Using Branch-and-Price to solve Origin-Destination Integer Multicommodity Flow Problems

Article ORATT - Master 2 IAD

Yasmina Seddik & Nicolas Cheifetz
Enseignant : Michel Minoux

Université Pierre et Marie Curie
Master 2 IAD

20 mars 2009

Plan

Cadre du problème

Problème **O**rigin-**D**estination **I**nteger **M**ulti**C**ommodity **F**low (

Problème multi-flots en nombres entiers, où 1 flot = (origine, destination) passe par un chemin unique.

Exemples d'applications :

- partage de la bande passante
- n'importe quel problème de flot

Quelques notations

Soit le graphe $G = (N, A)$.
Soit K, l'ensemble des types de flot du multi-flots.

Variables communes aux deux formulations :

- q^k= valeur de la k^e demande
- d_{ij}= capacité de l'arc $(i, j) \in A$

Formulation conventionnelle ou "noeud-arc"

Variables :

- $x_{ij}^k = \begin{cases} 1 & , si\ q^k\ passe\ par\ (i,j) \\ 0 & , sinon \end{cases}$
- c_{ij}^k = coût unitaire du k^e flot sur l'arc (i,j)
- $b_i^k = \begin{cases} 1 & , si\ i\ est\ le\ noeud\ origine\ du\ k^e\ flot \\ -1 & , si\ i\ est\ le\ noeud\ destination\ du\ k^e\ flot \\ 0 & , sinon \end{cases}$

Première formulation du PLNE :

$$\min_x \sum_{k\in K}\sum_{ij\in A} c_{ij}^k \cdot q^k \cdot x_{ij}^k$$

$$\begin{cases} \sum_{k\in K} q^k \cdot x_{ij}^k \cdot \delta_{ij}^p \leq d_{ij} & ,\forall (i,j)\in A \\ \sum_{ij\in A} x_{ij}^k - \sum_{ji\in A} x_{ji}^k = b_i^k & ,\forall k\in K \end{cases}$$

où $x_{ij}^k \in \{0,1\} \quad ,\forall (i,j)\in A$ et $\forall k \in K$

Formulation "chemin" ou "génération de colonnes"

Variables :

- $P(k)$ = ensemble des chemins pour le k^e flot
- $y_p^k = \begin{cases} 1, & si\ tout\ q^k\ passe\ par\ p \\ 0, & sinon \end{cases}$
- c_p^k = coût unitaire du k^e flot sur le chemin p
- $\delta_{ij}^p = \begin{cases} 1, & si\ l'arc\ (i,j)\ est\ contenu\ dans\ le\ chemin\ p \\ 0, & sinon \end{cases}$

Deuxième formulation du PLNE :

$$\min_y \sum_{k\in K}\sum_{p\in P(k)} c_p^k \cdot q^k \cdot y_p^k$$

$$\begin{cases} \sum_{k\in K}\sum_{p\in P(k)} q^k \cdot y_p^k \cdot \delta_{ij}^p \leq d_{ij} & ,\forall (i,j)\in A \\ \sum_{p\in P(k)} y_p^k = 1 & ,\forall k\in K \end{cases}$$

où $y_p^k \in \{0,1\} \quad ,\forall k\in K$ et $\forall p \in P(k)$

Dual de la 2^{de} formulation

Dual de la 2^{de} formulation du PLNE :

$$\max_{\pi,\sigma} \sum_{ij\in A} -\pi_{ij}\cdot d_{ij} + \sum_{k\in K}\sigma_k$$

