Formulations linéaires en nombres entiers pour des problèmes d'isomorphisme exact et inexact

P. Le Bodic[1,2], S. Adam[2], P. Héroux[2], A. Knippel[1] et Y. Lecourtier[2]

1. LMI/INSA de Rouen, Département Génie Mathématique
{pierre.le-bodic, arnaud.knippel}@insa-rouen.fr

2. Laboratoire d'Informatique, Traitement de l'Information et des Systèmes (LITIS), Université de Rouen, Saint-Etienne-du-Rouvray
{sebastien.adam, pierre.heroux, yves.lecourtier}@univ-rouen.fr

Mots-clefs : isomorphisme de sous-graphe, programmation linéaire 0-1

1 Introduction

De nombreuses applications requièrent l'évaluation d'une mesure de similarité entre objets. En particulier, ces objets peuvent être des images, des textes, des molécules. Selon les cas, on peut vouloir les reconnaître, les classifier, ou effectuer des recherches parmi ceux-ci. Les applications sont par exemple la reconnaissance d'écriture manuscrite, l'analyse d'images et de vidéos, l'identification biométrique.

Les graphes sont une manière puissante de représenter ces objets. Dans cette représentation, les sommets sont généralement eux-même des objets, tandis que les arcs ou les arêtes correspondent à des relations entre ceux-ci. Ainsi, les algorithmes identifiant des isomorphismes entre graphes permettent de mesurer la similarité de deux objets représentés par des graphes.

On distingue plusieurs problèmes d'isomorphisme de graphe. Dans tous les cas, les graphes peuvent être orientés, leurs arcs ou sommets étiquetés. On peut vouloir identifier un isomorphisme entre deux graphes, entre un graphe et un sous-graphe, ou trouver le plus grand sous-graphe commun à deux graphes. Pour une présentation des techniques et des applications relatives à l'isomorphisme de graphe, nous conseillons [1] au lecteur. Des formulations mathématiques ont déjà été proposées pour des problèmes proches : un programme linéaire 0-1 a par exemple été utilisé pour le calcul d'une distance d'édition entre deux graphes [2]. Une interprétation probabiliste d'une formulation quadratique continue a permis de traiter le problème d'isomorphisme de sous-graphe [3]. À notre connaissance, aucune formulation linéaire en nombre entiers n'a été proposée pour le problème de l'isomorphisme de sous-graphe.

On note $V_{\mathcal{X}}$ et $E_{\mathcal{X}}$ les ensembles de sommets et d'arcs d'un graphe $\mathcal{X}$. Étant donnés deux graphes $\mathcal{G} = (V_{\mathcal{G}}, E_{\mathcal{G}})$ et $\mathcal{S} = (V_{\mathcal{S}}, E_{\mathcal{S}})$, le problème de l'isomorphisme de sous-graphe consiste à trouver un sous-graphe $\mathcal{G}' = (V_{\mathcal{G}'}, E_{\mathcal{G}'})$ de $\mathcal{G}$ isomorphe à $\mathcal{S}$, avec $V_{\mathcal{G}'} \subseteq V_{\mathcal{G}}$ et $E_{\mathcal{G}'} \subseteq \{e = (v_1, v_2) \in E_{\mathcal{G}} / v_1, v_2 \subseteq V_{\mathcal{G}'}\}$.

Pour de nombreuses applications, des étiquettes sont associées aux sommets ou aux arcs. On cherche alors le sous-graphe $\mathcal{G}'$ le plus proche de $\mathcal{S}$, au sens de la minimisation des différences d'étiquettes entre sommets associés et arcs associés. Nous présentons ici deux formulations sous forme de programmes linéaires 0-1 pour ce problème. Dans la variante de l'isomorphisme inexact, on recherche un sous-graphe $\mathcal{G}'$ qui n'a pas nécessairement la même structure que $\mathcal{S}$. Nous exposons également une formulation autorisant ces différences de structure, les pénalisant au même titre que les différences d'étiquettes.

