

# Résolution de problèmes de multiflots linéaires

<http://www.unige.ch/hec/logilab/templeet/template/papiers/papier75LinearMCF.pdf>

---

Journée Francilienne de  
Recherche Opérationnelle  
11 juin 2004

J.-P. Vial (HEC-Genève),  
*co-auteurs*  
F. Babonneau et O. du Merle

# Formulations du problème de multiflot linéaire et relaxation lagrangienne

## Formulation par flot sur les arcs

$$\min \sum_{a \in \mathcal{A}} t_a \sum_{k \in \mathcal{K}} x_a^k \quad (1)$$

$$\sum_{k \in \mathcal{K}} x_a^k \leq c_a, \quad \forall a \in \mathcal{A}, \quad (2)$$

$$Nx^k = d_k \delta^k, \quad \forall k \in \mathcal{K}, \quad (3)$$

$$x_a^k \geq 0, \quad \forall a \in \mathcal{A}, \forall k \in \mathcal{K}.$$

$\mathcal{G}(\mathcal{N}, \mathcal{A})$ , graphe orienté

$N$  matrice (nœud/arc) du réseau ;  $t_a$  coût unitaire de transit sur l'arc  $a$  ( $t$  pour «travel time»);  $d_k$  demande de la commodité  $k$ ;  $\delta^k$  vecteur d'origine et de destination de la commodité  $k$ .  $c_a$  capacité de l'arc  $a$

$|\mathcal{A}| \times |\mathcal{K}|$  variables et  $|\mathcal{A}| + |\mathcal{N}| \times |\mathcal{K}|$  contraintes.

## Formulation par chemins

$$\min \sum_{k \in \mathcal{K}} \sum_{j \in J_k} \xi_j \gamma_j \quad (4)$$

$$\sum_{k \in \mathcal{K}} \sum_{j \in J_k} \xi_j \pi_j \leq c, \quad (5)$$

$$\sum_{j \in J_k} \xi_j = d_k, \quad \forall k \in \mathcal{K}, \quad (6)$$

$$\xi \geq 0.$$

$\pi$ , vecteur booléen sur l'ensemble des arcs, est un chemin d'une origine à destination.  $J_k$  est l'ensemble des chemins allant du nœud origine au nœud destination du bien  $k$ . Le coût de transit d'une unité de flot sur le chemin  $\pi_j$  est  $\gamma_j$  et  $\xi_j$  est le flot total sur ce chemin.

# Relaxation lagrangienne

$$\max_{u \geq 0} \mathcal{L}(u)$$

avec

$$\mathcal{L}(u) = \min_{x^k \geq 0, k \in \mathcal{K}} \left\{ L(x, u) \mid Nx^k = \delta^k, \forall k \in \mathcal{K} \right\}, \quad (7)$$

Le lagrangien  $L(x, u)$  est défini par

$$L(x, u) = \sum_{a \in \mathcal{A}} t_a \sum_{k \in \mathcal{K}} x_a^k + \sum_{a \in \mathcal{A}} u_a \left( \sum_{k \in \mathcal{K}} x_a^k - c_a \right).$$

Le problème (7) est separable en  $|\mathcal{K}|$  problèmes indépendants de plus court chemin.

## Propriétés de la fonction duale $\mathcal{L}$

$\mathcal{L}(u)$  est le minimum de plusieurs formes linéaires en  $u$  : c'est donc une fonction concave.

Soit  $\bar{x}$  une solution optimale du problème (7) qui définit  $\mathcal{L}(u)$  en  $u = \bar{u}$ . Pour tout  $u \neq \bar{u}$

$$\mathcal{L}(u) \leq \sum_{a \in \mathcal{A}} t_a \sum_{k \in \mathcal{K}} \bar{x}_a^k + \sum_{a \in \mathcal{A}} u_a \left( \sum_{k \in \mathcal{K}} \bar{x}_a^k - c_a \right). \quad (8)$$

L'inégalité (8) démontre que  $-c + \sum_{k \in \mathcal{K}} \bar{x}^k$  est un anti-sous-gradient. Elle définit une **approximation linéaire majorante** de  $\mathcal{L}$ .

On parle aussi de **coupe d'optimalité**.

