Sujet 5: Problème du flot de coût minimum

MSE3211A: Flot et Routage

(d'après Ahuja, R.K., T.L. Magnanti and J.B. Orlin, Prentice Hall, 1993,
et d'après les notes des cours de L.A. Wolsey et F. Vanderbeck)
Last update: January 10, 2011

Gautier Stauffer, MAB, Bur. A33:358, Email gautier.stauffer@math.u-bordeaux1.fr

PROBLÈME DE FLOT DE COÛT MINIMUM

Données : Soit $G = (N, A)$ un graphe orienté avec

- b_i : offre / demande au noeud $i \in N$
  - $b_i > 0$: i est un noeud source (offre)
  - $b_i < 0$: i est un noeud puits (demande)
  - $b_i = 0$: i est un noeud de transit
- u_{ij}: capacité de l'arc $(i, j) \in A$
- c_{ij} : coût unitaire pour pousser du flot sur l'arc $(i, j) \in A$

Hypothèses :

- Les données sont positives et entières
- Il existe un flot réalisable

Problème de Flot de coût minimum: Formulation

Problème :
Établir un flot de coût minimal qui satisfasse à toutes les demandes aux noeuds puits à partir des approvisionnements aux noeuds sources, tout en respectant les capacités des arcs.

$$\text{minimiser } z(x) = \sum_{(i,j)\in A} c_{ij} x_{ij}$$

$$\sum_{(i,j)\in\delta^+(i)} x_{ij} - \sum_{(j,i)\in\delta^-(i)} x_{ji} = b_i \quad \forall i \in N$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in A$$

TROUVER UN FLOT RÉALISABLE

- Vérifier que l' offre totale égale la demande totale:

$$\sum_{i \in N} b_i = 0$$

- Définir une source s et un puits t artificiels.
- Pour tout noeud source, i avec $b_i > 0$, créer un arc (s, i) de capacité $u_{s\,i} = b_i$.
- Pour tout noeud puits, i avec $b_i < 0$, créer un arc (i, t) de capacité $u_{i\,t} = -b_i$.
- On trouvera une solution réalisable pour le problème initial en résolvant le problème de flot maximum dans le réseau modifié.

Theorem

$\exists x$ tel que $\sum_{j \in N} x_{i\,j} - \sum_{j \in N} x_{j\,i} = b_i \quad \forall i$ and $x_{i\,j} \leq u_{i\,j}\ \forall (i, j)$ ssi le flot maximum de s à t dans le réseau modifié est $v = \sum_{i:\, b_i > 0} b_i = -\sum_{i:\, b_i < 0} b_i$

CNS pour la RÉALISABILITÉ

Theorem

$\exists$ flot x réalisable dans le réseau $G(A, N, b, 0, u)$ i.e. $\exists x_{ij}$ tel que

$$\begin{array}{l} \sum_{j:(i,j)\in A} x_{i\,j} - \sum_{j:(j,i)\in A} x_{j\,i} = b_i, \ \forall i \\ 0 \leq x_{i\,j} \leq u_{i\,j} \end{array}$$

$$\Longleftrightarrow$$

$$\forall\, S \subseteq N : \ \ b(S) \leq CAP[S, \overline{S}]$$

où $b(S) = \sum_{i\in S} b_i$ et
$CAP[S, \overline{S}] = \sum_{(i,j)\in(S,\overline{S})} u_{i,j}$

Conditions nécessaires:

$$\begin{array}{rccc} & \sum_{i\in S}(\sum_{j\in N} x_{i\,j} - \sum_{j\in N} x_{j\,i}) & = & \sum_{i\in S} b_i \\ \Rightarrow & \sum_{i\in S,\ j\in\overline{S}} x_{i\,j} - \sum_{j\in\overline{S},\ i\in S} x_{j\,i} & = & \sum_{i\in S} b_i \\ \Rightarrow & \sum_{i\in S,\ j\in\overline{S}} u_{i\,j} & \geq & \sum_{i\in S} b_i \end{array}$$

Conditions suffisantes: conséquence de max flot /min cut

Cas Particuliers/Applications

- Problème du plus court chemin de s vers t:
  - $b(s) = 1$, $b(t) = -1$, $b_i = 0$ pour les autres noeuds i
  - $u_{ij} = +\infty$ pour tous les arcs $(i, j) \in A$
- Problème des plus courts chemins de s vers tous les autres:
  - $b(s) = n - 1$, $b_i = -1$ pour les autres noeuds i
  - $u_{ij} = +\infty$ pour tous les arcs $(i, j) \in A$
- Problème de flot maximum:
  - $b_i = 0$ pour les noeuds i
  - $c_{ij} = 0$ pour tous les arcs $(i, j) \in A$
  - ajout d'un arc (t, s) avec $c_{ts} = -1$, et $u_{ts} = +\infty$

