

Le problème du flot maximal avec contraintes sur le nombre de chemins

Jérôme Truffot, Christophe Duhamel, Philippe Mahey

jerome.truffot@isima.fr, christophe.duhamel@isima.fr, philippe.mahey@isima.fr

LIMOS, UMR 6158 - CNRS
Université Blaise Pascal
Clermont-Ferrand - France

FRANCORO IV - 18-21 Août 2004



Introduction

- ▶ Nouvelles technologies : réseaux optiques, protocole MPLS, ...
- ▶ MPLS-TE : Traffic Engineering (QoS, gestion)
- ▶ Contraintes sur le nombre de chemins
- ▶ Problème canonique : le flot maximal

Plan

Présentation du problème

- Définitions et notations
- Complexité
- Etat de l'art
- Cas particuliers

Modélisation

- Modèle arcs-sommets
- Modèle arcs-chemins

Algorithmes

- Algorithme d'approximation
- Méthode exacte
- Métaheuristique

Résultats Numériques

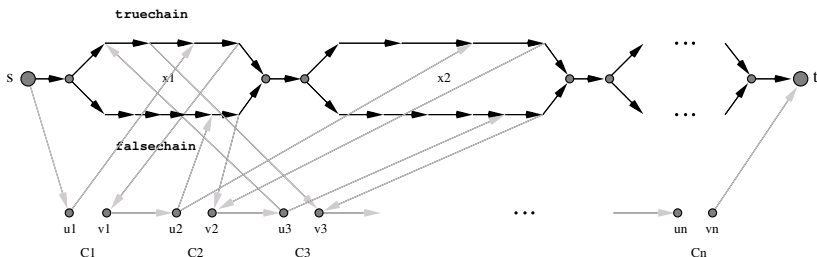
- Méthode exacte
- Méthodes approchées

Définitions et notations

- ▶ Graphe $G = (V, E)$ avec capacité u_e sur les arcs
- ▶ Flot k -séparable
- ▶ Une seule commodité $(s, t) \in V \times V$

Complexité du problème

- Réduction au SAT (Baier, Köhler et Skutella)



- Contrainte sur nombre de chemins \rightarrow problèmes NP-complets

Etat de l'art

- ▶ Edge Disjoint Path Problem (Survivability) : Menger (1982)
- ▶ Unsplittable Flow Problem : Kleinberg (1996), Atamtürk (2000)
- ▶ K -Splittable Flow Problem : Baier, Köhler et Skutella (2002)

Cas particuliers

- ▶ Flot uniforme :
 - ▶ quantité identique de flot routé par chaque chemin
 - ▶ Problème polynômial : Baier, Köhler et Skutella (2002)
 - ▶ Généralisation du théorème de flot maximal/coupe minimale
- ▶ Graphe uniforme :
 - ▶ capacité identique sur chaque arc
 - ▶ Théorème de Menger (1982)
 - ▶ Chemins disjoints

Modèle arcs-sommets : les variables

- ▶ Arc de retour $\bar{e} = (t, s)$ avec une capacité infinie,
 $E' = E \cup \{\bar{e}\}$

- ▶ Pour chaque chemin $k \in \{1, \dots, K\}$:
 - ▶ variables de flot : $x_e^k \geq 0 \quad \forall e \in E'$
 - ▶ variables de support de flot : $y_e^k \in \{0, 1\} \quad \forall e \in E'$

- ▶ Fonction objectif : $\max \sum_{k=1}^K x_e^k$

Modèle arcs-sommets : les contraintes sur le flot

- ▶ Contrainte de conservation de flot : $Ax^k = 0 \quad \forall k \in \{1, \dots, K\}$

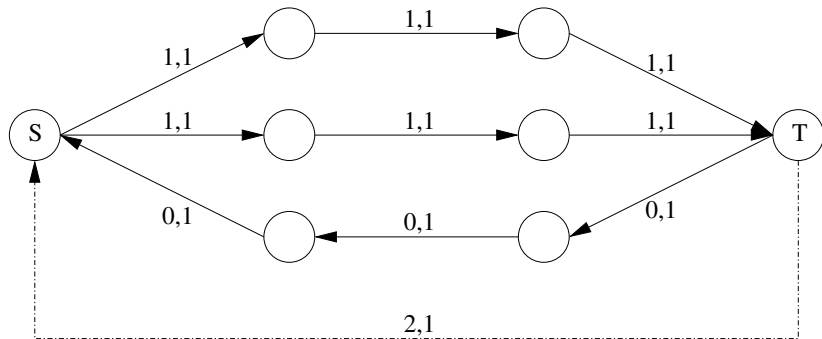
- ▶ Contrainte de capacité : $\sum_{k=1}^K x_e^k \leq u_e \quad \forall e \in E$

