



HAL
open science

Résoudre le problème de la distance d'édition entre graphes avec les matheuristiques

Mostafa Darwiche, Romain Raveaux, Donatello Conte, Vincent T'kindt

► To cite this version:

Mostafa Darwiche, Romain Raveaux, Donatello Conte, Vincent T'kindt. Résoudre le problème de la distance d'édition entre graphes avec les matheuristiques. ROADEF19, Feb 2019, Le Havre, France. ⟨hal-02077682⟩

HAL Id: hal-02077682

<https://hal.science/hal-02077682v1>

Submitted on 23 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Résoudre le problème de la distance d'édition entre graphes avec les matheuristiques

Mostafa Darwiche^{1,2}, Romain Raveaux¹, Donatello Conte¹, Vincent T'kindt²

¹ Laboratoire d'Informatique Fondamentale et Appliquée de Tours (EA 6300), Université de Tours,
France

² Laboratoire d'Informatique Fondamentale et Appliquée de Tours (EA 6300), ERL-CNRS 6305,
Université de Tours, France

{mostafa.darwiche,romain.raveaux,donatello.conte,tkindt}@univ-tours.fr

Mots-clés : *problème de la distance d'édition, appariement de graphes, matheuristique.*

1 Introduction

Dans le domaine de la reconnaissance de formes structurales, dans lequel les graphes sont fortement utilisés pour modéliser des objets, il apparait des problèmes d'optimisation qui sont la plupart de temps \mathcal{NP} -difficiles. C'est le cas du problème de la distance d'édition entre graphes (DEG) [7]. Ce problème fournit une mesure de dissimilarité entre deux graphes, qui peut être utilisée dans le domaine de la vision par ordinateur, pour effectuer des tâches comme : la reconnaissance automatique des formes ou le suivi d'objets dans des vidéos [6]. Ce problème est aussi appliqué dans les domaines de la chimie et la biologie pour comparer des molécules chimiques, des protéines ou des enzymes [5]. L'idée de base est de transformer un graphe G en un graphe G' en effectuant un ensemble d'opérations d'édition. Ces opérations sont des substitutions, suppressions, insertions de nœuds ou d'arêtes et un coût est associé à chaque opération. Le problème est combinatoire parce que pour le résoudre, il faut trouver l'ensemble des opérations dont le coût total est minimum. Récemment, une matheuristique a été proposée dans [1] pour résoudre un sous-problème du problème DEG. Ce sous-problème considère seulement les instances des graphes où les arêtes ne possèdent pas d'attributs. Ce qui signifie d'avoir des coûts fixes pour les opérations d'édition sur les arêtes. La matheuristique proposée est une version adaptée de la matheuristique *branchement local* (BraLoc) développée par Fischetti et Lodi [4]. BraLoc intègre et guide un solveur pour explorer l'espace des solutions d'une formulation mathématique (programme linéaire en nombre entiers - PLNE). Elle est constituée de trois étapes principales : la définition d'un voisinage, l'intensification et la diversification. Dans [1], la définition du voisinage et la diversification sont modifiées en considérant des informations et des propriétés du sous-problème DEG, afin d'améliorer la performance de l'heuristique. Les résultats obtenus ont montré l'efficacité de BraLoc comparée aux heuristiques existantes.

Dans ce travail, nous présentons une nouvelle version de BraLoc qui résout le problème général. En outre, nous proposons une version adaptée d'une autre matheuristique qui s'appelle *la recherche locale par partitionnement des variables* (RLPV), et qui a été proposée dans [3].

2 Des matheuristiques pour le problème DEG

Branchement local. Dans [1], BraLoc est limité à un sous-problème à cause du modèle de PLNE choisi. Pour pouvoir appliquer BraLoc sur le problème général, il suffit de remplacer le modèle par un autre qui modélise le problème général. Le modèle sélectionné est alors F3, qui a été montré comme un des plus efficace pour résoudre le problème DEG [2]. Ensuite,

la technique de la définition du voisinage a été améliorée en redéfinissant la *contrainte du branchement locale* comme dans l'équation suivante :

$$\Delta(x, x^p) = \sum_{(i,k) \in S^p} (1 - x_{i,k}) + \sum_{(i,k) \notin S^p} x_{i,k} \leq \pi \quad (1)$$

avec $S^p = \{(i, k) : x_{i,k}^p = 1\}$ l'ensemble des variables ayant une valeur égale à 1 dans une solution x^p . Le voisinage contient les solutions qui sont autour de x^p avec une distance π (dans le sens de la *distance de Hamming*). Une contrainte similaire a été proposée pour la diversification, mais plutôt sur un sous-ensemble bien défini des variables $x_{i,k}$. Le sous-ensemble contient des variables considérées comme *importantes*, puisque la fonction objectif est très impactée quand leurs valeurs changent dans une solution. Une procédure qui analyse les instances des graphes est exécutée pour définir ce sous-ensemble.

Recherche locale par partitionnement des variables. RLPV est une matheuristique qui effectue des recherches locales dans des voisinages bien définis du modèle F3. Étant donné une solution x^p , le voisinage est défini par :

$$N(x^p, S) = \{x_{i,k} \mid x_{i,k} = x_{i,k}^p, \forall (i, j) \notin S\} \quad (2)$$

avec S un sous-ensemble des variables binaires qui sont considérées *spéciales* dans le modèle. Une étape essentielle dans RLPV est de déterminer les variables *spéciales*, qui vont rendre le voisinage riche avec des solutions améliorantes. Nous proposons un algorithme basé sur la notion de sphère dans les graphes d'entrée. L'algorithme détermine la sphère qui contribue le plus à la fonction objectif, et donc toutes les variables binaires qui représentent des nœuds appartenant à cette sphère, sont ajoutées à l'ensemble S .

Lors de la conférence nous présenterons les résultats expérimentaux des deux matheuristiques que l'on propose afin de mettre en évidence leur efficacité par rapport aux méthodes existantes.

Références

- [1] Mostafa Darwiche, Donatello Conte, Romain Raveaux, and Vincent T'Kindt. A local branching heuristic for solving a graph edit distance problem. *Computers & Operations Research*, 2018.
- [2] Mostafa Darwiche, Romain Raveaux, Donatello Conte, and Vincent T'Kindt. Graph edit distance in the exact context. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 304–314. Springer, 2018.
- [3] F Della Croce, A Grosso, and F Salassa. Matheuristics : embedding milp solvers into heuristic algorithms for combinatorial optimization problems. In Patrick Siarry, editor, *The Oxford Handbook of Innovation*, chapter 3. NOVA Publisher, 2013.
- [4] Matteo Fischetti and Andrea Lodi. Local branching. *Mathematical programming*, 98(1-3) :23–47, 2003.
- [5] John W Raymond and Peter Willett. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of computer-aided molecular design*, 16(7) :521–533, 2002.
- [6] Alberto Sanfeliu, René Alquézar, J Andrade, Joan Climent, Francesc Serratos, and J Vergés. Graph-based representations and techniques for image processing and image analysis. *Pattern recognition*, 35(3) :639–650, 2002.
- [7] Zhiping Zeng, Anthony KH Tung, Jianyong Wang, Jianhua Feng, and Lizhu Zhou. Comparing stars : on approximating graph edit distance. *Proceedings of the VLDB Endowment*, 2(1) :25–36, 2009.