Problèmes de flots

Maël Forcier

REOP 2020

14/10/2020

Plan

Cadre

Soit $D = (V, A)$ un graphe orienté avec :

Cadre

Soit $D = (V, A)$ un graphe orienté avec :

- sur chaque arc $a \in A$ une capacité $u(a) \geq 0$

Cadre

Soit $D = (V, A)$ un graphe orienté avec :

- sur chaque arc $a \in A$ une capacité $u(a) \geq 0$
- deux noeuds spéciaux: une source s et un puit t (pour target en anglais)

Définition d'un $s - t$ flot

Définition d'un $s - t$ flot

Un $s - t$ flot est une fonction $f : A \longmapsto_{+}$ qui satisfait la loi de Kirchhoff

Définition d'un $s - t$ flot

Un $s - t$ flot est une fonction $f : A \longmapsto_{+}$ qui satisfait la loi de Kirchhoff

$$\forall v \in V \backslash \{s, t\}, \quad \sum_{a \in \delta^{-}(v)} f(a) = \sum_{a \in \delta^{+}(v)} f(a)$$

Définition d'un $s-t$ flot

Un $s-t$ flot est une fonction $f : A \longmapsto_{+}$ qui satisfait la loi de Kirchhoff

$$\forall v \in V \backslash \{s,t\}, \quad \sum_{a \in \delta^{-}(v)} f(a) = \sum_{a \in \delta^{+}(v)} f(a)$$

et les contraintes de capacité

Définition d'un $s-t$ flot

Un $s-t$ flot est une fonction $f : A \longmapsto_{+}$ qui satisfait la loi de Kirchhoff

$$\forall v \in V \backslash \{s,t\}, \quad \sum_{a \in \delta^{-}(v)} f(a) = \sum_{a \in \delta^{+}(v)} f(a)$$

et les contraintes de capacité

$$\forall a \in A, \quad f(a) \leq u(a)$$

Valeur d'un $s - t$ flot

La valeur d'un $s - t$ flow f est "la quantité qui sort de la source" :

Valeur d'un $s - t$ flot

La valeur d'un $s - t$ flow f est "la quantité qui sort de la source" :

$$\mathrm{val}(f) = \sum_{a \in \delta^+(s)} f(a) - \sum_{a \in \delta^-(s)} f(a)$$

Valeur d'un $s - t$ flot

La valeur d'un $s - t$ flow f est "la quantité qui sort de la source" :

$$\mathrm{val}(f) = \sum_{a \in \delta^+(s)} f(a) - \sum_{a \in \delta^-(s)} f(a)$$

Exercice: Calculer un $s - t$ flot maximal dans le graphe suivant :

images/flow_example.png

Définition d'une coupe

Une $s - t$ coupe (S, T) est une partition de l'ensemble des sommets $V = S \sqcup T$ telle que $s \in S$ et $t \in T$.

Définition d'une coupe

Une $s - t$ coupe (S, T) est une partition de l'ensemble des sommets $V = S \sqcup T$ telle que $s \in S$ et $t \in T$.

La capacité d'une coupe est la somme des capacités des arcs qui la traversent :

Définition d'une coupe

Une $s - t$ coupe (S, T) est une partition de l'ensemble des sommets $V = S \sqcup T$ telle que $s \in S$ et $t \in T$.
La capacité d'une coupe est la somme des capacités des arcs qui la traversent :

$$u(S, T) = \sum_{\substack{i \in S,\ j \in T \\ (i,j) \in A}} u(i, j)$$

On peut aussi définir la coupe comme un ensemble d'arcs qui $B = \delta^+(S)$ intersectent tous les $s - t$ chemins.

$$u(B) = \sum_{a \in B} u(a)$$

Flot maximal et coupe minimale

Flot maximal et coupe minimale

Problème du flot maximal

Trouver un $s - t$ flot de valeur maximale $\mathrm{val}(f)$

Flot maximal et coupe minimale

Problème du flot maximal

Trouver un $s - t$ flot de valeur maximale $\mathrm{val}(f)$

Problème de la coupe minimale

Trouve une coupe $s - t$ de capacité minimale $u(B)$

Coupe et flot : majoration

[6.3] Soit $f \leq u$ un $s - t$ flot et B une $s - t$ coupe.
Alors $\mathrm{val}(f) \leq u(B)$.

