Formulation relaxée de la séparation équilibrée d'un graphe

Marc Sarfati[1,2], Marc Queudot[2], Catherine Mancel[3], Marie-Jean Meurs[2]

[1] Ecole Polytechnique, Palaiseau, France
[2] Université du Québec à Montréal, Montréal, QC, Canada
[3] Ecole Nationale de l'Aviation Civile, Toulouse, France

Mots-clés : *séparation de graphe, programmation linéaire, vertex separator problem, vsp*

Dans ce travail, nous nous intéressons à la formulation de variantes du problème du séparateur dans un graphe (*VSP*, pour Vertex Separator Problem), permettant de fournir des méthodes alternatives aux méthodes classiques de clustering.

Etant donné $G = (V, E)$ un graphe connexe non-dirigé (dont les sommets peuvent être pondérés), le but du VSP est de trouver un ensemble de sommets de poids minimal dont la suppression déconnecte le graphe. En d'autres termes, il s'agit de trouver une partition (A, B, C) de V telle qu'aucune arête ne lie A et B et telle que C est minimal. C est appelé un *séparateur* du graphe G. Le VSP est un problème NP-complet. Dans [1], Kanevsky propose un algorithme permettant de trouver tous les séparateurs de taille minimale d'un graphe en se basant sur un algorithme de flot maximal. Cette solution présente deux inconvénients : i) elle ne permet pas de prendre en compte des poids sur les sommets et ii) la partition du graphe peut être totalement déséquilibrée. En effet, la Figure 1 présente le compromis entre la recherche d'un séparateur minimal et d'une partition équilibrée. On peut voir qu'un séparateur minimal du graphe est le singleton du sommet rouge, mais trouver le séparateur en bleu serait plus pertinent pour des applications au clustering. En effet la partition engendrée est bien plus équilibrée en utilisant le séparateur bleu. En 2011, pour faire suite aux travaux de Balas

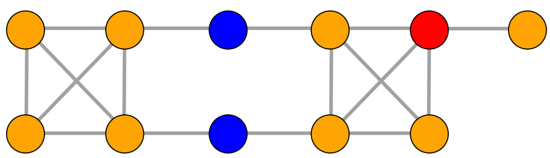

FIG. 1 – Compromis entre séparateur minimal et partition équilibrée

et De Souza [2, 3], Didi Biha et Meurs présentent dans [4] une nouvelle modélisation du VSP utilisant la programmation linéaire. Cette modélisation (sur laquelle se base cet article) permet de prendre en compte des poids sur les sommets et permet d'obtenir des partitions équilibrées. L'idée est de maximiser $f : (A, B, C) \longmapsto |A| + |B|$ sous contrainte que $E \cap (A \times B) = \emptyset$ et $|A|, |B| \leq \beta(|V|)$ où $\beta(|V|)$ est une constante arbitraire qui majore les tailles des piliers A et B afin d'équilibrer la partition calculée. Benlic et Hao présentent une solution approximée du VSP dans [5]. Le graphe est séparé en 2 sous-ensembles de sommets A et B, puis pour chaque arête liant A et B, l'une des deux extrémités est déplacée dans un autre ensemble C. Cela permet d'obtenir un premier séparateur a priori non optimal. Afin d'équilibrer le graphe, on retrouve la contrainte d'équilibre $|A|, |B| \leq \beta(|V|)$. La recherche d'un optimum local se fait ensuite en déplaçant aléatoirement un sommet du séparateur dans un des piliers A ou B, puis en déplaçant ses voisins dans le séparateur si nécessaire pour assurer la non-connectivité entre A et B.

Ici, nous introduisons une formulation relaxée du problème permettant de calculer un séparateur de poids faible engendrant une partition équilibrée. En effet, nous modélisons le problème sous forme de programme linéaire en nombre entiers (en gardant les notations de [4]) qui relâche les contraintes de majoration des tailles des piliers, à savoir $|A|, |B| \leq \beta(|V|)$.

Le programme linéaire prend alors la forme :

$$max(c(A)+c(B))-\lambda(c(A)-c(B)) \tag{1}$$

sous contrainte de (i) (A,B,C) est une partition de V, (ii) $c(A)\geq c(B)$, (iii) $E\cap(A\times B)=\emptyset$, où c est la fonction qui associe un ensemble de sommets à son poids et où $\lambda\in[0,1]$ est un hyper-paramètre permettant de calibrer le compromis entre l'obtention d'un séparateur de faible poids et l'équilibre entre les piliers A et B.