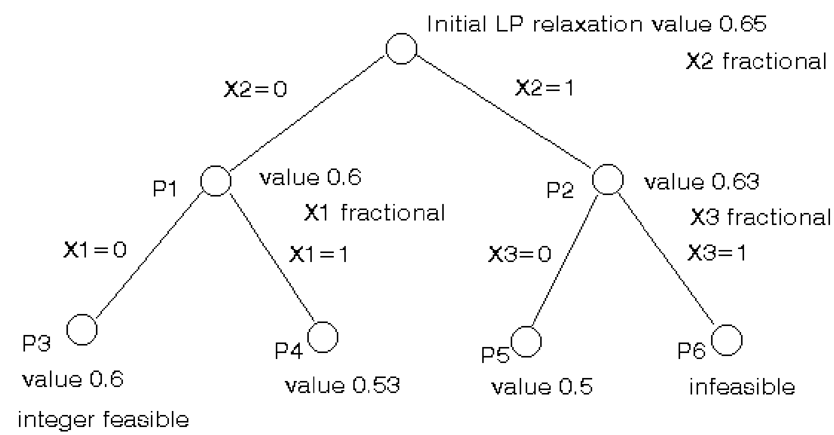
$$\begin{cases} \sum_{ij\in A} q^k \cdot \delta_{ij}^p \cdot (-\pi_{ij}) + \sigma_k \leq c_p^k \cdot q^k & ,\quad \forall k\in K \\ & \quad\ \forall p \in P(k) \end{cases}$$

où $\left| \begin{array}{l} -\pi_{ij} \geq 0 \ \Leftrightarrow\ \pi_{ij}\leq 0 \ ,\forall (i,j)\in A \\ \sigma_k \in \mathbb{R} \ ,\forall k\in K \end{array}\right.$

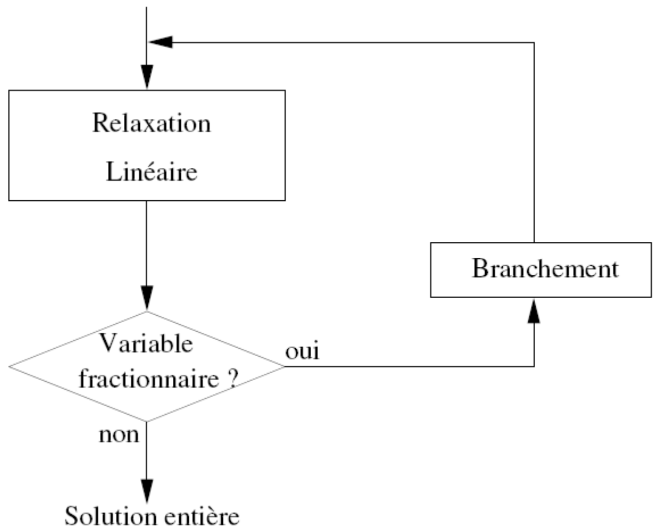
	Nb de contraintes	Nb de variables
1^{re} formulation	$\#A + \#N \times \#K$	$\#A \times \#K$
2^{de} formulation	$\#A + \#K$	$\sum_{k\in K} \#P(k)$
Dual de la 2^{de} formulation	$\sum_{k\in K} \#P(k)$	$\#A + \#K$

Représentation graphique du Branch&Bound

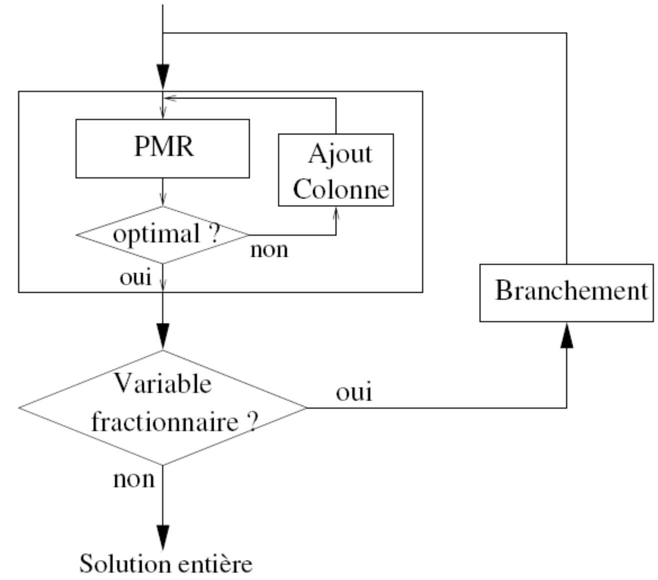
- Branch and Bound sur un PLNE : méthode exacte



Représentation graphique du Branch&Bound



Représentation graphique du Branch&Price



Pourquoi une nouvelle règle de branchement ?