2 Deux formulations pour l'isomorphisme de sous-graphe

On considère deux graphes orientés $\mathcal{G}$ et $\mathcal{S}$, dont les arcs et les sommets sont valués. Les formulations suivantes permettent de trouver un isomorphisme de sous-graphe, inexact au sens des étiquettes. On note $d(.,.)$ la distance entre deux sommets ou deux arcs, l'un pris dans $\mathcal{S}$, l'autre dans $\mathcal{G}$; les distances sont des données du problème.

On utilise des variables de décision pour les sommets et les arcs. $x_{i,k} = 1$ signifie qu'on apparie le sommet $k \in V_\mathcal{G}$ au sommet $i \in V_\mathcal{S}$. De manière similaire, $y_{ij,kl} = 1$ si l'arc $kl \in E_\mathcal{G}$ est associé à l'arc $ij \in E_\mathcal{S}$.

(F1)

$$\text{Minimiser} \sum_{i \in V_\mathcal{S}} \sum_{k \in V_\mathcal{G}} d(i,k) * x_{i,k} + \sum_{ij \in E_\mathcal{S}} \sum_{kl \in E_\mathcal{G}} d(ij,kl) * y_{ij,kl} \quad (1)$$

$$\text{s.c.} \sum_{k \in V_\mathcal{G}} x_{i,k} = 1 \quad \forall i \in V_\mathcal{S} \quad (2)$$

$$\sum_{kl \in E_\mathcal{G}} y_{ij,kl} = 1 \quad \forall ij \in E_\mathcal{S} \quad (3)$$

$$\sum_{i \in V_\mathcal{S}} x_{i,k} \leq 1 \quad \forall k \in V_\mathcal{G} \quad (4)$$

$$y_{ij,kl} \leq x_{i,k} \quad \forall ij \in E_\mathcal{S}, \forall kl \in E_\mathcal{G} \quad (5)$$

$$y_{ij,kl} \leq x_{j,l} \quad \forall ij \in E_\mathcal{S}, \forall kl \in E_\mathcal{G} \quad (6)$$

$$y_{ij,kl} \geq x_{i,k} + x_{j,l} - 1 \quad \forall ij \in E_\mathcal{S}, \forall kl \in E_\mathcal{G} \quad (7)$$

$$x_{i,k} \in \{0,1\} \quad \forall i \in V_\mathcal{S}, \forall k \in V_\mathcal{G} \quad (8)$$

$$y_{ij,kl} \in \{0,1\} \quad \forall ij \in E_\mathcal{S}, \forall kl \in E_\mathcal{G} \quad (9)$$

Notons $n_S = |V_\mathcal{S}|$, $n_G = |V_\mathcal{G}|$, $m_S = |E_\mathcal{S}|$, et $m_G = |E_\mathcal{G}|$. (F1) utilise $n_S n_G + m_S m_G$ variables de décision binaires, et $n_S + m_S + 3m_S m_G$ contraintes.

On souhaite minimiser les distances entre chaque sommet (ou arc) du graphe $\mathcal{S}$ et le sommet (respectivement arc) de $\mathcal{G}$ associé à celui-ci. La fonction objectif (1) minimise donc la somme de ces distances. La contrainte (2) permet de s'assurer qu'à chaque sommet de $\mathcal{S}$ est bien associé un et un seul sommet de $\mathcal{G}$. De la même manière, la contrainte (3) permet de s'assurer que chaque arc de $\mathcal{S}$ est bien apparié à un et un seul arc de $\mathcal{G}$. La contrainte (4) contraint chaque sommet de $\mathcal{G}$ à être associé à au plus un sommet de $\mathcal{S}$. La contrainte (5) implique que si un arc de $\mathcal{S}$ est apparié avec un arc de $\mathcal{G}$, leurs sommets d'origine doivent également être appariés. Il en va de même avec leurs sommets d'arrivée, dont la contrainte (6) impose l'association. La contrainte (7) impose que deux arcs soient appariés si leurs extrémités sont appariées deux à deux.

