Résumé de l'article scientifique de SOD321

David DAHAN MONSONEGO

September 26, 2022

1 Introduction

On considère un graphe $G = (V, A)$ où V représente l'ensemble des sommets et A celui des arrêtes et le coût c_{ij} de l'arrête $(i, j) \in A$. On s'intéresse au problème du plus court chemin entre deux noeuds s et t.
On assume souvent, de manière implicite, qu'un tel chemin ne devrait passer au plus une fois par le même sommet. Lorsque les coûts c_{ij} ne forment pas de cycles négatifs dans G (i.e $c_{ij} > 0$) alors le problème du plus court chemin peut être résolu par des algorithmes efficaces comme celui de Bellman–Ford ou de Dijikstra. Cependant si des cycles négatifs apparaissent il est primordial pour la résolution du problème de ne pas y rester bloquer. C'est ce qu'on appelle communément le problème du plus court chemin élémentaire (**ESPP** en anglais).

L'ESPP est donc réduit à un problème de chemin hamiltonien, c'est à dire que chaque noeud doit être visité une seule fois uniquement. La résolution de ce dernier s'inscrit dans le cadre de problème similaire notamment les variantes du problème du voyageur de commerce.Bien qu'il s'agisse d'un problème intéressant en soi, l'ESPP se pose également dans les sous-problèmes de tarification des algorithmes de branch-and-price. Souvent, la phase de tarification dans les problèmes de routage de véhicules implique des variantes contraintes en ressources du problème du plus court chemin élémentaire. Ces problèmes sont généralement résolus avec des algorithmes où on assigne des étiquettes à chaque noeuds et basés sur la programmation dynamique rapide. Il est possible d'adapter ce type d'approche à l'ESPP sans contrainte en considérant une ressource artificielle pour chaque nœud qui est consommée lorsque le nœud est visité, et en imposant de ne pas utiliser plus d'une unité de chaque ressource. Cependant, il s'agit d'une contrainte assez faible, dans le sens où elle autorise des chemins très longs, de sorte que les approches basées sur des algorithmes d'étiquettage deviennent très coûteux en temps.

Cette limitation est déjà mise en évidence pour l'ESPP avec une contrainte de capacité par Jepsen, Petersen et Spoorendonk (2008), qui proposent un algorithme branch-and-cut nettement plus performant que les algorithmes d'étiquettage. Dans le contexte d'un algorithme de branch-and-price, où l'ESPP est résolu à plusieurs reprises dans la phase de tarification, une autre caractéristique souhaitable d'une approche de programmation en nombres entiers est sa flexibilité, qui permet d'incorporer facilement des décisions de branchement générales ou des inégalités valides. C'est ce qui motive l'étude des techniques de programmation en nombres entiers pour l'ESPP. De plus peu de travaux antérieurs ont été publiés sur les approches de programmation en nombres entiers spécialement conçues pour l'ESPP.

2 Formulation des contraintes pour la programmation en nombres entiers

On note n et m le cardinal de V et A respectivement. Un chemin est une séquence de noeux $v_1 \dots v_k$. Il est dit élémentaire si les noeuds qui le compose apparaissent qu'une seule fois. Un cycle est chemin avec $v_1 = v_k$. On note $\delta^+(i)$ et $\delta^-(i)$ l'ensemble des arcs entrants et sortant du noeud i, par $\delta^+(S)$ et $\delta^-(S)$ les arcs sortants et entrants de l'ensemble $S \subseteq V$ et par $A(S)$ l'ensemble des arcs dont les deux extrémités sont dans $S \subseteq V$. On définit aussi $V_i := V \setminus \{i\}$, $V_{ij} := V \setminus \{i, j\}$ et pour tout ensemble d'arcs $B \subseteq A$, on définit $\chi(B) := \sum_{b \in B} x_b$. On assume également, sans perte de généralité, que $|\delta^-(s)| = |\delta^+(t)| = 0$. Une formulation standard de programmation en nombres entiers pour

déterminer un chemin le plus court du noeud s au noeud t est $\min \sum_{(i,j)\in A} c_{ij}x_{ij}$ où $c_{ij} \in \mathbb{R}$ et $x_{ij} \in \{0,1\}$ représente respectivement le coût de l'arc (i,j) et la variable qui prend la valeur 1 si l'arc (i,j) appartient au chemin. Les contraintes sont les suivantes :