- ▶ Contrainte de support de flot :
 $x_e^k \leq u_e y_e^k \quad \forall e \in E \quad \forall k \in \{1, \dots, K\}$

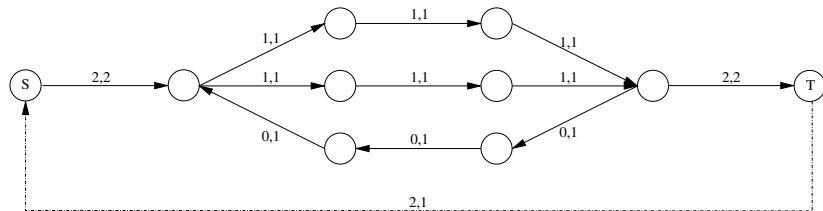
Modèle arcs-sommets : les contraintes sur le support de flot

- ▶ Contrainte de conservation de flot : $Ay^k = 0 \quad \forall k \in \{1, \dots, K\}$
- ▶ Contrainte d'intégrité : $y_e^k \in \{0, 1\} \quad \forall e \in E \quad \forall k \in \{1, \dots, K\}$
- ▶ Contrainte sur l'arc de retour : $y_e^k = 1 \quad \forall k \in \{1, \dots, K\}$

Modèle arcs-sommets : contre-exemples (1/2)



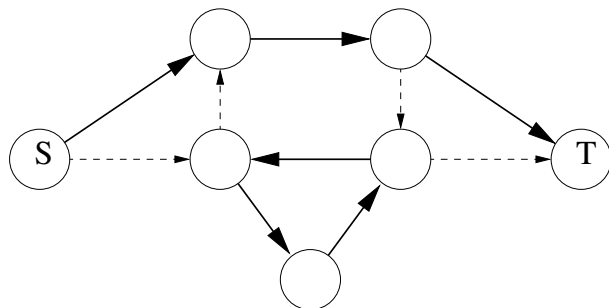
Modèle arcs-sommets : contre-exemples (2/2)



Modèle arcs-sommets : contrainte supplémentaire

- ▶ Contrainte sur les cocyles :

$$\sum_{e \in \omega^-(v)} y_e^k \leq 1 \quad \forall v \in V \quad \forall k \in \{1, \dots, K\}$$



Modèle arcs-sommets :

$$\max \sum_{k=1}^K x_e^k$$

s.c.

$$Ax^k = 0 \quad \forall k \in \{1, \dots, K\} \quad (1.a)$$

$$Ay^k = 0 \quad \forall k \in \{1, \dots, K\} \quad (1.b)$$

$$\sum_{k=1}^K x_e^k \leq u_e \quad \forall e \in E \quad (1.c)$$

$$x_e^k - u_e y_e^k \leq 0 \quad \forall e \in E \quad \forall k \in \{1, \dots, K\} \quad (1.d)$$

$$\sum_{e \in \omega^-(v)} y_e^k \leq 1 \quad \forall v \in V \quad \forall k \in \{1, \dots, K\} \quad (1.e)$$

$$x_e^k \geq 0 \quad \forall e \in E \quad \forall k \in \{1, \dots, K\} \quad (1.f)$$

$$y_e^k \in \{0, 1\} \quad \forall e \in E \quad \forall k \in \{1, \dots, K\} \quad (1.g)$$

$$y_e^k = 1 \quad \forall k \in \{1, \dots, K\} \quad (1.h)$$

Modèle arcs-chemins : les variables

- ▶ Soit P l'ensemble des chemins élémentaires de s à t
- ▶ Pour chaque chemin $p \in P$:
 - ▶ variable de flot : $x_p \geq 0$
 - ▶ variable de décision : $y_p \in \{0, 1\}$

- ▶ Fonction objectif : $\max \sum_{p \in P} x_p$

Modèle arcs-chemins : les contraintes

- ▶ Contrainte de capacité : $\sum_{p \in P} \delta_e^p x_p \leq u_e \quad \forall e \in E$
- ▶ Contrainte de couplage : $x_p \leq c_p y_p \quad \forall p \in P$
- ▶ Contrainte de nombre de chemins : $\sum_{p \in P} y_p \leq K$

Modèle arcs-chemins :

$$\max \sum_{p \in P} x_p$$

s.c.