Coupe et flot : majoration

[6.3] Soit $f \leq u$ un $s-t$ flot et B une $s-t$ coupe.
Alors $\mathrm{val}(f) \leq u(B)$. Exercice: Calculer un $s-t$ flot maximal dans le graphe suivant :

images/flow_example.png

Graphe résiduel

Pour tout arc $a = (i, j) \in A$, on définit l'arc $\overleftarrow{a} = (j, i)$

Graphe résiduel

Pour tout arc $a = (i, j) \in A$, on définit l'arc $\overleftarrow{a} = (j, i)$ et les capacités résiduelles :

$$u_f(a) = u(a) - f(a) + f\left(\overleftarrow{a}\right)$$

Graphe résiduel

Pour tout arc $a = (i, j) \in A$, on définit l'arc $\overleftarrow{a} = (j, i)$ et les capacités résiduelles :

$$u_f(a) = u(a) - f(a) + f\left(\overleftarrow{a}\right)$$

images/residual_graph_1.png

Graphe résiduel

Le graphe résiduel est le graphe $D_f = (V, A_f)$ avec

$$A_f = \{a \in A \cup \overleftarrow{A} : \ u_f(a) > 0\}$$

munis des capacités u_f

Graphe résiduel

Le graphe résiduel est le graphe $D_f = (V, A_f)$ avec

$$A_f = \{a \in A \cup \overleftarrow{A} : \; u_f(a) > 0\}$$

munis des capacités u_f

images/residual_graph_2.png

Chemin dans le graphe résiduel

Un $s - t$ chemin dans le graphe résiduel D_f permet de trouver un flot de plus grande valeur.

Chemin dans le graphe résiduel

Un $s - t$ chemin dans le graphe résiduel D_f permet de trouver un flot de plus grande valeur.

images/residual_graph_3.png

Critère d'optimalité

Critère d'optimalité

Theorem (6.4)

Un $s - t$ flot f est maximal ssi il n'existe pas de chemin dans son graphe résiduel D_f.

Illustration du critère d'optimalité

Illustration du critère d'optimalité

images/residual_graph_4.png

Figure: Un $s - t$ flot maximal

Illustration du critère d'optimalité

images/residual_graph_4.png

Figure: Un $s - t$ flot maximal

Théorème Max flow / min cut

Théorème Max flow / min cut

Theorem (6.5)

La valeur maximale d'un $s-t$ *flot est égale à la capacité minimale d'une* $s-t$ *coupe.*

Ford-Fulkerson

[H] Algorithme de Ford-Fulkerson $f(a) := 0$ pour tout $a \in A$

Ford-Fulkerson

[H] Algorithme de Ford-Fulkerson $f(a) := 0$ pour tout $a \in A$ il existe un chemin dans le graphe résiduel

Ford-Fulkerson

[H] Algorithme de Ford-Fulkerson $f(a) := 0$ pour tout $a \in A$ il existe un chemin dans le graphe résiduel Sélectionner P un $s - t$ chemin dans le graphe résiduel Augmenter f le long de P par $\min_{a \in P} u_f(a)$

Ford-Fulkerson

[H] Algorithme de Ford-Fulkerson $f(a) := 0$ pour tout $a \in A$ il existe un chemin dans le graphe résiduel Sélectionner P un $s - t$ chemin dans le graphe résiduel Augmenter f le long de P par $\min_{a \in P} u_f(a)$ Retourner f

Programmation linéaire pour les flots maximaux

Programmation linéaire pour les flots maximaux

On peut reformuler le problème de flot maximal comme de la programmation linéaire :

Programmation linéaire pour les flots maximaux

On peut reformuler le problème de flot maximal comme de la programmation linéaire :

$$
\begin{aligned}
\max \quad & \sum_{a \in \delta^+(s)} x_a - \sum_{a \in \delta^-(s)} x_a \\
\text{s.t.} \quad & \sum_{a \in \delta^-(v)} x_a = \sum_{a \in \delta^+(v)} x_a \qquad \forall\ v \in V \backslash \{s, t\} \\
& 0 \leq x_a \leq u(a) \qquad \forall\ a \in A
\end{aligned}
$$

Propriété des LP de type max flow

Propriété des LP de type max flow

[6.11] La matrice des contraintes d'un LP de type max flow est totalement unimodulaire.

Propriété des LP de type max flow

[6.11] La matrice des contraintes d'un LP de type max flow est totalement unimodulaire.

[6.12] Le problème de coupe minimale est le dual du problème de flot maximal.

Cadre

Cadre

Soit $D = (V, A)$ un graphe orienté avec :

Cadre

Soit $D = (V, A)$ un graphe orienté avec :

- des bornes inférieures et supérieurs sur les capacités $0 \leq \ell(a) \leq u(a)$ pour chaque arc

Cadre

Soit $D = (V, A)$ un graphe orienté avec :

- des bornes inférieures et supérieurs sur les capacités $0 \leq \ell(a) \leq u(a)$ pour chaque arc
- des coûts $c(a) \geq 0$ pour chaque arc $a \in A$

Cadre

Soit $D = (V, A)$ un graphe orienté avec :

- des bornes inférieures et supérieurs sur les capacités $0 \leq \ell(a) \leq u(a)$ pour chaque arc
- des coûts $c(a) \geq 0$ pour chaque arc $a \in A$
- une "source" algébrique $b(v) \in$ à chaque sommet

Définition d'un *b*-flot

Définition d'un b-flot

Un b-flot est une fonction $f : A \longmapsto_{+}$ satisfaisant la loi de Kirchhoff

$$\forall v \in V, \quad b(v) + \sum_{a \in \delta^{-}(v)} f(a) = \sum_{a \in \delta^{+}(v)} f(a)$$

