Le problème de flot de coût minimum en variables entières

Objectifs pédagogiques :

- Travailler sur un problème à l'intersection des Unités d'Enseignements "Integer Programming" et 'Graphs and Networks"
- Mobiliser des connaissances acquises dans ces deux cours
- (Mieux) comprendre un problème fondamental
- Découvrir les algorithmes de ranking

Présentation :

Le problème de flot de coût minimum en variables entières (également appelé problème de transbordement) est un des problèmes fondamentaux de l'optimisation combinatoire. Ce problème contient en effet comme cas particuliers d'autres problèmes d'optimisation combinatoire, comme le problème de plus court chemin dans un graphe ou le problème d'affectation linéaire, et se retrouve également fréquemment en tant que sous-problème d'autres problèmes plus difficiles. Le but de ce problème consiste à trouver le moyen le moins cher afin d'acheminer des quantités de marchandises depuis des sources vers des destinations spécifiées en utilisant un réseau de transport donné. Ce problème est généralement formulé en s'appuyant simultanément sur le formalisme de la théorie des graphes, et celui de la Programmation Linéaire.

Soit $G = (V, A)$ un graphe orienté dont $V = \{1, \ldots, n\}$ est l'ensemble des sommets, et A l'ensemble des arcs. Pour chaque sommet $i \in V$, on associe un entier b_i qui peut être

- Positif : le sommet i est alors une source à partir de laquelle des marchandises doivent être transportées,
- Négatif : le sommet i est alors une destination à laquelle doivent être livrées des marchandises,
- Nul : le sommet i sert alors uniquement à faire transiter des marchandises (i.e. autant de marchandises entrent et sortent de ce sommet).

Pour chaque arc $(i, j) \in A$, des bornes supérieure $u_{ij} \in \mathbb{N}$ et inférieure $l_{ij} \in \mathbb{N}$ sur le nombre d'unités de flot transitant par cet arc doivent également être respectées. Enfin, on note c_{ij} le coût par unité de flot transitant par l'arc (i, j). Le but du problème est bien entendu de minimiser le coût total, défini par la somme des coûts sur chaque arc tout en respectant l'ensemble des contraintes.

On associe à chaque arc $(i, j) \in A$ une variable de décision $x_{ij} \in \mathbb{N}$ représentant le nombre d'unités de flot transitant par cet arc. On note pour chaque sommet $i \in V$, $Suc(i) := \{j \in V | (i, j) \in A\}$ l'ensemble des successeurs du sommet i dans le graphe, et $Pred(i) := \{j \in V | (j, i) \in A\}$ l'ensemble de ses prédécesseurs. Le flot de coût minimum peut alors être posé sous la forme d'un Programme Linéaire en variables entières.

$$
\begin{array}{lrcll}
\min\ z(x) & = \displaystyle\sum_{i \in V} \sum_{j \in Suc(i)} c_{ij} x_{ij} & & & \\
\text{subject to} & \displaystyle\sum_{j \in Suc(i)} x_{ij} - \sum_{j \in Pred(i)} x_{ji} & = & b_i & \forall i \in V, \\
 & x_{ij} & \geq & l_{ij} & \forall (i, j) \in A, \\
 & x_{ij} & \leq & u_{ij} & \forall (i, j) \in A, \\
 & x_{ij} & \in & \mathbb{N} & \forall (i, j) \in A.
\end{array}
$$

Comme la matrice des contraintes de ce problème vérifie la propriété de totale unimodularité et que les membres de droite des contraintes sont entiers, ce problème est équivalent à sa relaxation continue. On peut donc simplement le résoudre en utilisant l'algorithme du simplexe. Cependant comme pour le problème de plus court chemin ou le problème d'affectation, une telle solution ne sera pas satisfaisante. Il y a en effet un grand nombre d'algorithmes qui ont été développés en exploitant spécifiquement la structure de ce problème, et qui se montrent bien plus efficaces [1]. Nous nous intéresserons en particulier à deux algorithmes qui apparaîtront comme des extensions/cas particuliers naturels de deux algorithmes étudiées au premier semestre : l'algorithme simplexe des réseaux, l'algorithme des plus courts chemins successifs.

L'algorithme simplexe des réseaux est une version spécialisée de l'algorithme du simplexe [2]. Tout comme la version classique de l'algorithme du simplexe, cet algorithme consiste à déterminer une première solution de base admissible, puis à effectuer des changements de base jusqu'à obtenir une solution optimale. Les principales différences sont ici une modification de la façon de représenter une base, et d'effectuer un changement de base. En effet, une base sera ici représentée par un arbre recouvrant le graphe G dans sa version non-orientée, et un changement de base consistera à passer d'un arbre couvrant à un autre dont seule une arête est différente. Il sera particulièrement intéressant de comparer les performances de cet algorithme à la version classique de l'algorithme du simplexe.

L'algorithme des plus courts chemins successifs est un algorithme de type primal-dual. À chaque itération de cet algorithme, une nouvelle solution est obtenue à partir de la solution de l'itération précédente, tout en s'assurant de vérifier une condition d'optimalité. Cette condition est définie à l'aide des coûts réduits sur les différents arcs du graphe, eux-mêmes définis à l'aide des valeurs des variables duales associées aux sommets de ce graphe. Une solution optimale est donc obtenue dès la première solution admissible connue. Cet algorithme se montre particulièrement efficace pour le problème d'affectation. L'exécution de l'algorithme des plus courts chemins successifs sur une instance de ce dernier problème est alors très similaire à la méthode hongroise vue lors du premier semestre, et sera d'ailleurs particulièrement intéressante à observer.

Comme le but d'un TER est de s'initier à la recherche (ce qui implique d'être curieux), d'autres algorithmes de la littérature [1] pourront être considérés si cela motive les étudiants choisissant ce projet.

Dans une deuxième partie, nous pourrons également considérer les algorithmes de *ranking* proposés pour ce problème. Un algorithme de ranking consiste à générer les k-meilleures solutions d'un problème. C'est-à-dire de générer la solution optimale, puis la seconde meilleure,... puis la k^{eme} meilleure. Autrement dit, de générer les solutions $x^1, x^2, \ldots, x^k$ telles que $z(x^1) \leq z(x^2) \leq \ldots \leq z(x^k) \leq z(x)$ pour toute solution admissible $x \notin \{x^1, \ldots, x^k\}$. Ce type de méthode est parfois utilisé pour résoudre des problèmes incluant des contraintes additionnelles. Des algorithmes de ranking ont été proposés spécifiquement pour le problème de flot de coût minimum en variables entières [3, 4].

Mots-clés : Problème de flot de coût minimum, algorithme du simplexe, plus court chemin, algorithme primal-dual, algorithme de ranking.

Encadrant : Anthony Przybylski

Nombre d'étudiants : 2 à 4

Références

[1] R.K. Ahuja, T.L. Maganti, and J.B. Orlin. *Network Flows - Theory, Algorithms, and Applications.* Prentice-Hall, 1993.

[2] V. Chvatal. *Linear Programming.* Series of books in the mathematical sciences. W. H. Freeman, 1983.

[3] H.W. Hamacher. K best network flows. *Annals of Operations Research*, 57 :65–72, 1995.

[4] Antonio Sede no Noda and Juan José Espino-Martin. On the k best integer network flow. *Computers & Operations Research*, 40 :616–626, 2014.