Généralités sur la relaxation lagrangienne et  
la résolution par une méthode de plans  
coupants

## Problème linéaire avec contraintes de capacité

Le problème de multiflot linéaire peut s'écrire de manière compacte

$$\min\{t^T x \mid Bx \leq c, x \in X\}. \quad (9)$$

$x \in X$  représente le représente l'ensemble des contraintes de satisfaction des demandes.  $Bx \leq c$  représente la contrainte que le flot total sur les arcs n'excède pas leur capacité.

En relaxant les contraintes de capacité, on définit le problème lagrangien dual

$$\max_{u \geq 0} \mathcal{L}(u)$$

$$\mathcal{L}(u) = -c^T u + \min_{x \in X} (t + B^T u)^T x.$$

## Anti-sous-gradient de $\mathcal{L}(u)$

Oracle

$$\mathcal{M}(u) = \min_{x \in X} (t + B^T u)^T x. \quad (10)$$

Soit  $\bar{x}$  la solution optimale de l'oracle (10) au point  $\bar{u}$ . Par définition de l'opération de minimisation,

$$\mathcal{M}(u) \leq t^T \bar{x} + (B\bar{x})^T \bar{u}$$

et

$$\mathcal{L}(u) \leq -\bar{a}^T u + \bar{\gamma}$$

avec

$$\bar{a} = c - B^T \bar{x} \quad \text{et} \quad \bar{\gamma} = t^T \bar{x}.$$

## Ensemble de localisation

Soient  $\{u^i\}_{i=1,\dots,m}$  une suite de points d'appel et soient  $\{x^i\}_{i=1,\dots,m}$  les réponses de l'oracle. On dénote

$$\underline{\theta} = \max_{i \leq m} \mathcal{L}(u^i)$$

Dans l'espace de l'hypographe de la fonction duale lagrangienne, les coupes d'optimalité définissent un ensemble de localisation

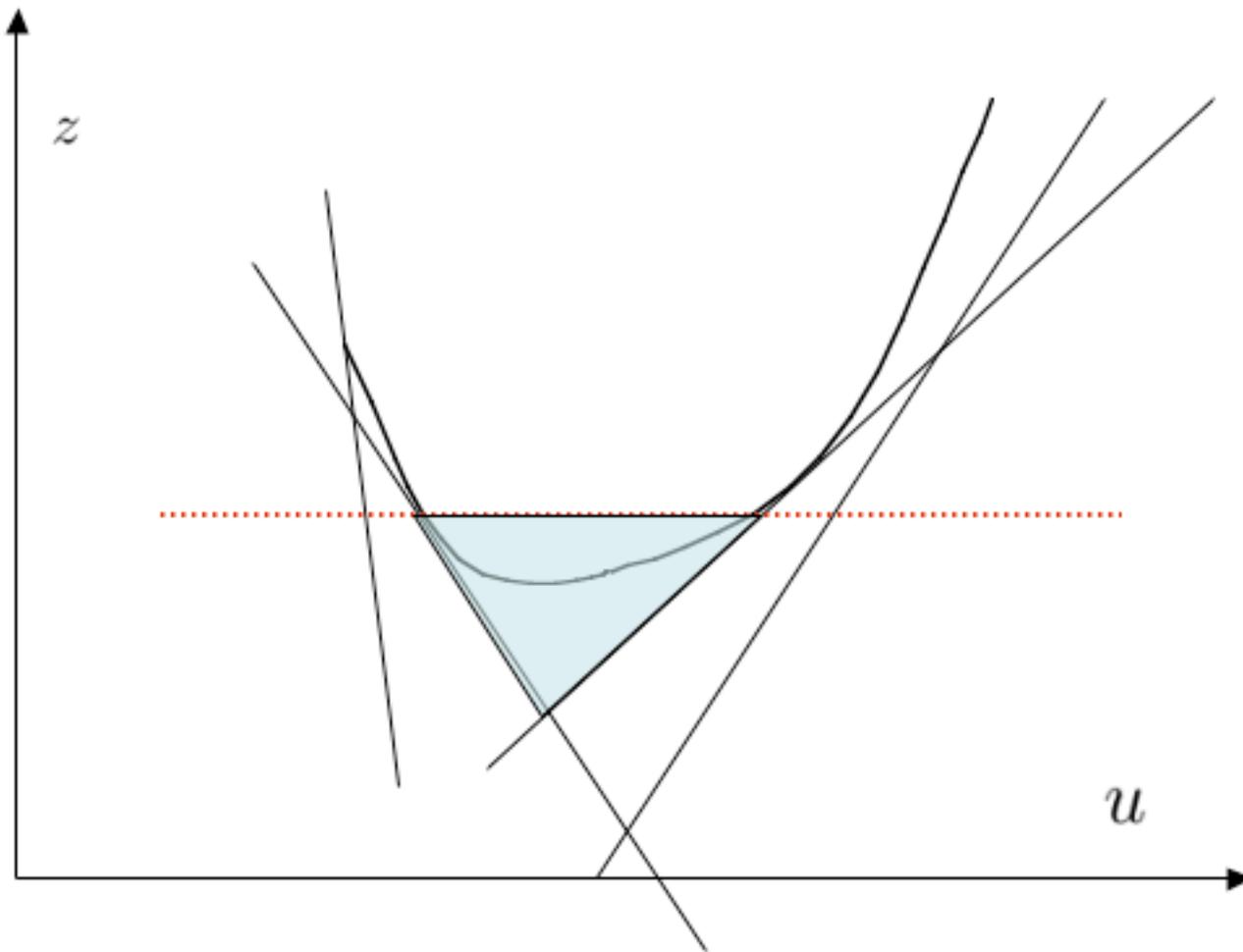
$$\mathcal{F} = \{(u, z) \mid (a^{(i)})^T u + z \leq \gamma_i, i = 1, \dots, m, z \geq \underline{\theta}, u \geq 0\}.$$

