 0; \rho \leftarrow 2;$

**répéter**

$t \leftarrow t+1;$

**résoudre**  $R(\pi^t);$

**si** il a une solution **alors**  $\{(x^{(t)}, y^{(t)}, z^{(t)})$  solution optimale de coût minimal  $W(\pi^t)\}$

**début**

$BI \leftarrow \max \{ \lceil W(\pi^{(t-1)}) \rceil, \lceil W(\pi^t) \rceil \};$

$norme2 \leftarrow \sum_{k \in K} \sum_{j \in G_k} (y_j^{k(t)} - z_j^{k(t)})^2;$  {norme au carré du sous-gradient}

**si**  $norme2 > 0$  **alors**

**début**

$\lambda_t \leftarrow \rho(L - W(\pi^t)) / norme2;$

**pour**  $k \in K$  **faire**

**pour**  $j \in G_k$  **faire**  $\pi_j^{k(t+1)} \leftarrow \pi_j^{k(t)} + \lambda_t \sum_{k \in K} \sum_{j \in G_k} (y_j^{k(t)} - z_j^{k(t)});$

**si**  $\lceil W(\pi^{(t-1)}) \rceil > \lceil W(\pi^t) \rceil$  **alors**  $\rho \leftarrow \rho / 2$

**fin**

**fin**

**jusqu'à**  $(R(\pi^t)$  sans solution) **ou**  $(norme2 = 0)$  **ou**  $(|\lambda_t| \leq \varepsilon)$  **ou**  $(t = nmi)$  **ou**  $(BI \geq BS)$

**fin;**

En chaque nœud de l'arborescence de décision, on utilise une méthode de sous-gradients [4], résumée ci-dessus, pour résoudre (D). Cela fournira une borne inférieure (BI). La borne supérieure (BS) est le coût de la meilleure solution réalisable courante.

La valeur  $L$  doit être une surestimation de la solution optimale. Tant qu'on n'a pas de solution réalisable de (PTDG), elle est initialisée, comme en [6], par la valeur de  $W(0)$  calculée à la racine de l'arbre à la première itération de la méthode de sous-gradients ( $W(0)$  est une sous-estimation de l'optimum) multipliée par un facteur approprié, 1.2 par exemple. Dès que des solutions réalisables sont obtenues,  $L$  sera mis à jour par le coût de la meilleure solution réalisable courante. Une règle empirique classique [4] consiste à débiter avec le coefficient de relaxation  $\rho = 2$  puis le diviser par 2 à chaque fois que la méthode de sous-gradients échoue dans l'amélioration de la fonction objective  $W(\pi)$  de (D), c'est-à-dire lorsque d'une itération à la suivante la valeur de  $W(\pi)$  diminue. Plusieurs tests d'arrêt sont indiqués. Si le problème  $(R(\pi))$  n'a pas de solution, il en va de même de (P), et donc de (PTDG), puisque le premier contient le second. Si la norme au carré du sous-gradient est nulle alors la solution de  $(R(\pi))$  est un flot réalisable satisfaisant les contraintes (26) et sa restriction aux arcs de  $U_2$  est une solution réalisable de (PTDG). Cependant, lors de la méthode de sous-gradients, on ne se contente pas d'attendre une solution réalisable qui satisfait les contraintes relaxées. On teste à chaque itération si la restriction à  $U_2$  du flot qui est solution réalisable du problème  $(R(\pi))$  constitue une solution réalisable de PTDG. Si c'est le cas et si son coût est inférieur à celui de la borne supérieure  $BS$ , on met à jour cette borne. Si la borne inférieure n'est pas inférieure à la borne supérieure alors le sous-problème correspondant au nœud évalué ne peut contenir une solution meilleure que la meilleure solution

réalisable courante. Lorsque le pas  $\lambda_t$  est trop petit en valeur absolue, les coefficients de Lagrange ne changent pratiquement plus. Dans tous les cas, le nombre maximum d'itérations ( $nmi$ ) est fixé.

Puisque le nombre d'itérations de la méthode de sous-gradients est fixé et puisque la procédure dominante est celle de résolution de  $(R(\pi))$ , l'évaluation d'un nœud a même complexité qu'un problème de flot de coût minimum.

#### 4.2- Procédures de branchement, sélection et séparation

On ne garde en mémoire que la pile des nœuds non

élagués. On commence avec une borne supérieure infinie. On évalue chaque nœud non encore évalué, on met à jour la borne supérieure et on élague le nœud correspondant à chaque fois qu'on trouve une solution réalisable avec un coût strictement plus faible que la borne supérieure courante. De même, on élague chaque nœud sans solution ou dont la borne inférieure est plus grande ou égale à la borne supérieure courante. Lorsque tous les nœuds auront été évalués sans vider la pile, on se branche sur celui ayant la plus faible évaluation.