Fixer les variables $\Rightarrow$ Destruction de la structure du Pricing problem

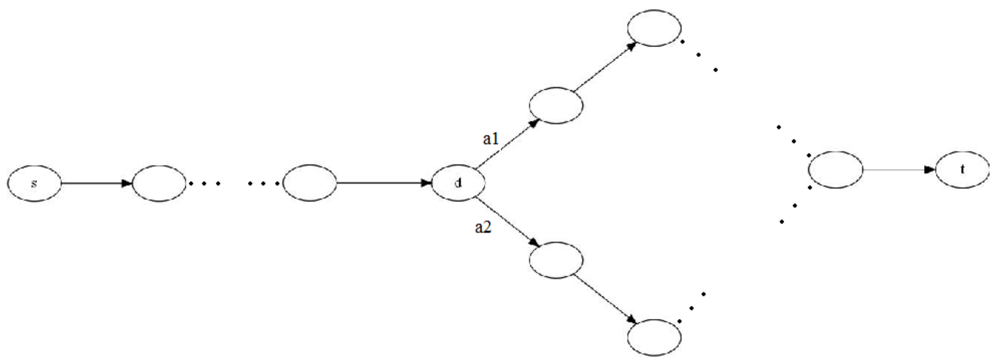
- Fixer $y_p^k = 0 \Leftrightarrow$ interdire au flot k d'emprunter le chemin p
  Problème : l'algorithme de génération va peut-être sélectionner cette variable comme entrante !

- Fixer $x_{ij}^k = 1 \Leftrightarrow$ imposer au flot k d'emprunter l'arc ij
  Problème : à une profondeur ≥ 2 il s'agit d'imposer le passage par un ensemble d'arcs !

Règle de branchement : Notations (1)

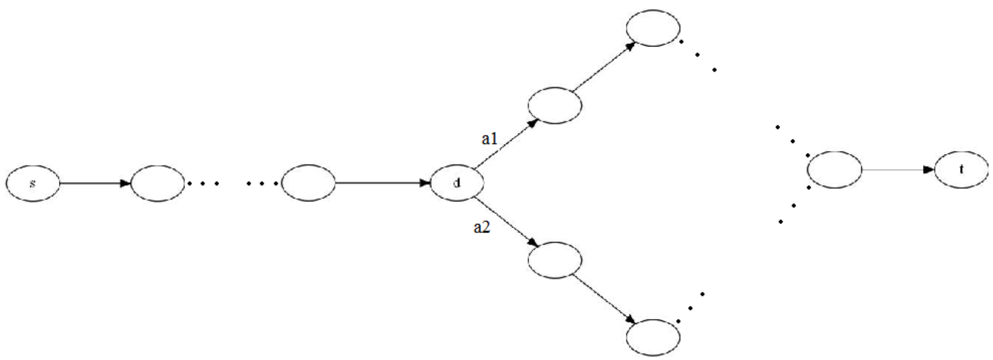
Notations :

- k : flot “splité” ayant la demande q^k maximale $[k = (s,t)]$
- p et p' : chemins transportant les deux plus grosses fractions du flot k.
  On a : $longueur(p) \leq longueur(p')$.
- d : noeud de divergence entre p et p'



Règle de branchement : Notations (2)

