Optimisation dans les réseaux

F. Bendali-Mailfert
bendali@isima.fr

Bur. D119 Bat. Isima

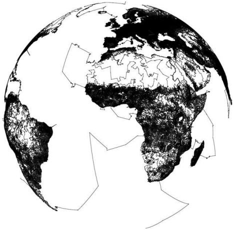

Image issue de [2]

Plan du cours

- La Recherche Opérationnelle : Applications et entreprises impliquées [1]
- Problèmes de réseaux et structures de modélisation.
- Domaine des télécommunications
  - Topologie et routage dans les réseaux filaires
  - Topologie dans les réseaux sans fil.

Principaux problèmes dans les réseaux :

- Le dimensionnement
  Rechercher les capacités à mettre sur les liaisons dans un réseau.
- La topologie (design)
  Rechercher la structure : nœuds principaux et liens.
- Le routage
  Rechercher "les meilleurs" chemins

Rappels de graphes. Définitions et notations

- Graphe $G(V, E)$ non orienté
- Graphe $G(V, E)$ orienté
- Extrémité d'une arête, d'un arc.
- Incidence. Adjacence
- Graphe complet
- Sous graphe. Sous graphe Induit. Graphe partiel.

- Chaîne. Chemin
- Cycle. Circuit. Graphe acyclique.
- Connexité : Un graphe est connexe ssi entre tout couple de sommets existe une chaîne.
- Forte connexité : Un graphe est fortement connexe ssi entre tout couple de sommets existe un chemin.
- Une Coupe dans un graphe $G = (V, E)$ est un sous ensemble d'arêtes dont une extremité exactement est dans $S \subset V$. Notation $[S, V \setminus S] = \delta(S) = \{e = uv \in E : u \in S, v \in V \setminus S\}$
- Un arbre $A = (V, E)$ est un graphe connexe sans cycle. Un arbre est tel que :
  - $|E| = |V| - 1$,
  - il existe exactement une chaîne entre tout couple de sommets
  - sans cycle maximal
  - connexe minimal

Un graphe $G = (V, E)$ est biparti si $V = V^1 \cup V^2$ avec $V^1 \cap V^2 = \emptyset$ et $E = \delta(V^1) = \delta(V^2)$.

Tous les cycles d'un graphe biparti sont de longueur paire.

Un réseau est un graphe orienté $G = (V, E)$ dont les arcs ou les noeuds sont munis d'une ou plusieurs valeurs.

Dans la suite, nous utiliserons les notations et notions suivantes :

- $\forall i \in V, b_i < 0$ est une demande et $b_i > 0$ est une offre.
- $\forall (i,j) \in E, c_{i,j}$ est un coût unitaire pour traverser l'arc ij $(< C)$.
- $\forall (i,j) \in E, u_{i,j}$ est une capacité supérieure pour l'arc $ij (< U)$.
- $\forall (i,j) \in E, l_{i,j}$ est une capacité inférieure pour l'arc $ij (< L)$.

Complexité d'un algorithme

La complexité temps d'un algorithme est une fonction du nombre d'opérations *élémentaires* effectuées dans l'algorithme.
On utilise la notation O pour donner un Ordre de grandeur de la complexité.
Un algorithme a une complexité $O(f(n))$, si $\exists c$ et n_0 tels que le temps pris par l'algorithme dans le pire des cas est au plus $c.f(n)$ pour $n \geq n_0$.

- Problème du plus court chemin : Il s'agit de trouver la façon la plus économique (temps, distance, difficulté,...) de passer d'un nœud d'un réseau à un autre.
- Problème du flot maximum : Il s'agit d'envoyer la plus grande valeur de flot (quantité, volume, usagers,...) à travers un réseau entre deux points en tenant compte d'une capacité limitative.
- Problème de flot de coût minimum : Il s'agit d'envoyer du flot à travers un réseau entre deux points en tenant compte d'une capacité limitative et en minimisant le coût global de circulation.

Résolution du problème de plus court chemin

1. Données :
   $G = (V, E)$ orienté ; Deux nœuds s origine et p destination ; un coût unitaire c_{ij} par arc ij
2. Formulation linéaire
   - Les variables
     $$x_{ij} = \begin{cases} 1 & \text{si l'arc } ij \text{ est choisi} \\ 0 & \text{sinon} \end{cases}$$
   - Les contraintes
     $$\sum_{j \in V} x_{ij} - \sum_{j \in V} x_{ji} = \begin{cases} +1 & \text{si } i = s \text{ (origine)} \\ -1 & \text{si } i = p \text{ (destination)} \\ 0 & \text{sinon} \end{cases}$$
   - L'objectif
     $$\text{Min} \sum_{ij \in E} c_{ij} x_{ij}$$
3. Remarques
   - L'objectif est infini en présence de circuits négatifs.

Les algorithmes (Voir TD)

- Bellman : Graphe sans circuits
- Djikstra : Poids positifs
- Bellman-Ford : Général. Detecte les circuits négatifs.

Résolution du problème du flot maximum

1. Données :
   $G = (V, E)$ orienté ; Deux nœuds s origine et p destination ;
   Une capacité $0 \leq u_{ij} \leq U$ par arc ij.
2. Formulation linéaire
   - Les variables
     $x_{ij} = \text{ Valeur du flot sur l'arc } ij$
   - Les contraintes
     $$\sum_{j \in V} x_{ij} - \sum_{j \in V} x_{ji} = \begin{cases} +v & \text{si } i = s \text{ (origine)} \\ -v & \text{si } i = p \text{ (destination)} \\ 0 & \text{sinon} \end{cases}$$
     $0 \leq x_{ij} \leq u_{ij}, \quad \forall ij \in E$
   - L'objectif : Max v
3. Algorithmes : Ford & Fulkerson, Karp, Push & relabel.

1. Données :
   $G = (V, E)$ orienté ; Nœuds origines et destinations ; Demandes et Offres : $b_i > 0$ aux sources, $b_i < 0$ aux destinations et nulle aux intermédiaires. Une capacité inférieure $0 \leq l_{ij} \leq L$ par arc ij et une capacité supérieure $0 \leq u_{ij} \leq U$ par arc ij. Un coût unitaire c_{ij} par arc ij
2. Formulation linéaire
   - Les variables
     $x_{ij} = \text{ Valeur du flot sur l'arc } ij$
   - Les contraintes
     $$\begin{array}{rcll} \sum_{j \in V} x_{ij} - \sum_{j \in V} x_{ji} & = & b_i, & \forall ij \in E \\ l_{ij} & \leq x_{ij} \leq & u_{ij}, & \forall ij \in E \end{array}$$
   - L'objectif :
     $$\text{Min} \sum_{ij \in E} c_{ij} x_{ij}$$

- Association Roadef *Le livre blanc de la recherche opérationnelle*. http ://www.roadef.org/ (2011)
- David L. Applegate, Robert E. Bixby, Vasek Chvátal, William J. Cook : *The Traveling Salesman Problem* . Chap 1.(2006)
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein : *Introduction to Algorithms*. MIT press (1990)
- David Simplot-Ryl, Eric Fleury : *Réseaux de capteurs - Théorie et modélisation*. Ed. Lavoisier (2009).