1.5em 0pt

Flot Maximum

Thomas Suau, Noé Bourgeois

March 2023

Table des matières

1 Introduction

Le problème de flot maximum est un problème d'optimisation en théorie des graphes, qui consiste à trouver le flot maximum qu'il est possible d'envoyer d'un nœud source à un nœud puits dans un graphe pondéré orienté, où les arcs ont des capacités maximales.

Dans ce rapport, nous allons présenter deux méthodes pour résoudre le problème de flot maximum. La première méthode est la résolution à l'aide d'un programme linéaire avec le solver `glpk`, et la seconde méthode est la méthode des chemins augmentants implémentée en python3.

2 Programme linéaire

2.1 Formulation

— la formulation utilisée en dérivant les différentes notations comme en TP. Expliquez en détail pourquoi les variables, contraintes et fonction objectif du programme modélisent entièrement le problème, tâchez d'avoir une formulation de qualité.

Le problème de flot maximum peut être formulé comme un programme linéaire. Soit G=(V, A) un graphe orienté pondéré avec un nœud source s et un nœud puits t. Soit f_{ij} la quantité de flux circulant sur l'arc $(i, j) \in A$, et c_{ij} la capacité maximale de cet arc. Le problème de flot maximum peut être formulé comme suit :

Maximiser :

$\sum_{(i,j)\in A} f_{ij}$

sous les contraintes :

$$\sum_{(i,j)\in A} f_{ij} - \sum_{(j,i)\in A} f_{ji} = \begin{cases} -q & \text{si } i = s \ q \\ \text{si } i = t \ 0 & \text{sinon} \end{cases}$$

où q est le flot sortant de la source s

$f_{ij} \leq c_{ij}$ pour tout $(i, j) \in A$

$f_{ij} \geq 0$ pour tout $(i, j) \in A$

Les variables f_{ij} représentent le flot circulant sur l'arc (i, j) et c_{ij} est la capacité maximale de l'arc (i, j). Les contraintes expriment la conservation de flux en chaque nœud, la capacité maximale de chaque arc et le fait que le flot doit être positif.

3 Chemins augmentants

— une description de la m´ethode des chemins augmentant impl´ement´ee,

4 Résultats

système d'exploitation de la machine sur laquelle nous fait nos tests : Windows 11

4.1 Programme Linéaire

Méthode	Nom de l'instance	Temps de traitement (s)
Linéaire	2	0.2
Chemin augmentant	5	0.5
Linéaire	100	1

4.2 Chemin augmentant

Nom de l'instance	Temps de traitement (s)
2	0.2
5	0.5
100	1

4.3 Comparaison

Nom de l'instance	Temps de traitement (s) Linéaire	Temps de traitement (s) Augmentant
2	0.2	0.3
5	0.5	0.6
Linéaire	100	1

— une analyse des r´esultats de la r´esolution des diff´erentes instances. Cette analyse peut comparer les temps de r´esolution pour des instances de taille croissante

pour les deux m´ethodes de r´esolution.

Vous pouvez analyser les temps de r´esolution moyens ainsi que

l'´ecart-type pour les diff´erentes tailles d'instances. T^achez d'interpr´eter le pourquoi des r´esultats et

identifiez si une m´ethode semble plus appropri´ee.

Un nombre exhaustif d'instances est fournies. Le rapport ne doit pas contenir les r´esultats de toutes les instances. S´electionnez les instances afin d'^etre le plus complet dans vos r´esultats.

4.4 Conclusions

5 Ressources

(cf. énoncé) Rédaction scientifique :

http://informatique.umons.ac.be/algo/redacSci.pdf.

Ressources bibliographiques :

https://www.bibtex.com/.

L'écriture du code a été accélérée à l'aide du plugin "Github Copilot" https://copilot.github.com/