Recherche opérationnelle

DUT Info 2e année, parcours A

Programmation linéaire et flots

Florent Foucaud

IUT CLERMONT AUVERGNE

Aurillac - Clermont-Ferrand - Le Puy-en-Velay
Montluçon - Moulins - Vichy

Flot maximum dans un graphe

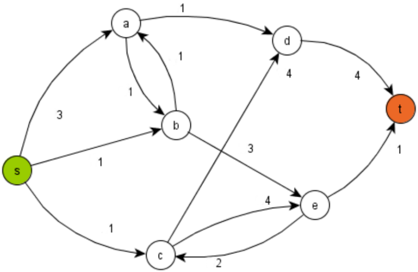
On considère un graphe orienté pondéré $G = (V, A, c)$ avec une capacité $c(\overrightarrow{xy}) \in \mathbb{R}^+$ pour chaque arc $\overrightarrow{xy} \in A$, une source s et une destination t.

Définition

Un flot est une assignation $f : A \to \mathbb{R}^+$ telle que :

1. pour tout arc $\overrightarrow{xy} \in A$: $f(\overrightarrow{xy}) \leq c(\overrightarrow{xy})$, *(respect des capacités)*
2. pour tout sommet $v \in V - \{s, t\}$: $\sum_{x:x\to v} f(\overrightarrow{xv}) = \sum_{y:v\to y} f(\overrightarrow{vy})$. *(conservation locale du flot)*

On cherche à maximiser la valeur $\sum_{y:s\to y} f(\overrightarrow{sy})$ du flot.



Flot maximum dans un graphe

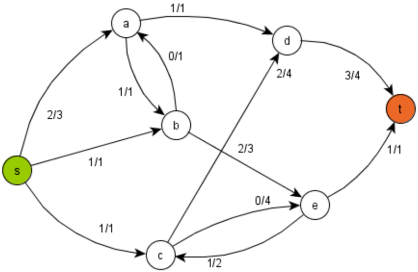
On considère un graphe orienté pondéré $G = (V, A, c)$ avec une capacité $c(\overrightarrow{xy}) \in \mathbb{R}^+$ pour chaque arc $\overrightarrow{xy} \in A$, une source s et une destination t.

Définition

Un flot est une assignation $f : A \to \mathbb{R}^+$ telle que :

1. pour tout arc $\overrightarrow{xy} \in A$: $f(\overrightarrow{xy}) \leq c(\overrightarrow{xy})$, *(respect des capacités)*
2. pour tout sommet $v \in V - \{s, t\}$: $\sum_{x:x\to v} f(\overrightarrow{xv}) = \sum_{y:v\to y} f(\overrightarrow{vy})$. *(conservation locale du flot)*

On cherche à maximiser la valeur $\sum_{y:s\to y} f(\overrightarrow{sy})$ du flot.



Flot maximum dans un graphe

On considère un graphe orienté pondéré $G = (V, A, c)$ avec une capacité $c(\overrightarrow{xy}) \in \mathbb{R}^+$ pour chaque arc $\overrightarrow{xy} \in A$, une source s et une destination t.

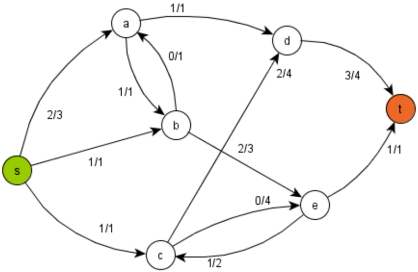
Définition

Un flot est une assignation $f : A \to \mathbb{R}^+$ telle que :

1. pour tout arc $\overrightarrow{xy} \in A$: $f(\overrightarrow{xy}) \leq c(\overrightarrow{xy})$, *(respect des capacités)*
2. pour tout sommet $v \in V - \{s, t\}$: $\sum_{x:x \to v} f(\overrightarrow{xv}) = \sum_{y:v \to y} f(\overrightarrow{vy})$. *(conservation locale du flot)*

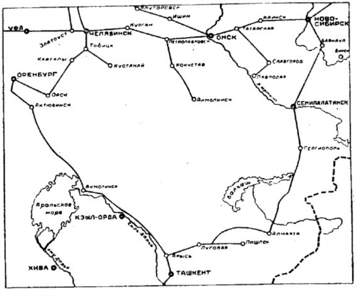
On cherche à maximiser la valeur $\sum_{y:s \to y} f(\overrightarrow{sy})$ du flot.