L'ensemble de localisation contient l'ensemble des solutions optimales  $(u^*, \mathcal{L}(u^*))$ .

# Attention !

Le transparent qui suit doit se  
regarder la tête en bas !

# Localization set



Les transparents qui suivent se regardent en position normale.

# Algorithme générique de plans coupants

Après  $m$  appels à l'oracle, on a bâti l'ensemble de localisation  $\mathcal{F}_m$ .

## Itération extérieure

1. Choisir  $(u^{(m+1)}, z_{m+1}) \in \mathcal{F}_m$ .  
(Calculer une borne supérieure  $\theta_{m+1}$ .)
2. Appelé en  $u^{(m+1)}$ , l'oracle renvoie  $\mathcal{L}(u^{(m+1)})$  et la coupe

$$(a^{(m+1)})^T u + z \leq \gamma_{m+1}.$$

3. Mettre à jour
  - L'ensemble de localisation  
 $\mathcal{F}_{m+1} = \mathcal{F}_m \cap \{(a^{(m+1)})^T u + z \leq \gamma_{m+1}\}$ .
  - La borne supérieure  $\bar{\theta} = \min\{\bar{\theta}, \theta_{m+1}\}$ .
  - La borne inférieure  $\underline{\theta} = \max\{\underline{\theta}, \mathcal{L}(u^{(m+1)})\}$ .
4. Test d'arrêt : si  $(\bar{\theta} - \underline{\theta}) \leq \epsilon \bar{\theta} \Rightarrow \text{STOP}$

Fonction barrière proximale

$$\Phi_k(u, z) = \frac{\rho}{2} \|u - \bar{u}\|^2 - \sum_{i=1}^k \log(\gamma_i - (a^{(i)})^T u - z) - \log(z - \underline{\theta}) - \sum_j \log u_j.$$

Le centre proximal  $\bar{u} = u^j$ , avec  $j = \arg \max_{i \leq k} \{L(u^i)\}$ , correspond à la meilleure réponse de l'oracle. Le centre analytique proximal est unique et existe toujours.

## Calcul du centre analytique proximal

Méthode de Newton

$$(u, z)^{\dagger} = (u, z) + \tau d, \quad \tau > 0$$

$$d = -[\Phi_k''(u, z)]^{-1} \Phi_k'(u, z).$$

La nouvelle coupe exclut le point courant :  $\gamma_{k+1} < (a^{(k+1)})^T u + z$ .  
La fonction barrière  $\Phi_{k+1}(u, z)$  n'est pas définie en ce point. Il faut donc avoir recours à une variante avec départ non réalisable.

## Rappel : la méthode de K-G-C

Le point d'appel est déterminé par

$$\max\{z \mid (a^i)^T u + z \leq \gamma_i, i = 1, \dots, m, u \geq 0\}. \quad (11)$$

avec  $a^i = Bx^i$  et  $\gamma_i = t^T x^i$ .