Cas Particuliers/Applications

- Problème d'affectation:
  - graphe bipartite où N_1 est un ensemble de n tâches à affecter à un ensemble N_2 de n processeurs ($|N_1| = |N_2|$)
  - les arcs $(i, j) \in A$ vont de N_1 vers N_2
  - $b_i = 1$ pour les noeuds $i \in N_1$
  - $b(j) = -1$ pour les noeuds $j \in N_2$
  - $u_{ij} = 1$ pour tous les arcs $(i, j) \in A$

  Exemples:
  Affectation de tâches à des travailleurs ou des machines

Cas Particuliers/Applications

- Problème de transport:
  - graphe bipartite où N_1 est un ensemble de fournisseurs qui desservent un ensemble N_2 de clients
  - les arcs $(i, j) \in A$ vont de N_1 vers N_2
  - $b_i > 0$ pour les noeuds $i \in N_1$
  - $b_j < 0$ pour les noeuds $j \in N_2$

- Problème de transport généralisé: idem mais graphe non bipartite avec noeuds de transit intermédiaires

  Exemples:

  Transport de matières premières, de produits finis vers les entrepôts, des entrepôts vers les magasins; transport de personnes.

Graphe résiduel

même idée que pour le problème de flot maximum, mais attention aux coûts et aux duplications d'arcs...

Soit x un flot réalisable, on note $G(x) = (N, A(x))$ le graphe défini par

- pour chaque arc (i, j) on definit deux arcs (i, j) et (j, i) avec
  - capacités résiduelles :

    $r_{ij} = u_{ij} - x_{ij}$ pour tout $(i, j) \in A$ tel que $x_{ij} < u_{ij}$
    $r_{ji} = x_{ij}$ pour tout $(i, j) \in A$ tel que $x_{ij} > 0$
  - coûts :

    c_{ij} inchangé pour tout $(i, j) \in A$ tel que $x_{ij} < u_{ij}$
    $c_{ji} = -c_{ij}$ pour tout $(i, j) \in A$ tel que $x_{ij} > 0$
- Attention si on a deja deux arcs (i, j) et (j, i) initialement, on se retrouve avec 4 arcs !

PROBLÈME DE FLOT à COÛT MINIMUM: ALGORITHME DES CYCLES DE COÛT NÉGATIF

- Construire un flot x réalisable.

- Tenter d'améliorer son coût:
  - **Tant qu'il existe un cycle de coût négatif** avec une capacité résiduelle (dans $G(x)$), envoyer du flot sur ce cycle jusqu'à sa saturation.
  - **S'il n'en existe plus**, le flot courant est **optimum**. En effet, 2 flots réalisables diffèrent seulement par des flots autour de cycles (cfr THM de décomposition des flots).

$$x^* \in X,\ x' \in X,\ x^* \neq x' \Rightarrow$$

$x' = x^* + \Delta x$ où Δx est un flot autour d'un cycle dans $G(x^*)$.

$$c(\Delta x) \geq 0 \Rightarrow cx' \geq cx^* \ .$$

PROPRIÉTÉS DE CET ALGORITHME

- L'algorithme est correct (termine à l'optimum).
- la **complexité** est: Si $-C \leq c_{ij} \leq C$ et $u_{ij} \leq U \;\; \forall (i,j) \in A$, le coût $c\, x$ est borné:

  $$-mCU \leq cx \leq mCU$$

  A chaque itération de l'algo., le coût décroît d'au moins 1,

  $$\Rightarrow O(mCU) \text{ itérations.}$$

  Donc, la complexité totale est

  $$O(\underbrace{nmU}_{\text{flow-max}} + \underbrace{n\,m}_{\text{plus-court-chemin}} .mCU) = O(nm^2CU)$$

- Les solutions sont entières (si les données sont entières):
  Théorème : Si toutes les données sont entières alors le flot optimal sur chaque arc est entier.

AUTRES ALGORITHMES

- Itérativement envoyer du flot le long d’un plus court chemin d’un noeud source à un noeud puit.
- Algorithmes utilisant la technique de “Scaling”.
- **“The network simplex algorithm”**