Applications : réseaux de transport



Flots : un peu d'histoire

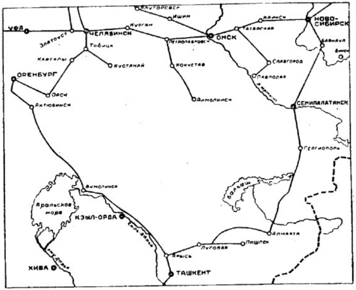
- 1930 : Problème formulé par A. N. Tolstoï, appliqué au réseau de trains en URSS



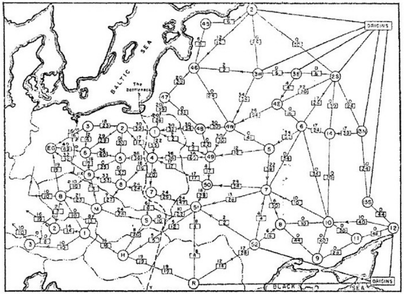
10 sources et 68 destinations soviétiques en 1930

Flots : un peu d’histoire

- 1930 : Problème formulé par A. N. Tolstoï, appliqué au réseau de trains en URSS
- 1954 : T. E. Harris et Gen. F. S. Ross étudient à leur tour le problème ! (rapport secret de l’US Air Force déclassifié en 1999)



10 sources et 68 destinations soviétiques en 1930



Le réseau ferré d’Europe de l’Est en 1954

Flots : un peu d'histoire

- 1930 : Problème formulé par A. N. Tolstoï, appliqué au réseau de trains en URSS
- 1954 : T. E. Harris et Gen. F. S. Ross étudient à leur tour le problème ! (rapport secret de l'US Air Force déclassifié en 1999)
- 1955 : Algorithme optimal de Ford-Fulkerson pour trouver le flot max

Lester R. Ford

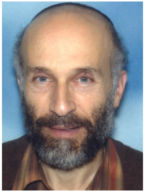
Delbert R. Fulkerson

Flots : un peu d'histoire

- 1930 : Problème formulé par A. N. Tolstoĭ, appliqué au réseau de trains en URSS
- 1954 : T. E. Harris et Gen. F. S. Ross étudient à leur tour le problème ! (rapport secret de l'US Air Force déclassifié en 1999)
- 1955 : Algorithme optimal de Ford-Fulkerson pour trouver le flot max
- 1970-72 : Algorithme de Dinitz-Edmonds-Karp, qui termine à tous les coups, en temps $O(|V| \cdot |A|^2)$ ou $O(|V|^2 \cdot |A|)$

Lester R. Ford

Delbert R. Fulkerson



Yefim Dinitz

Jack Edmonds

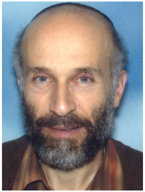
Richard M. Karp

Flots : un peu d'histoire

- 1930 : Problème formulé par A. N. Tolstoï, appliqué au réseau de trains en URSS
- 1954 : T. E. Harris et Gen. F. S. Ross étudient à leur tour le problème ! (rapport secret de l'US Air Force déclassifié en 1999)
- 1955 : Algorithme optimal de Ford-Fulkerson pour trouver le flot max
- 1970-72 : Algorithme de Dinitz-Edmonds-Karp, qui termine à tous les coups, en temps $O(|V| \cdot |A|^2)$ ou $O(|V|^2 \cdot |A|)$
- ...
- 2013 : Meilleur algorithme à ce jour, en temps $O(|V| \cdot |A|)$ (J. B. Orlin)


Lester R. Ford


Delbert R. Fulkerson


Yefim Dinitz


Jack Edmonds


Richard M. Karp


James B. Orlin

Flots sous forme de programmes linéaires

On considère un graphe orienté pondéré $G = (V, A, c)$ avec une capacité $c(\overrightarrow{xy}) \in \mathbb{R}^+$ pour chaque arc $\overrightarrow{xy} \in A$, une source s et une destination t.

Définition

Un flot est une assignation $f : A \to \mathbb{R}^+$ telle que :

1. pour tout arc $\overrightarrow{xy} \in A$: $f(\overrightarrow{xy}) \leq c(\overrightarrow{xy})$, *(respect des capacités)*
2. pour tout sommet $v \in V - \{s, t\}$: $\sum_{x : x \to v} f(\overrightarrow{xv}) = \sum_{y : v \to y} f(\overrightarrow{vy})$. *(conservation locale du flot)*

On cherche à maximiser la valeur $\sum_{y : s \to y} f(\overrightarrow{sy})$ du flot.

Flots sous forme de programmes linéaires

On considère un graphe orienté pondéré $G = (V, A, c)$ avec une capacité $c(\overrightarrow{xy}) \in \mathbb{R}^+$ pour chaque arc $\overrightarrow{xy} \in A$, une source s et une destination t.

Définition

Un flot est une assignation $f : A \to \mathbb{R}^+$ telle que :

1. pour tout arc $\overrightarrow{xy} \in A$: $f(\overrightarrow{xy}) \leq c(\overrightarrow{xy})$, *(respect des capacités)*
2. pour tout sommet $v \in V - \{s, t\}$: $\sum_{x:x\to v} f(\overrightarrow{xv}) = \sum_{y:v\to y} f(\overrightarrow{vy})$. *(conservation locale du flot)*

On cherche à maximiser la valeur $\sum_{y:s\to y} f(\overrightarrow{sy})$ du flot.

Programme linéaire correspondant :

On prend une variable f_a pour chaque arc a.

$$\begin{array}{lcccc}
\text{maximiser :} & \sum_{y:s\to y} f_{\overrightarrow{sy}} & & & \\
\text{tel que :} & f_a & \leq & c(a) & \forall a \in A \\
 & f_a & \geq & 0 & \forall a \in A \\
 & \sum_{x:x\to v} f_{\overrightarrow{xv}} & = & \sum_{y:v\to y} f_{\overrightarrow{vy}} & \forall v \in V
\end{array}$$