Remarque : le problème (11) peut être non-borné. Il est nécessaire de rendre l'ensemble de localisation compact en introduisant des bornes supérieures sur les variables  $u$ .

Le problème dual s'écrit

$$\min\{\gamma^T \mu \mid \sum_{i=1}^m a^i \mu_i \geq 0, \sum_{i=1}^m \mu_i = 1, \mu \geq 0\}.$$

Ensemble actif : une outil pour rendre le  
problème plus facile

**Observation** : en pratique, 10 à 20% des arcs sont saturés.

$\mathcal{A} = \mathcal{A}_1^* \cup \mathcal{A}_2^*$  partition des arcs saturés et non saturés. Par le théorème de complémentarité stricte, il existe une partition unique telle que

$$\left\{ \begin{array}{ll} u_a^* > 0 & \text{et} \quad \sum_{k \in \mathcal{K}} (x_a^k)^* = c_a, \quad \forall a \in \mathcal{A}_1^*, \\ u_a^* = 0 & \text{et} \quad \sum_{k \in \mathcal{K}} (x_a^k)^* < c_a, \quad \forall a \in \mathcal{A}_2^*. \end{array} \right.$$

## Relaxation lagrangienne partielle

Le problème dual devient

$$\max\{\mathcal{L}(u) \mid u_a \geq 0, a \in \mathcal{A}_1^*, u_a = 0, a \in \mathcal{A}_2^*\}.$$

Ce problème est équivalent à

$$\max\{\mathcal{L}_1(u_{\mathcal{A}_1^*}) \mid u_{\mathcal{A}_1^*} \geq 0\}$$

avec

$$\mathcal{L}_1(u_{\mathcal{A}_1^*}) = \min_{x \in X} (t + B_{\mathcal{A}_1^*}^T u_{\mathcal{A}_1^*})^T x,$$

où  $B_{\mathcal{A}_1^*}$  est la sous-matrice de la partition en lignes  $B = \begin{pmatrix} B_{\mathcal{A}_1^*} \\ B_{\mathcal{A}_2^*} \end{pmatrix}$ .

L'oracle est inchangé, mais le problème dual est maintenant défini dans un espace de dimension restreinte  $|\mathcal{A}_1^*| \ll |\mathcal{A}|$ .

## Test de saturation des arcs

$m$  appels à l'oracle aux points  $u^1, \dots, u^T$ .

$m$  réponses de l'oracle  $x^1, \dots, x^T$ , avec  $x^j \in X$

$Bx^j$  = utilisation des ressources par la solution de l'oracle.

En général,  $Bx^j \not\leq c$ , sinon,  $x^j$  serait optimal pour le problème de départ (9). Pour toute combinaison linéaire,  $\mu \geq 0$ ,  $\sum_{j=1}^T \mu_j = 1$ ,

$$x^\mu = \sum_{j=1}^T \mu_j x^j \in X.$$

$(b_i)^T x^j > c_i \Rightarrow$  la capacité de l'arc  $i$  est violée  $\Rightarrow i \in \mathcal{A}_1$

$(b_i)^T x^j \leq c_i \Rightarrow$  la capacité de l'arc  $i$  est respectée  $\Rightarrow i \in \mathcal{A}_2$

## Trouver la bonne combinaison convexe

Conditions d'optimalité du premier ordre au centre analytique :

$$\Phi'_k(u, z) = 0.$$

$$\rho(u - \bar{u}) - As^{-1} - u^{-1} = 0,$$

$$e^T s^{-1} - s_0^{-1} = 0,$$

$$A^T u + ze + s = \gamma$$

$$z - s_0 = \underline{\theta}.$$

Choisir comme vecteur de combinaison convexe

$$\mu = s^{-1}.$$

## Heuristique de mise à jour de la partition

Partition courante  $\mathcal{A} = \mathcal{A}_\infty \cup \mathcal{A}_\epsilon$ . L'ensemble  $\mathcal{A}_\infty$  approxime  $\mathcal{A}_\infty^*$  l'ensemble des arcs actifs à l'optimum. Vecteur de combinaison convexe  $\mu$  et variable duale  $u$ .