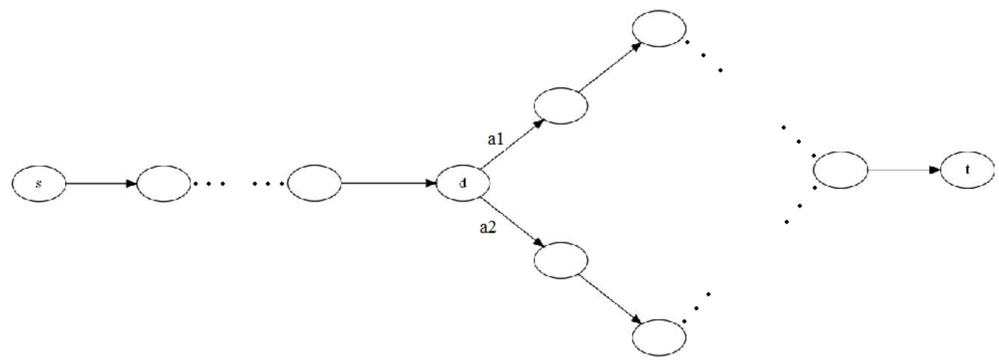
- a_1 : arc de p sortant de d
- a_2 : arc de p' sortant de d
- $A(d)$ = ensemble des arcs sortant de d
- $A(d, a_1)$ et $A(d, a_2)$: partition de $A(d)$
  - $a_1 \in A(d, a_1)$ et $a_2 \in A(d, a_2)$
  - $\#A(d, a1) \approx \#A(d, a2)$



Règle de branchement

Règle de branchement pour le flot k :

- sur une branche on interdit au flot k d'emprunter les arcs de $A(d, a_1)$
- sur l'autre branche on interdit au flot k d'emprunter les arcs de $A(d, a_2)$
- Coûts c_p^k très élevés pour les chemins p empruntant des arcs interdits



Sélection d'une branche

Choix d'une feuille à développer :

- Recherche par profondeur d'abord, avec priorité à la branche où p est autorisé : $A(d, a_2)$ interdit
- Dans ce cas, l'exploration par meilleure borne d'abord dégrade la solution

Génération de colonnes

Pricing Problem ($\rightarrow$ 2^e formulation du problème ODIMCF relâché)

Coût réduit de la variable y_p^k

$$\overline{c}_p^k = q^k \sum_{ij \in A} (c_{ij}^k + \pi_{ij}) \delta_{ij}^p - \sigma^k \quad = \sum_{ij \in A} c_{ij}^k \cdot \delta_{ij}^p \cdot q^k - \left(\sum_{ij \in A} (-\pi_{ij}) \cdot q^k \cdot \delta_{ij}^p + \sigma^k \right)$$

- On se place dans un graphe où chaque arc (i, j) a un coût $(c_{ij}^k + \pi_{ij})$
- On cherche : $p^* = \arg\min_{p \in P(k)} \overline{c}_p^k \quad , \forall k \in K$
  ($\hookrightarrow$ résolution de $\#K$ problèmes de plus court chemin)
- On a alors le coût réduit : $\boxed{\overline{c}_{p^*}^k = q^k \cdot c_{p^*}^k - \sigma^k}$

$$\forall k \in K, \begin{cases} \text{si } \overline{c}_{p^*}^k \geq 0 & \text{, alors le MP est resolu} \\ \text{sinon} & \text{, alors } y_{p*}^k \text{ entre dans le RMP} \end{cases}$$

Problème de symétrie !

