Le problème de sommets vitaux pour le plus court chemin

Youcef Magnouche[1], Sébastien Martin[1]

Huawei technologies France, 18 Quai du Point du Jour, 92100 Boulogne-Billancourt

{youcef.magnouche}@huawei.com, {sebastien.martin}@huawei.com

Mots-clés : *Bloqueur, Le plus court chemin, Complexité algorithmique, Programmation linéaire en nombres entiers, Polyèdre, Branch-and-Cut.*

1 Introduction

Dans de nombreuses applications, l'identification des sommets les plus critiques est une nécessité pour assurer la sécurité. En effet, la capacité de survie des réseaux (réseaux de télécommunication ou de transport) est une priorité pour prévenir les pannes. L'un des problèmes les plus importants est de réassurer une connexion (chemin) entre s et t respectant certaines contraintes après la suppression de certains nœuds du réseau. En informatique théorique, ces types de problèmes sont appelés *Problèmes d'interdiction de réseau* ou *Problèmes de bloqueur de réseau* [1], [2], [3].

Dans cet article, nous introduisons le problème de sommets vitaux pour le plus court chemin entre deux sommets s et t (MVVSP) défini comme suit : étant donné un digraphe $D = (V \cup \{s,t\}, A)$ de longueur non-négative $l(a)$ associé à chaque arc $a \in A$ et à un seuil $d > 0$, un ensemble de sommets est dit vital si sa suppression garantit qu'il n'existe aucun chemin entre s et t de longueur inférieur ou égal à d. L'objectif est de trouver le plus petit ensemble vital.

Nous supposons qu'il n'y a pas d'arc direct de s à t. Sinon, le problème est trivial. Par exemple, dans la Figure 1, la meilleure solution pour le MVVSP lorsque $d = 3$ est $\{v_5\}$. Celui-ci est vital dans cet exemple car sa suppression engendre une perte de connexion entre s et t.

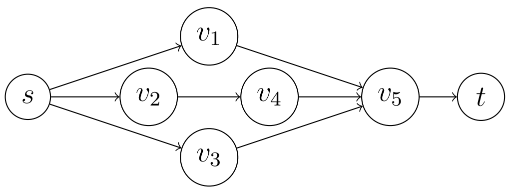

FIG. 1 – Exemple d'une solution du MVVSP

2 Programme linéaire en nombres entiers

Dans cette section, nous proposons un programme linéaire en nombres entiers pour modéliser le MVVSP. Soit $x \in \{0,1\}^{|V|}$ définit par

$$x_v = \begin{cases} 1 & \text{si } v \text{ est un sommet vital,} \\ 0 & \text{sinon,} \end{cases} \qquad \forall v \in V.$$

Le problème MVVSP est equivalent au programme suivant :

$$\min \sum_{v \in V} x_v$$

$$\sum_{v \in P} x_v \geq 1, \quad \text{pour tout } P \in \mathcal{P}, \tag{1}$$

$$x_v \in \{0,1\}, \quad \text{pour tout } v \in V. \tag{2}$$

où $\mathcal{P}$ est l'ensemble de tous les chemins entre s et t de longueur inférieure ou égale à d. Les inégalités (1) permettent d'interdire au moins un nœud dans tous les chemins de $\mathcal{P}$. Les inégalités (1) sont en nombre exponentiel et le problème de séparation associé est NP-Difficile.

3 Compléxité et Analyse polyédrale

Soit $P(D,d) = conv(x \in \{0,1\}^{|V|} | \, x \; satisfies \; (1))$ le polytope des ensembles de sommets vitaux dans $D = (V, A)$ pour une longueur d.

Théorème 1. *Le problème MVVSP est NP-Difficile.*

Théorème 2. *$P(D,d)$ est de pleine dimension.*

Théorème 3. *Pour un chemin $P \in \mathcal{P}$, l'inégalité (1) définit une facette pour $P(D,d)$ si et seulement si P est minimale.*

Théorème 4. *Le polytope donné par (1) et les inégalités triviales est entier.*

4 Résultats numériques

Nous avons développé un algorithme de Branch-and-Cut pour résoudre le MVVSP. Comme indiqué dans la section 2, le programme linéaire en nombres entiers a un nombre exponentiel d'inégalités (1). Dans notre section Branch-and-Cut, nous ne séparons que les solutions entières en séparant les inégalités (1) à l'aide de l'algorithme de Dijkstra pour résoudre le problème le plus court chemin entre s et t.

TAB. 1 – Average of CPU time.

$\|V\|$	Densité %	d	Noeuds Branchement	CPU	#cuts	Nombre sommets vitaux
16000	2	3	1	9.61	7	7
16000	2	4	1	1136.42	857	334
19000	2	3	1	16.35	7	7
19000	2	4	1	2482.30	1110	360
8000	10	3	1	138.77	83	83
8000	10	4	1	Time Limit	2759	780

5 Conclusion

Dans cet article, nous avons considéré le problème de sommets vitaux pour le plus court chemin. Nous avons étudié la complexité de ce problème et proposé un modèle sous forme d'un programme linéaire en nombres entiers. Nous avons montré que le polytope associé est un entier et développé un algorithme efficace de type Branch-and-Cut pour le résoudre.

Références

[1] E. Israeli and P. K. Wood, Shortest-Path Network Interdiction, NETWORKS, Vol. 40(2), 97–111 2002.

[2] L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, V. Gurvich, G. Rudolf, and J. Zhao On Short Paths Interdiction Problems : Total and Node-Wise Limited Interdiction, Theory Comput Syst, vol.43, 204–233, 2008.

[3] Y. Song and S. Shen, Risk Averse Shortest Path Interdiction, INFORMS Journal on Computing, Vol. 23(3), 527-539, 2016.

[4] Y. Magnouche and S. Martin Most vital vertices for the shortest s–t path problem : complexity and Branch-and-Cut algorithm, Optimization Letters, 2020