Si  $i \in \mathcal{A}_2$  et  $(b^i)^T x^\mu > c_i$ , alors

$$\mathcal{A}_2 = \mathcal{A}_2 \setminus \{i\} \quad \text{et} \quad \mathcal{A}_1 = \mathcal{A}_1 \cup \{i\}.$$

Si  $i \in \mathcal{A}_1$  et

1.  $(b^i)^T x^\mu \gg c_i$ ,
2.  $0 < u_i$  très petit,
3.  $u_i c_i \ll \sum_{k \in \mathcal{A}_1} c_k u_k$

alors

$$\mathcal{A}_1 = \mathcal{A}_1 \setminus \{i\} \quad \text{et} \quad \mathcal{A}_2 = \mathcal{A}_2 \cup \{i\}.$$

# Implémentation

Borne inférieure :  $\mathcal{L}(u^i)$  (fournie par l'oracle)

Borne supérieure :

$$\mu > 0, \sum \mu_j = 1 \text{ (fourni par ACCPM)}$$

1. Si  $Bx^\mu = \sum_j \mu_j Bx^j \leq c$ , alors  $t^T x^\mu$  est une borne supérieure.
2. Si  $Bx^\mu \not\leq c$  alors  $t^T x^\mu$  est une approximation d'une borne supérieure et le résidu  $r = (Bx^\mu - c)^+$  permet d'estimer l'erreur d'approximation.

En pratique, le cas 2 permet d'évaluer la qualité de la borne inférieure. Lorsque l'on approche du but, on diminue fortement le poids du terme proximal ( $\rho = 10^{-10}$ ) et on obtient toujours une solution  $x^\mu$  satisfaisant le cas 1.

Gestion du terme proximal

Pondération de l'objectif

Elimination de coupes inessentiels

## Initialisation

Tous les arcs sont d'abord déclarés inactifs. Résoudre les problèmes de plus court chemin (correspondant à  $u = 0$ ).

$$\mathcal{M} = \sum_{k \in \mathcal{K}} \min \left\{ \sum_{a \in \mathcal{A}} t_a x_a^k \mid N x^k = \delta^k, x_a^k \geq 0, \quad \forall a \in \mathcal{A} \right\}.$$

Créer la partition  $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$  à partir de la solution  $x^k$

$$\mathcal{A}_1 = \{a \mid \sum_{k \in \mathcal{K}} x_a^k \geq c_a\} \quad \text{et} \quad \mathcal{A}_2 = \{a \mid \sum_{k \in \mathcal{K}} x_a^k < c_a\}.$$

Démarrer l'algorithme avec  $u_{\mathcal{A}_1} > 0$  (en pratique, prendre  $u_a = 10^{-3}$ ).

## Itération de base

1. Résoudre (oracle) les problèmes de plus courts chemins  $\mathcal{M}_{\mathcal{A}_1}(u_{\mathcal{A}_1})$ . Obtenir la fonction objective  $\mathcal{L}(u_{\mathcal{A}_1})$ , un nouveau super-gradient  $a \in -\partial(-\mathcal{L}(u_{\mathcal{A}_1}))$  et un vecteur de flot sur des chemins  $\Pi_j$  ( $a = \Pi_j - c$ ). Cette information définit un nouveau plan coupant

$$a^T(u'_{\mathcal{A}_1} - u_{\mathcal{A}_1}) + \mathcal{L}(u_{\mathcal{A}_1}) \geq \mathcal{L}(u'_{\mathcal{A}_1}), \quad \text{for all } u'_{\mathcal{A}_1}.$$

2. Proximal-ACCPM calcule un nouveau point d'appel  $u_{\mathcal{A}_1}$ , la variable duale  $\mu$  (et le flot correspondant  $y = \sum_{t=1}^T \mu_t \Pi_t$ ) and une borne supérieure  $\bar{\theta}$  (exacte or approchée).
3. Mettre à jour la borne inférieure par

$$\underline{\theta} = \max(\underline{\theta}, \mathcal{L}(u_{\mathcal{A}_1})).$$

4. Si le défaut d'optimalité relatif est inférieur au seuil donné, STOP
5. Mettre à jour la partition  $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$  à l'aide du vecteur de flot  $y$  et des variables duales  $u_{\mathcal{A}_1}$ .
  - Tous les arcs (inactifs)  $a \in \mathcal{A}_2$  tels que  $y_a \geq c_a$  sont déclarés actifs.
  - Tous les arcs  $a \in \mathcal{A}_1$  tels que  $u_a \leq \eta_1$ ,  $y_a \leq \eta_2 c_a$  et

$$u_a c_a \leq \eta_3 \sum_{a \in \mathcal{A}_1} u_a c_a,$$

sont déclaré inactifs.