Rappelons que les contraintes relaxées sont celles qui expriment l'égalité de la valeur du flot sur les arcs homologues. On cherche, à l'aide de la procédure de séparation, à essayer de forcer cette égalité. On sélectionne le paquet d'arcs homologues identifié par

$$j^* = \operatorname{argmax}_{j \in J} \left\{ \max_{G_k \in j} \{y_j^k, z_j^k\} - \min_{G_k \in j} \{y_j^k, z_j^k\} \right\}. \quad (32)$$

C'est donc celui qui 'viole le plus' les contraintes (26) d'égalité du flot. Soit

$$\eta = \frac{1}{2} \left( \max_{G_k \in j^*} \{y_j^{k^*}, z_j^{k^*}\} + \min_{G_k \in j^*} \{y_j^{k^*}, z_j^{k^*}\} \right) \quad (33)$$

le 'flot moyen' sur les arcs du même paquet identifié par  $j^*$ . On sépare ce nœud sur lequel on s'est branché en deux nœuds descendants et on pose dans le premier  $b_u = \lfloor \eta \rfloor$  pour tout arc  $u$  du paquet  $j^*$ , et dans le second  $a_u = \lceil \eta \rceil$  pour tout arc  $u$  du même paquet. L'algorithme finit lorsque la pile est vide. Si la borne supérieure courante est finie, c'est alors le coût optimal de (PTDG), sinon ( $BS = +\infty$ ) le problème n'a pas de solution.

#### 5- EXPERIMENTATION NUMERIQUE

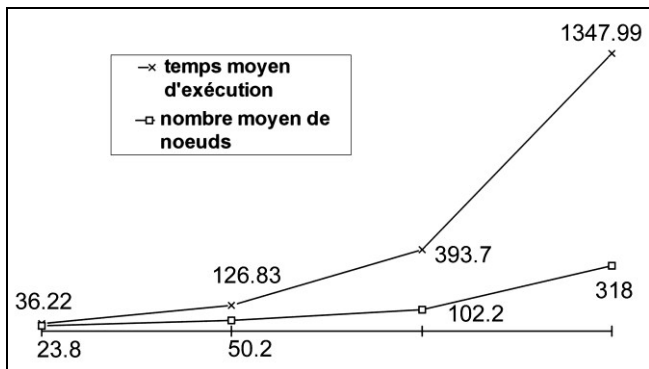
Notre algorithme est codé en *Turbo Pascal* et testé sur un ordinateur *Alfatron Pentium II* à 350 MHz. Le temps

consommé, en secondes, est évalué à l'aide de la fonction DOS *gettime*. La valeur de  $\epsilon$  en-dessous de laquelle on arrête la procédure de sous-gradients est fixée à  $10^{-3}$  et le nombre maximum d'itérations est fixé à 50. L'algorithme *out-of-kilter* [1] est utilisé pour résoudre le problème  $(R(\pi))$ .

L'analyse théorique de l'algorithme présenté est très difficile, sinon impossible, à mener. On voit bien qu'elle dépend d'un grand nombre de paramètres parmi lesquels on peut citer : la dimension des instances  $(m, n, p)$ , la densité des super-destinations  $E_k$ , les coefficients  $a_i, c_{ij}$  et  $b_k$ . On se contentera donc de présenter une analyse empirique sur 25 instances. Les données ont été engendrées aléatoirement en fixant le nombre de dépôts à 10 et le nombre de destinations à 50. Les coûts de transport sont uniformément distribués dans l'intervalle  $[1,100]$ , les disponibilités de ressources dans  $[5,25]$ . Afin d'éviter l'inconsistance, les bornes  $b_k$  sont calculées de la manière suivante

$$b_k = \left\lfloor \frac{|E_k|}{m} \sum_{i \in I} a_i \right\rfloor \quad (34)$$

La matrice binaire des contraintes (19) est engendrée aléatoirement avec, cependant, la règle que chaque destination appartient à au moins une superdestination, et que chaque superdestination contient au moins trois destinations. On calcule, pour chaque problème, le coût optimal, la taille de l'arbre de décision et le temps consommé en secondes (incluant les opérations d'entrée/sortie.) Les résultats obtenus sont consignés au tableau 1, dont les valeurs moyennes sont représentées sur un graphique (Fig.2.)



**Figure 2:** Variation de la taille de l'arbre de décision et du temps moyen d'exécution en fonction du nombre de superdestinations (15,20,25,30).