$$\sum_{p \in P} \delta_e^p x_p \leq u_e \quad \forall e \in E \quad (2.a)$$

$$x_p - c_p y_p \leq 0 \quad \forall p \in P \quad (2.b)$$

$$\sum_{p \in P} y_p \leq K \quad (2.c)$$

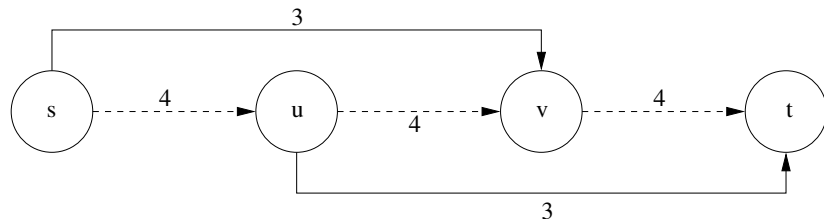
$$x_p \geq 0 \quad \forall p \in P \quad (2.d)$$

$$y_p \in \{0, 1\} \quad \forall p \in P \quad (2.e)$$

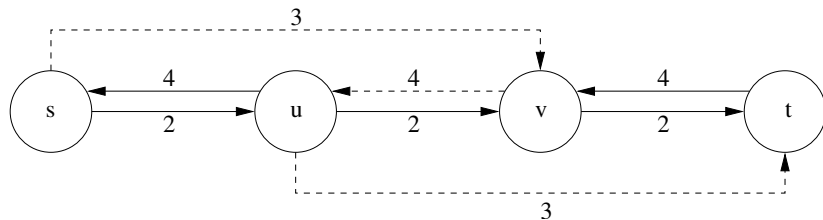
Algorithme d'approximation

- ▶ Flot uniforme : Baier, Köhler et Skutella (2002)
- ▶ A chaque itération i :
 - ▶ Flot uniforme maximal i -séparable : i chemins routant f_i
 - ▶ Graphe résiduel \overline{G}_i : capacité résiduelle $\overline{u}_e = \frac{u_e}{q_e^{i+1}}$
 - ▶ Calcul du chemin de capacité maximale $f_{i+1} \Rightarrow$ flot uniforme maximal $(i + 1)$ -séparable

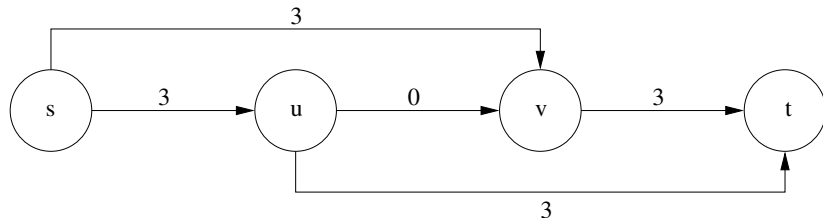
Algorithme d'approximation : exemple 1/3



Algorithme d'approximation : exemple 2/3

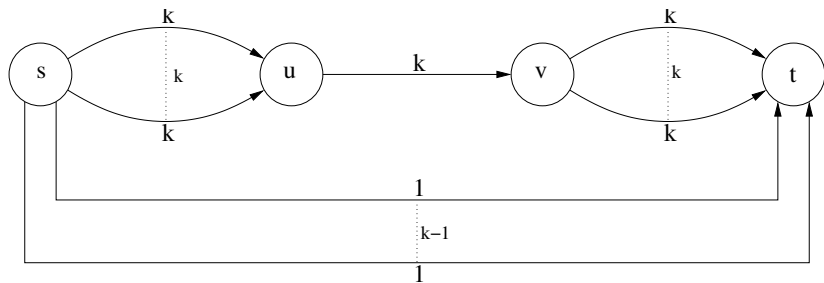


Algorithme d'approximation : exemple 3/3



Algorithme d'approximation : coefficient d'approximation

$$\frac{k-1}{2k-1} \leq \rho \leq \frac{1}{2}$$



Algorithme d'approximation : extensions

- ▶ Flot maximal sur un ensemble de chemins donné P_1
- ▶ Nouvel ensemble de chemins P_2 sur le graphe résiduel
- ▶ Flot maximal k -séparable sur l'union $P_1 \cup P_2$

Méthode exacte : Branch & Bound

- ▶ Branch & Bound : Land et Doig (1960)
- ▶ Utilisation du modèle arcs-sommets
- ▶ Borne supérieure : relaxation linéaire
- ▶ Stratégies de branchement :
 - ▶ première variable fractionnaire
 - ▶ *goulot d'étranglement*
 - ▶ ...