# Expériences numériques

## Problèmes test.

Problem ID	$ \mathcal{N} $	$ \mathcal{A} $	$ \mathcal{K} $	$z^*$	$\% \frac{ \mathcal{A}_1^* }{ \mathcal{A} }$
planar problems					
planar300	300	1680	3584	$6.89982 \times 10^8$	7.4
planar500	500	2842	3525	$4.81984 \times 10^8$	2.0
planar800	800	4388	12756	$1.16737 \times 10^8$	3.0
planar1000	1000	5200	20026	$3.44962 \times 10^9$	9.6
planar2500	2500	12990	81430	$1.26624 \times 10^{10}$	14.7
grid problems					
grid7	400	1520	400	$2.58641 \times 10^7$	7.0
grid11	625	2400	3000	$3.29259 \times 10^8$	11.0
grid13	900	3480	12000	$1.15932 \times 10^9$	8.0
grid15	1225	4760	32000	$3.59353 \times 10^9$	4.0
Telecommunication-like problems					
904	106	904	11130	$1.37850 \times 10^7$	9.2
Transportation problems					
Winnipeg	1067	2975	4345	$2.94065 \times 10^7$	2.0
Barcelona	1020	2522	7922	$3.89400 \times 10^7$	0.4
Chicago-sketch	933	2950	93513	$5.49053 \times 10^6$	1.0
Chicago-region*	12982	39018	2297945	$3.05199 \times 10^6$	0.6
Philadelphia	13389	40003	1151166	$1.65428 \times 10^7$	0.4

## *Proximal-Accpm vs. Accpm*

Pour garantir l'existence d'un centre analytique dans Accpm, il faut que l'ensemble de localisation soit compact. Des bornes supérieures artificielles sur les variables duales  $u$  assurent qu'il en sera bien ainsi. Ces bornes sont fixées uniformément dans nos expériences à une grande valeur  $10^6$ . En revanche, le centre analytique proximal existe toujours, indépendamment de toute hypothèse de compacité.

Les tests ont été effectués sur le même algorithme (avec bornes pour l'un et un terme proximal pour l'autre). On n'a pas utilisé de stratégie d'ensemble actif.

**Les deux méthodes se comportent de manière identique**

(moins de 5% de différence.)

## *Impact d'une stratégie d'ensemble actif*

Problem ID	$\% \frac{ A_1 }{ A }$	$\% \frac{ A_1^* }{ A }$	Outer	Inner	CPU	%Oracle	CPU improv.
planar300	9.1	7.4	343	835	45.3	72	<b>3.5</b>
planar500	2.6	2.0	140	359	32.9	95	<b>3.8</b>
planar800	3.6	3.0	317	786	240	95	<b>3.2</b>
planar1000	10.4	9.6	1249	2860	2501	68	<b>4.5</b>
planar2500	15.8	14.7	2643	7160	58448	54	$\infty$
grid7	9.3	7.0	264	704	25.2	68	<b>3.5</b>
grid11	12.2	11.0	569	1391	243.4	69	<b>3.5</b>
grid13	9.6	8.0	558	1333	455.8	81	<b>3.4</b>
grid15	4.9	4.0	364	902	497.1	95	<b>3.1</b>
904	13.1	9.2	321	984	21	47	<b>1.7</b>
Sioux-Falls	7.9	2.6	13	56	0.13	46	<b>2.4</b>
Winnipeg	2.9	2.0	158	393	27.6	92	<b>3.8</b>
Barcelona	0.5	0.4	35	111	4.3	93	<b>11.0</b>
Chicago-sketch	1.2	1.0	60	195	30	98	<b>4.4</b>
Chicago-region	-	-	-	-	-	-	-
Philadelphia	0.6	0.4	193	529	16487	99	$\infty$