- Conservation du flow : $\sum_{(i,j)\in\delta^+(i)} x_{ij} - \sum_{(j,i)\in\delta^-(i)} x_{ji} = \mathbb{1}_{\{i=s\}} - \mathbb{1}_{\{i=t\}} \quad \forall i \in V$
- Le degré sortant de chaque nœud est au plus un : $\sum_{(i,j)\in\delta^+(i)} x_{ij} \leq 1$
- Les variables sont binaires : $x_{ij} \in \{0,1\} \forall (i,j) \in A$

Lorsque les coûts c_{ij} induisent des cycles négatifs sur G, c'est-à-dire qu'il existe un sous-tour tel que le coût total de ses arcs soit négatif, ce système d'inégalités n'est pas suffisante pour garantir l'élémentarité du chemin. Ainsi, des contraintes supplémentaires (et éventuellement des variables) sont nécessaires pour empêcher les sous-tours :

- **Les contraintes de Dantzig–Fulkerson–Johnson (DFJ)** : dans chaque sous-ensemble S, les sous-tours sont empêchés en s'assurant que le nombre d'arcs dans S qui sont sélectionnés est inférieur au nombre de noeuds dans S. Cette formulation inclut $O(m)$ variables et $O(2^n)$ contraintes.
- **Generalized cutset inequalities (GCS)** : pour chaque sous-ensemble S, le nombre d'arcs sélectionnés quittant S n'est pas inférieur au nombre d'arcs sélectionnés sortant de n'importe quel noeud de S. Le nombre de variables dans la formulation est $O(m)$, tandis que le nombre de contraintes est $O(n2^n)$.
- **Formulation séquentielle (MTZ)** : introduire, pour chaque nœud, une variable auxiliaire qui peut être vue comme la position du nœud le long du chemin et une contrainte pour chaque arc. Elle ne nécessite que $O(m)$ contraintes supplémentaires et $O(n)$ variables auxiliaires.
- **Formulation basée sur la reformulation-linéarisation (RLT)** : pour un arc donné $(i,j) \in A$, on introduit les variables α_{ij} et β_{ij} que l'on interprète comme la position des noeuds i et j le long du chemin. Cette formulation étendue nécessite $O(m)$ contraintes et $O(m)$ variables auxiliaires.
- **Formulation du flow unique (SF)** : on introduit un flux auxiliaire q à délivrer aux nœuds appartenant au chemin de s à t. Les contraintes assurent que le flux auxiliaire partant du noeud s a une valeur égale au nombre de noeuds atteints par le chemin de s à t et que l'équilibre du flux auxiliaire sur chaque noeud vaut 1 si le noeud se trouve dans le chemin de s à t et 0 sinon. Cette formulation étendue nécessite $O(m)$ contraintes et $O(m)$ variables auxiliaires.
- **Formulation multiflow (MCF)** : Une extension de la formulation à flux unique est obtenue en désagrégeant le flux auxiliaire en $n-1$ flux unitaires. Les sous-tours sont empêchés en appliquant une unité d'un flux auxiliaire distinct de s à chaque noeud appartenant au chemin de s à t. La formulation inclut $O(nm)$ variables et contraintes supplémentaires.

3 Résultats et conclusion

Il est important de comprendre combien sont efficaces les formulations d'un point de vue compléxité. A cet égard, plusieurs expériences approfondies ont été menées. Il s'avère que la séparation dynamique des contraintes d'élimination des sous-tours soit la meilleur option lorsqu'on considère l'ESPP comme un problème de programmation en nombres entiers. L'approche GCS avec la procédure de séparation forte basée sur les composants est capable de résoudre des instances de petite taille en quelques secondes et des instances de taille moyenne, jusqu'à 1000 noeuds en une minute, mais elle n'est toujours pas suffisante pour résoudre des problèmes à grande échelle. Lorsque la mise en oeuvre d'une procédure de séparation n'est pas possible ou pratique, la formulation SF est probablement le meilleur choix. Il semble probable que le développement de bonnes heuristiques primales et d'inégalités valides fortes supplémentaires, pourrait encore accélérer les temps de calcul et permettre la résolution d'instances de plus grande taille. Il pourrait également être utile d'emprunter des techniques aux approches typiques utilisées pour les variantes à ressources limitées du ESPP, comme une phase de prétraitement dans le but de réduire l'espace de recherche.