Définition d'un *b*-flot

Un b-flot est une fonction $f : A \longmapsto_{+}$ satisfaisant la loi de Kirchhoff

$$\forall v \in V, \quad b(v) + \sum_{a \in \delta^{-}(v)} f(a) = \sum_{a \in \delta^{+}(v)} f(a)$$

et les contraintes de capacité

$$\forall a \in A, \quad \ell(a) \leq f(a) \leq u(a)$$

Définition d'un b-flot

Un b-flot est une fonction $f : A \longmapsto_{+}$ satisfaisant la loi de Kirchhoff

$$\forall v \in V, \quad b(v) + \sum_{a \in \delta^-(v)} f(a) = \sum_{a \in \delta^+(v)} f(a)$$

et les contraintes de capacité

$$\forall a \in A, \quad \ell(a) \leq f(a) \leq u(a)$$

Si $b(v) = 0$ partout, on dit que f est une circulation.

Définition d'un *b*-flot

Un b-flot est une fonction $f : A \longmapsto_{+}$ satisfaisant la loi de Kirchhoff

$$\forall v \in V, \quad b(v) + \sum_{a \in \delta^-(v)} f(a) = \sum_{a \in \delta^+(v)} f(a)$$

et les contraintes de capacité

$$\forall a \in A, \quad \ell(a) \leq f(a) \leq u(a)$$

Si $b(v) = 0$ partout, on dit que f est une circulation.

Le coût d'un b-flot f est

$$c(f) = \sum_{a \in A} c(a) f(a)$$

On doit avoir $\sum_{v \in V} b(v) = 0$ si on veut satisfaire la loi de Kirchhoff.

Graphe résiduel et cycles

Graphe résiduel et cycles

Pour tout arc $a = (i, j) \in A$, on définit l'arc $\overleftarrow{a} = (j, i)$ et les capacités résiduelles :

$$u_f(i, j) = u(i, j) - f(i, j) + f(j, i) - l(j, i)$$

Graphe résiduel et cycles

Pour tout arc $a = (i, j) \in A$, on définit l'arc $\overleftarrow{a} = (j, i)$ et les capacités résiduelles :

$$u_f(i, j) = u(i, j) - f(i, j) + f(j, i) - l(j, i)$$

On étend c en définissant $\tilde{c}(i, j) = c(i, j) - c(j, i)$.

Graphe résiduel et cycles

Pour tout arc $a = (i,j) \in A$, on définit l'arc $\overleftarrow{a} = (j,i)$ et les capacités résiduelles :

$$u_f(i,j) = u(i,j) - f(i,j) + f(j,i) - l(j,i)$$

On étend c en définissant $\tilde{c}(i,j) = c(i,j) - c(j,i)$.

Le graphe résiduel est le graphe $D_f = (V, A_f)$, munis des capacités u_f et des coût $\tilde{c}$ avec

$$A_f = \{a \in A \cup \overleftarrow{A} : \ u_f(a) > 0\}$$

Graphe résiduel et cycles

Pour tout arc $a = (i, j) \in A$, on définit l'arc $\overleftarrow{a} = (j, i)$ et les capacités résiduelles :

$$u_f(i, j) = u(i, j) - f(i, j) + f(j, i) - l(j, i)$$

On étend c en définissant $\tilde{c}(i, j) = c(i, j) - c(j, i)$.

Le graphe résiduel est le graphe $D_f = (V, A_f)$, munis des capacités u_f et des coût $\tilde{c}$ avec

$$A_f = \{a \in A \cup \overleftarrow{A} : \ u_f(a) > 0\}$$

On peut construire un algorithme en regardant les cycles dans le graphe résiduel. On définit le coût d'un cycle par

$$c(C) = \sum_{a \in C} c(a)$$

Exercices

Monge's transportation problem (ex. 6.9)
Consider m holes that we want to fill using n piles of sand. Let us call s_i the mass of the i-th pile of sand and t_j the mass of sand necessary to fill the j-th hole. For each couple (i, j) we know the distance d_{ij} of the i-th pile to the j-th hole. If a mass x_{ij} is moved from pile i to hole j, the cost of the displacement is equal to $d_{ij}x_{ij}$. We want to find the transportation plan that allows the holes to be filled at the minimum cost.

Exercices

Seat alocation of a bus company (ex. 6.10)
A bus capable of carrying not more than B passengers will depart from city 1 and visit the cities $2, 3, \cdots, n$ successively. The number of passengers wanting to travel from city i to city j (with $i < j$) is $d_{i,j}$ ande the price of this trip is $p_{i,j}$. How many passengers should we take in each city to maximize total revenue ? Model this problem as a flow problem.

Exercices

Taxi fleet (ex. 6.13)
A cab company has p passenger journeys to complete over a day. For each of these journeys $i = 1, \ldots, p$, they know its departure location o_i and its departure time h_i, as well as the duration of the trip t_i and its arrival location d_i. Moreover, the time τ_{ji} to get from d_j to o_i is known for all couples (i, j). The company wants to minimize the number of cabs needed to satisfy the demand. All cabs are assumed to be located in one depot at the beginning of the day.