On peut exprimer différemment les contraintes liant les variables de décision des sommets et celles des arcs :

$$\sum_{ij \in E_\mathcal{S}} y_{ij,kl} = x_{i,k} \quad \forall i \in V_\mathcal{S}, \forall kl \in E_\mathcal{G}$$

Cette relation se comprend de la manière suivante : pour un arc kl de $E_\mathcal{G}$ et pour un sommet i de $V_\mathcal{S}$, si i et k sont appariés, alors il y a exactement un arc issu de i apparié à kl. Si i et k ne sont pas appariés, alors aucun arc issu de i ne peut être associé à kl. Les relations suivantes

sont bâties selon le même principe :

$$\sum_{ij \in E_S} y_{ij,kl} = x_{j,l} \quad \forall j \in V_S, \forall kl \in E_G$$
$$\sum_{kl \in E_G} y_{ij,kl} = x_{i,k} \quad \forall k \in V_G, \forall ij \in E_S$$
$$\sum_{kl \in E_G} y_{ij,kl} = x_{j,l} \quad \forall l \in V_G, \forall ij \in E_S$$

Ces relations nous permettent de proposer la formulation suivante :

(F2)

$$\text{Minimiser} \sum_{i \in V_S} \sum_{k \in V_G} d(i,k) * x_{i,k} + \sum_{ij \in E_S} \sum_{kl \in E_G} d(ij,kl) * y_{ij,kl} \tag{10}$$
$$\text{s.c.} \sum_{k \in V_G} x_{i,k} = 1 \quad \forall i \in V_S \tag{11}$$
$$\sum_{kl \in E_G} y_{ij,kl} = 1 \quad \forall ij \in E_S \tag{12}$$
$$\sum_{i \in V_S} x_{i,k} \leq 1 \quad \forall k \in V_G \tag{13}$$
$$\sum_{kl \in E_G} y_{ij,kl} = x_{i,k} \quad \forall k \in V_G, \forall ij \in E_S \tag{14}$$
$$\sum_{kl \in E_G} y_{ij,kl} = x_{j,l} \quad \forall l \in V_G, \forall ij \in E_S \tag{15}$$
$$x_{i,k} \in \{0,1\} \quad \forall i \in V_S, \forall k \in V_G \tag{16}$$
$$y_{ij,kl} \in \{0,1\} \quad \forall ij \in E_S, \forall kl \in E_G \tag{17}$$

(F2) utilise $n_S n_G + m_S m_G$ variables de décision binaires, et $n_S + 2m_S n_G$ contraintes.
Soit $\mathcal{P}_1$ (respectivement $\mathcal{P}_2$) le polyèdre des solutions de la relaxation continue de (F1) (respectivement (F2)).

Proposition *Nous démontrons que $\mathcal{P}_2 \subseteq \mathcal{P}_1$ et que cette inclusion est stricte au moins dans certains cas.*

Les premiers résultats numériques confirment la supériorité de la formulation (F2).

3 Isomorphisme inexact de sous-graphe

On propose ici une formulation robuste à l'absence dans $\mathcal{G}$ de sommets ou d'arcs pouvant être associés à ceux du graphe $\mathcal{S}$. On procède par l'ajout direct dans $\mathcal{G}$ des éléments non appariés de $\mathcal{S}$. Cet ajout est effectué à l'aide de variables de décision plutôt qu'en modifiant les données du problème.

On introduit les variables binaires $u_i \ \forall i \in V_S$ et $e_{ij} \ \forall ij \in E_S$. u_i est associée au sommet i de $\mathcal{S}$ de la manière suivante : $u_i = 0$ si et seulement si un sommet k de $\mathcal{G}$ est associé à i. Dans le cas contraire, pour pouvoir identifier un isomorphisme de $\mathcal{S}$ dans $\mathcal{G}$, on considère que le sommet i se trouve également dans $\mathcal{G}$, ce qu'on traduit par $u_i = 1$.