Nombre de Superdestinations	Taille du réseau		Nombre de Contraintes du type (9)	Séparation et évaluation	
	Nombre de Sommets	Nombre d'arcs		Nombre de noeuds	Temps d'exécution (secondes)
10	72	573	2	5	3.79
		572	1	1	0.38
		573	2	1	1.26
		573	2	5	2.47
		574	3	3	2.36
<b>Moyennes</b>				<b>3.0</b>	<b>2.05</b>
15	77	583	7	11	8.95
		584	8	13	22.41
		583	7	51	94.14
		580	4	21	27.36
		583	7	23	28.24
<b>Moyennes</b>				<b>23.8</b>	<b>36.22</b>
20	82	595	14	13	14.45
		600	19	115	315.33
		602	21	57	133.08
		596	15	33	62.28
		601	20	33	109.02
<b>Moyennes</b>				<b>50.2</b>	<b>126.83</b>
25	87	613	27	107	338.23
		615	29	71	293.80
		611	25	75	173.07
		616	30	147	649.50
		615	29	111	513.89
<b>Moyennes</b>				<b>102.2</b>	<b>393.70</b>
30	92	633	42	335	1418.95
		640	49	377	1266.20
		635	44	277	1376.00
		636	45	159	883.53
		636	45	411	1795.28
<b>Moyennes</b>				<b>311.8</b>	<b>1347.99</b>

**Tableau 1:** Résultats obtenus.

Les choix précédents ne sont pas arbitraires. Ils sont justifiés par deux considérations :

- Le compilateur de Turbo-Pascal n'alloue qu'une quantité limitée de mémoire d'exécution. Il ne nous a donc pas été possible de traiter des réseaux de plus de 100 sommets et 650 arcs (Tab. 1), d'autant plus qu'il fallait ménager de l'espace pour l'arbre de décision engendré dynamiquement.

- On peut voir à partir des résultats obtenus (Tab. 1) que la performance de l'algorithme n'est pas tellement liée à la taille du réseau, puisque toutes les instances traitées induisent des réseaux de dimensions analogues. En revanche, un paramètre crucial est le nombre  $p$  de superdestinations.

En vertu de ce qui précède, on a essayé d'analyser le comportement de l'algorithme en faisant varier  $p$ . Clairement, plus  $p$  augmente plus le nombre de contraintes d'arcs homologues relaxées du type (26) induit augmente, les superdestinations devenant de plus en plus imbriquées. Deux observations s'en suivent : tout d'abord, le nombre de noeuds nécessaires pour trouver la solution optimale, et la

confirmer, augmente considérablement. Le temps consommé semble augmenter dans des proportions plus importantes. Cela provient du fait que le temps de traitement d'un nœud augmente car, si la taille du réseau n'augmente pas significativement, la dimension du vecteur des multiplicateurs de Lagrange, en correspondance avec le nombre de contraintes relaxées, augmente considérablement.

### CONCLUSION

Notre modeste prétention était de présenter un algorithme de résolution pour un problème difficile, dont on a analysé empiriquement la performance sur un échantillon réduit d'instances. Les auteurs sont conscients qu'il manque une étude comparative (avec des méthodes connues) des performances, ne serait-ce que par rapport à ce que donnerait un logiciel spécialisé tel que *Cplex*. Cette perspective est dans la préoccupation des auteurs.

#### Remerciements:

Les auteurs tiennent à remercier l'un des référés anonymes pour ses corrections et suggestions, concernant notamment les sections 1 et 5, qui ont permis d'améliorer substantiellement la première version.

### REFERENCES

- [1]- Ahuja R.K., Magnanti T.L. and Orlin J.B., "Network flows", in: G.L. Nemhauser, A.H.G. Rinnooy Kan et M.J. Todd (eds.): *Handbooks in Operations Research and Management Science*, Vol.1, "Optimization", North-Holland, Amsterdam, (1989), pp.211-369.
- [2]- Garey M.R. and Johnson D.S., "Computers and intractability: a guide to the theory of NP-completeness", W.H. Freeman and Co., New York, (1979).
- [3]- Guignard M., and Kim S., "Lagrangian decomposition: a model yielding stronger lagrangian bounds", *Mathematical Programming*, 39, (1987), pp.215-228.
- [4]- Held M., Wolfe P. and Crowder H.D., "Validation of subgradient optimisation", *Mathematical Programming*, 6, (1974), pp.62-88.
- [5]- Korte B. and Vygen J., "Combinatorial optimization: theory and algorithms", Springer-Verlag, Berlin, (2000).
- [6]- Legendre J.P. et Minoux M., "Une application de la notion de dualité en programmation en nombres entiers: sélection et affectation optimales d'une flotte d'avions", *Rairo*, 11, (1977), pp.201-222.
- [7]- Nicolas P., "Planification et allocation de ressources avec contraintes temporelles", thèse de doctorat de l'université Blaise Pascal, Clermont-Ferrand, France, (1991). □