## Elimination de colonnes

Problem ID	Nb cuts	Outer	Inner	CPU	%Or	Ratios	
planar300	199	325	721	33.8	79	1.3	4.5
planar500	76	118	258	26.1	96	1.3	4.9
planar800	168	252	545	176.8	97	1.4	4.5
planar1000	628	890	1904	1279	80	1.9	8.6
planar2500	2089	3009	7546	54700	61	$\infty$	$\infty$
grid7	159	228	582	17.4	76	1.5	5.2
grid11	329	458	1094	155.6	79	1.6	5.6
grid13	343	460	1086	323.4	89	1.4	4.8
grid15	206	294	708	381.3	97	1.3	4.0
904	171	311	819	15.3	51	1.4	4.3
Sioux-Falls	2.6	13	56	0.13	46	1.0	2.4
Winnipeg	93	143	372	24.6	94	1.1	4.2
Barcelona	35	35	111	4.3	93	1.0	11.0
Chicago-sketch	44	65	170	30	98	1.0	4.4
Chicago-region*	239	332	1026	35788	99	$\infty$	$\infty$
Philadelphia	127	192	525	15995	99	1.0	$\infty$

Tous les problèmes sont résolus à  $10^{-5}$  (sauf Chicago-region à  $10^{-4}$ )

## *Comparaison avec ALA : solutions approchées*

Les meilleurs résultats numériques trouvés sont

T. Larsson et Di Yuan. An augmented lagrangian algorithm for large scale multicommodity routing. *Computational Optimization and Applications*, 27(2) :187–215, 2004.

Les auteurs utilisent une méthode de Lagrangien augmenté pour trouver des solutions réalisables mais avec une précision limitée (environ  $10^{-3}$ ).

Leurs expériences sont faites sur une machine environ 10 fois plus lente, mais avec même RAM.

## Comparaison avec ALA

Problem ID	Proximal-ACCPM				ALA		Ratio
	Outer	CPU	%Or	Gap	CPU	Gap	CPU
planar300	56	4.6	78	0.0018	103.09	0.0016	<b>2.2</b>
planar500	27	5.5	94	0.001	211.22	0.0017	<b>3.8</b>
planar800	41	26.3	97	0.0014	1572.33	0.0017	<b>6.0</b>
planar1000	108	113.4	94	0.0021	3097.22	0.0018	<b>2.7</b>
planar2500	229	2218	96	0.0018	34123.14	0.0013	<b>1.6</b>
grid7	23	1.4	71	0.0014	8.1	0.0016	0.57
grid11	34	8.34	90	0.0015	262.54	0.0014	<b>3.15</b>
grid13	26	14.8	95	0.0014	948.26	0.0019	<b>6.4</b>
grid15	21	24.9	98	0.0012	2835.03	0.0014	<b>11.4</b>
Sioux-Falls	8	0.08	10	0.0023	0.47	0.0043	0.09
Winnipeg	65	10.5	92	0.00055	239.2	0.00061	<b>2.2</b>
Barcelona	30	3.7	91	0.00034	283.64	0.00057	<b>7.6</b>

## Problèmes non résolus avec ALA

Problem ID	$\% \frac{ \mathcal{A}_1 }{ \mathcal{A} }$	Outer	Inner	CPU	%Or	Gap
904	14.5	248	721	13.7	50	$10^{-3}$
Chicago-sketch	1.2	12	50	5.3	98	$10^{-3}$
Chicago-region	2.9	12	274	838	99	0.026
Chicago-region	2.3	111	463	12192	99	$10^{-3}$
Philadelphia	1.2	8	145	442	99	0.012
Philadelphia	1.1	40	203	3336	99	$10^{-3}$

Solutions optimales à  $10^{-3}$

## Comparaison avec d'autres méthodes

Network	CPLEX5.0	DW	Bundle	Prox-ACCPM
planar100	3034	2.1	19	7
planar150	94621	19	167	84
planar300	-	54	382	34
planar500	-	191	1297	26
planar800	-	4763	-	177
planar1000	-	14200	-	1279
grid8	>254349	23	71	132
grid9	-	92	259	292
grid10	-	184	644	318
grid11	-	408	1921	156
grid12	-	696	4534	199
grid13	-	3361	-	323
grid14	-	7564	-	308
grid15	-	28179	-	381

D'après Larsson et Yuang. (Les temps CPU de L&Y CPUs ont été **divisés par 10** pour permettre les comparaisons.)