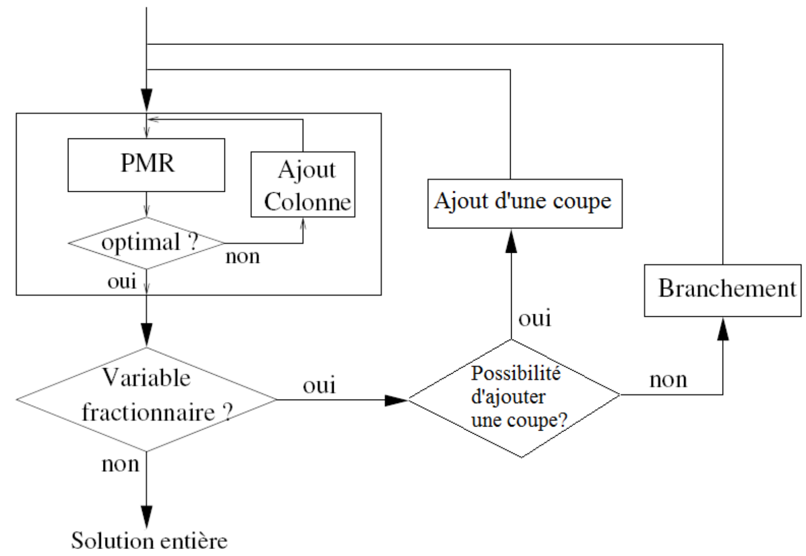
Effet de symétrie

Certains arcs interdits pour certains flots deviennent aussi interdits pour d'autres flots.

- (idée) plusieurs flots peuvent avoir un noeud de divergence commun
- Soit k^*, le flot "splité" sélectionné par la règle de branchement. Le flot k^* sera interdit sur un ensemble d'arc. Le même problème d'affectation peut se reproduire pour un autre flot ($\in K \backslash \{k^*\}$).

Il faut "couper" avant la décision de branchement.

Représentation graphique du Branch&Price&Cut



Génération de colonnes et de coupes (1)

Ajout d'une inégalité valide, qui est violée par la solution courante du PL.

- (idée) il faut déterminer des *classes d'inégalités valides* (non redondantes) pour l'enveloppe convexe des solutions.
  [solution fractionnaire $\not\rightleftarrows$ arc saturé]

Mise en oeuvre avec la 1^{re} formulation :

- on cherche les *arcs saturés* parmi les contraintes de capacités :

$$\sum_{k \in K} q^k \cdot x_{ij}^k \cdot \delta_{ij}^p \leq d_{ij} \qquad , \forall (i,j) \in A$$

- $\forall$ arc saturé, $\exists$ une ou plusieurs *couverture(s) minimale(s)*
  [$C \subset K$ est une *couverture* pour l'arc (i,j) si : $\sum_{k \in C} q^k > d_{ij}$]
- *Cover inequalities* : inégalités valides de la forme : $\sum_{k \in C} x_{ij}^k \leq |C| - 1$

Génération de colonnes et de coupes (2)

Inégalités "Lifted Cover"

$$\sum_{k \in C} x_{ij}^k + \sum_{k \in \overline{C}} \alpha_k \cdot x_{ij}^k \leq |C| - 1 \quad , \forall (i,j) \text{ arc saturé}$$

- C est une couverture minimale
- $\overline{C} = K \setminus C$
- $\alpha_k \in \mathbb{N}$ calculé en résolvant un problème de sac-à-dos
  $\Rightarrow \ \#\overline{C}$ problèmes de sac-à-dos à résoudre
- Déterminer la meilleure LCI : problème NP-difficile

Réécriture pour la 2^e formulation du PL :

- $x_{ij}^k = \sum_{p \in P(k)} y_p^k \cdot \delta_{ij}^p$
- **(LCI)** $\sum_{k \in C} \sum_{p \in P(k)} y_p^k \cdot \delta_{ij}^p + \sum_{k \in \overline{C}} \alpha_k \cdot \sum_{p \in P(k)} y_p^k \cdot \delta_{ij}^p \leq |C| - 1$

Préambule

On considère 15 problèmes ODIMCF.