Méthode exacte : autres branchements

- ▶ Branchement de type Ryan et Foster (1981) :

- ▶ soit $J \subset E$ tel que $\sum_{e \in J} y_e^k \leq 1$ et $J' \subset J$

$$\text{branchement : } \left(\sum_{e \in J'} y_e^k = 0 \right) \text{ vs. } \left(\sum_{e \in J \setminus J'} y_e^k = 0 \right)$$

- ▶ branchement sur le cocycle sortant du premier nœud de divergence (Barnhart, Hane et Vance 1998)
- ▶ *goulot d'étranglement*
- ▶ Explicit-Constraint Branching : Applegate (1997)
 - ▶ soit $J_i^k \subset E$, ajout de la contrainte $\sum_{e \in J_i^k} y_e^k = z_i^k$ avec $z_i^k \in \mathbb{N}$
 - ▶ branchement en priorité sur les z_i^k puis de type Ryan et Foster

Métaheuristique

- ▶ Schéma de type GRASP : Feo (1989), Feo et Resende (1995)
- ▶ Présence de deux mécanismes
 - ▶ exploration globale : générateur de solutions initiales
 - ▶ exploration locale : stratégie d'amélioration locale
- ▶ Alternance des deux stratégies selon le contexte
- ▶ Structure décisionnelle hiérarchique (global/local)

Recherche Locale

- ▶ Utilisation de plusieurs générateurs de voisinage
 - ▶ voisinage 1 : ajout d'un chemin
 - ▶ voisinage 2 : suppression d'un chemin
 - ▶ voisinage 3 : transfert de flot entre deux chemins
- ▶ Stockage des meilleurs voisins dans une liste de capacité fixée
- ▶ Sélection aléatoire (avec critère d'aspiration)
- ▶ Critères d'arrêt : nombre d'itérations, stagnation

Générateur de solutions

- ▶ Fréquence d'utilisation pour chaque chemin (RL)
- ▶ Calcul d'un cumul d'utilisation pour chaque arc
- ▶ Capacité fictive inversement proportionnelle au cumul
- ▶ Pour $i = 1 \dots K$
 - ▶ calculer le chemin de capacité fictive maximale
 - ▶ router au maximum sur les capacités réelles
 - ▶ mettre à jour les capacités fictives

Améliorations

- ▶ Path relinking
 - ▶ utilisation d'un pool de solutions élites
 - ▶ à la fin de chaque RL
 - prendre la solution record, s_1
 - prendre une solution élite, s_2
 - évaluer les solutions sur un chemin de s_1 vers s_2
 - mettre à jour le pool

- ▶ Postoptimisation
 - ▶ à la fin de GRASP, $P' \subset P$ sous-ensemble des chemins générés
 - ▶ application du modèle arc-chemin sur P'
 - ▶ $P' \rightarrow$ génération de colonnes heuristique
 - ▶ permet de mesurer la qualité de la recherche

Résultats Numériques

- ▶ Instances :
 - ▶ 5 - 20 nœuds
 - ▶ densité de 4 - 5
 - ▶ génération aléatoire du graphe
 - ▶ s et t choisies aléatoirement
- ▶ Processeur pentium 2.6GHz, linux 2.6, 2Go RAM
- ▶ Compilateur gcc 3.3, ILOG CPLEX 8.0

Méthode exacte

Graphe	k	z^*	BB	BB + VO	RF	RF + VO	CPLEX
5 - 35	1	88	0.01	0.01	0.01	0.01	0.01
	2	170	0.01	0.01	0.01	0.01	0.02
	3	234	0.34	0.57	0.31	0.26	0.03
	4	291	1.22	2.51	3.73	0.22	0.28
	5	334	4738.61	479.13	1113.23	38.02	4.98
	6	375	-	-	-	396.33	1459.91
	7	412	-	-	-	1317.92	93181.55
10 - 40	1	63	0.10	0.01	0.04	0.01	0.01
	2	118	86.80	0.05	2.24	0.24	0.36
	3	164	-	3275.02	50.10	23.55	6.22
	4	199	-	-	-	749.10	2198.27

Stratégies de branchement : temps CPU (s)

Méthodes approchées

Graphe	k	z^*	BKS+	GRASP
5 - 35	1	88	88	88
	2	170	170	170
	3	234	218	234
	4	291	275	291
	5	334	284	334
	6	375	306	375
10 - 40	1	63	63	63
	2	118	112	118
	3	164	158	164
	4	199	176	199

Méthodes approchées : solutions obtenues

Conclusions et perspectives

- ▶ Modèles exacts limités malgré les améliorations
- ▶ Vers un schéma de Branch and Price

- ▶ Amélioration de l'approche de Baier et al.
- ▶ Peu de résultats sur les bornes d'approximation

- ▶ Métaheuristique GRASP efficace
- ▶ Améliorations en cours (postoptimisation + path relinking)

- ▶ Problèmes plus réalistes : multiflots de coût min