Nous proposons ensuite une nouvelle version du VSP se basant sur la formulation (1), qui permet de prendre aussi en compte des distances entre les sommets, afin de trouver un séparateur de faible taille engendrant une partition équilibrée et minimisant les distances intra-clusters. Pour cela, définissons $\mathbf{x},\mathbf{y}\in\{0,1\}^{|V|}$ les indicatrices d'appartenance à A et B (*i.e.* pour $v\in V$, $x_v=1$ si $v\in A$ et $y_v=1$ si $v\in B$) et $\mathbf{c}\in(\mathbb{R}^+)^{|V|}$ le vecteur des poids de V. (1) se reformule alors de la manière suivante :

$$max^\top\mathbf{c}(\mathbf{x}\ +\ \mathbf{y})-\lambda^\top\mathbf{c}(\mathbf{x}\ -\ \mathbf{y}) \tag{2}$$

Le programme d'optimisation prenant en compte les distances intra-clusters est le suivant :

$$max^\top\mathbf{c}(\mathbf{x}\ +\ \mathbf{y})-\lambda^\top\mathbf{c}(\mathbf{x}\ -\ \mathbf{y})-\mu\sum_{u,v\in V}d(u,v)(x_ux_v+y_uy_v) \tag{3}$$

où l'hyper-paramètre $\mu\in\mathbb{R}^+$ permet de choisir l'importance des distances intra-cluster. Ce problème fait maintenant intervenir des termes quadratiques donc est plus difficilement calculable par des solveurs classiques. Nous présentons donc une nouvelle famille de variables contraintes permettant de linéariser l'intégration des distances au programme linéaire en nombre entiers.

On introduit $(z_{uv})_{u,v\in V}$ tel que pour $u,v\in V$ $z_{uv}=1$ si u,v appartiennent au même pilier et $z_{uv}=0$ sinon. Le programme d'optimisation prend alors la forme

$$max^\top\mathbf{c}(\mathbf{x}\ +\ \mathbf{y})-\lambda^\top\mathbf{c}(\mathbf{x}\ -\ \mathbf{y})-\mu\sum_{u,v\in V}d(u,v)z_{uv} \tag{4}$$

sous contrainte de (i)-(iii) et $\forall u,v\in V, z_{uv}=x_ux_v+y_uy_v$. Ces dernières contraintes sont quadratiques mais peuvent être linéarisées sous la forme : (iv) $\forall u,v\in V, z_{uv}\geq x_u+x_v-1$ (v) $\forall u,v\in V, z_{uv}\geq y_u+y_v-1$ (vi) $\forall u,v\in V, z_{uv}\in\{0,1\}$.

Nous avons présenté une nouvelle modélisation relaxée du VSP permettant de séparer un graphe de manière équilibrée où le poids du séparateur est faible et minimisant les distances intra-clusters. Les deux reformulations du VSP ainsi obtenues peuvent être utilisées à des fins de clustering de graphe puisqu'elles permettent de séparer un graphe en composantes connexes. La formulation (1) ne requiert pas la définition d'une distance entre les sommets du graphe, tandis que le seconde permet de minimiser des distances intra-clusters. Des premiers tests menés sur différents ensembles de données montrent que la recherche de séparateurs avec la formulation (1) permet de détecter des sous-groupes sémantiquement liés au sein d'un graphe.

Références

[1] Arkady Kanevsky Finding All Minimum Size Separating Vertex Sets in a Graph Coordinated Science Laboratory Report no. ACT-93 (UJLU-ENG 88-2233), 1988

[2] Balas, Egon and de Souza, Cid The vertex separator problem : a polyhedral investigation Mathematical Programming, 103, 3, 583–608, 2005

[3] de Souza, Cid and Balas, Egon The vertex separator problem : algorithms and computations, Mathematical Programming, 103, 3, 609–631, 2005

[4] Mohamed Didi Biha and Marie-Jean Meurs An Exact Algorithm for Solving the Vertex Separator Problem Journal of Global Optimization, 49, 3, 425–434, 2011

[5] Una Benlic and Jin-Kao Hao Breakout Local Search for the Vertex Separator Problem. International Joint Conference on Artificial Intelligence (IJCAI), 2013