- ∃ une capacité pour tous les flots des 14 premiers problèmes
- ∃ une capacité pour 96 des 130 flots du 15^{e} problème

Paramètres à faire varier :

- `commodities` = nombre de flots
- `nodes` = nombre de noeuds du problème
- `arcs` = nombre d'arcs du problème
- `satured arcs` = nombre d'arcs saturés à l'optimum du PL

Caractéristiques du système : IBM RS6000/590, MINTO 2.1, CPLEX 3.0

Caractéristiques des problèmes

problem	commodities	nodes	arcs	saturated arcs
1	35	14	16	8
2	68	24	24	7
3	70	29	61	31
4	58	18	29	16
5	47	19	25	6
6	93	27	37	27
7	93	23	29	18
8	41	28	31	8
9	87	24	42	24
10	41	19	19	8
11	23	14	16	5
12	81	26	36	10
13	52	29	31	8
14	46	20	23	8
15	585	50	130	28

B&B standard VS B&B customisé

		custom branching			standard branching		
problem	columns	nodes	gap	time	nodes	gap	time
1	180	41	0.00%	0.69	47	0.00%	0.62
2	295	5	0.00%	0.58	9	0.00%	0.68
3	489	57404	10.70%*	2989.80	233571	12.34%*	3600.00
4	347	2712	0.75%	29.69	5412	0.75%	55.72
5	232	21	0.23%	0.63	93	0.23%	1.19
6	463	3462	1.45%*	55.74	256333	3.56%*	3600.00
7	439	7813	0.17%	101.73	69156	0.17%	854.84
8	195	7	0.00%	0.45	5	0.00%	0.42
9	464	20147	2.86%*	318.29	292445	3.16%*	3600.00
10	182	47	0.00%	0.67	312	0.00%	2.33
11	106	21	0.00%	0.36	18	0.00%	0.34
12	420	663	0.27%	10.80	3201	0.27%	46.39
13	253	125	0.00%	1.55	107	0.00%	1.46
14	225	82	0.22%	1.20	148	0.22 %	1.62
15	2142	78577	4.76%*	3600.00	252648	—	3600.00

* : problème dont la solution optimale du PLNE est inconnue

Branch & Price

problem	columns	nodes	gap	time
1	195	45	0.00%	1.61
2	295	5	0.00%	0.88
3	9088	7618	10.3%	3600.00
4	1936	1252	0.00%	115.00
5	261	19	0.00%	1.40
6	12353	13274	2.79%	3600.00
7	9188	12808	0.00%	2637.40
8	195	7	0.00%	0.68
9	19184	8512	6.0%	3600.00
10	185	51	0.00%	2.10
11	109	25	0.00%	0.61
12	5939	6405	0.00%	1145.70
13	287	133	0.00%	7.20
14	461	156	0.00%	8.76
15	5156	3194	26.94%	3600.00

Branch & Price & Cut

problem	columns	rows	nodes	gap	time
1	180	7	131	0.00%	0.50
2	295	3	1	0.00%	0.88
3	7378	835	1564	3.63%	3600.00
4	418	88	17	0.00%	7.41
5	232	9	1	0.00%	0.63
6	4345	728	5174	2.16%	3600.00
7	493	154	28	0.00%	23.11
8	206	8	3	0.00%	0.84
9	8186	751	2084	1.16%	3600.00
10	190	23	7	0.00%	1.11
11	106	7	1	0.00%	0.31
12	430	54	4	0.00%	4.25
13	253	7	1	0.00%	0.74
14	248	37	4	0.00%	2.36
15	5836	69	2396	—	3600.00

Références

- Cynthia Barnhart, Christopher A. Hane and Pamela H. Vance
  Using Branch-and-Price to solve Origin-Destination Integer Multicommodity Flow Problems, 28 janvier 1998
- Michel Minoux
  Programmation Mathématique. Théorie et Algorithmes, 2008
- Jérôme Truffot, Christophe Duhamel, Philippe Mahey
  Branch & Price pour le problème du multiflot k-séparable de coût minimal, ROADEF05 - 14-16 Février 2005
  http://www.truffot.fr/jerome/public/publications/roadef05-pres.pdf
- A. Land et A. Doig
  Branch & Bound, 1960
- C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, et P.H. Vance
  Branch & Price, 1998