Les variables e_{ij} revêtent la même signification pour les arcs que les variables u_i pour les sommets. Ils indiquent si un arc de $\mathcal{S}$ a été assigné à un arc de $\mathcal{G}$ ou si il a été ajouté dans $\mathcal{G}$.

L'ajout d'un sommet et d'un arc engendre un sur-coût. On note $c(.)$ ce coût de création, pour les sommets et les arcs de $\mathcal{S}$.

On introduit (F1a), une formulation basée sur (F1) utilisant les variables u_i et e_{ij} :

(F1a)

$$\text{Minimiser} \sum_{i \in V_\mathcal{S}} \sum_{k \in V_\mathcal{G}} d(i,k) * x_{i,k} + \sum_{ij \in E_\mathcal{S}} \sum_{kl \in E_\mathcal{G}} d(ij,kl) * y_{ij,kl} + \sum_{i \in V_\mathcal{S}} c(i) * u_i + \sum_{ij \in E_\mathcal{S}} c(ij) * e_{ij} \tag{18}$$

$$\text{s.c.} \sum_{k \in V_\mathcal{G}} x_{i,k} + u_i = 1 \quad \forall i \in V_\mathcal{S} \tag{19}$$

$$\sum_{kl \in E_\mathcal{G}} y_{ij,kl} + e_{ij} = 1 \quad \forall ij \in E_\mathcal{S} \tag{20}$$

$$\sum_{i \in V_\mathcal{S}} x_{i,k} \leq 1 \quad \forall k \in V_\mathcal{G} \tag{21}$$

$$y_{ij,kl} \leq x_{i,k} \quad \forall ij \in E_\mathcal{S}, \forall kl \in E_\mathcal{G} \tag{22}$$

$$y_{ij,kl} \leq x_{j,l} \quad \forall ij \in E_\mathcal{S}, \forall kl \in E_\mathcal{G} \tag{23}$$

$$y_{ij,kl} \geq x_{i,k} + x_{j,l} - 1 \quad \forall ij \in E_\mathcal{S}, \forall kl \in E_\mathcal{G} \tag{24}$$

$$x_{i,k} \in \{0,1\} \quad \forall i \in V_\mathcal{S}, \forall k \in V_\mathcal{G} \tag{25}$$

$$y_{ij,kl} \in \{0,1\} \quad \forall ij \in E_\mathcal{S}, \forall kl \in E_\mathcal{G} \tag{26}$$

$$u_i \in \{0,1\} \quad \forall i \in V_\mathcal{S} \tag{27}$$

$$e_{ij} \in \{0,1\} \quad \forall ij \in E_\mathcal{S} \tag{28}$$

La fonction objectif ainsi que les contraintes de (F1a) correspondent à celles de (F1), adaptées si besoin est à l'aide des variables u_i et e_{ij}. Les contraintes de (21) à (26) sont identiques à celles de (F1).

La fonction objectif (18) reprend sur la première ligne les termes de la fonction objectif (1) de la formulation (F1). La deuxième ligne correspond aux coûts de création des éléments de $\mathcal{S}$ qui n'ont été associé à aucun élément de $\mathcal{G}$. La contrainte (19) implique que si l'on n'associe pas un sommet k à chaque sommet i, alors il faut créer i dans $\mathcal{G}$. La contrainte (20) joue le même rôle pour les arcs ij.

De façon similaire, nous étendons la formulation (F2) pour prendre en compte les différences structurelles. Nous comparons ces formulations à partir de résultats numériques et envisageons les améliorations possibles.

Références

[1] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 2004.

[2] Derek Justice. A binary linear programming formulation of the graph edit distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8) :1200–1214, 2006. Fellow-Alfred Hero.

[3] C. Schellewald and C. Schnorr. Probabilistic subgraph matching based on convex relaxation. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 171–186, 2005.