HAL
open science

Recherche de motifs quasi-similaires dans des graphes

Fanny Chevalier, Maylis Delest, Jean-Philippe Domenger

To cite this version:

Fanny Chevalier, Maylis Delest, Jean-Philippe Domenger. Recherche de motifs quasi-similaires dans des graphes. EGC 2008, 2008, France. hal-00306972

HAL Id: hal-00306972

https://hal.science/hal-00306972v1

Submitted on 28 Jul 2008

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.

Recherche de motifs quasi-similaires dans des graphes

Fanny Chevalier*, Maylis Delest*
Jean-Philippe Domenger*

*Université de Bordeaux, 351 Cours de la Libération, 33400 TALENCE
nom.prenom@labri.fr,
http://www.labri.fr/

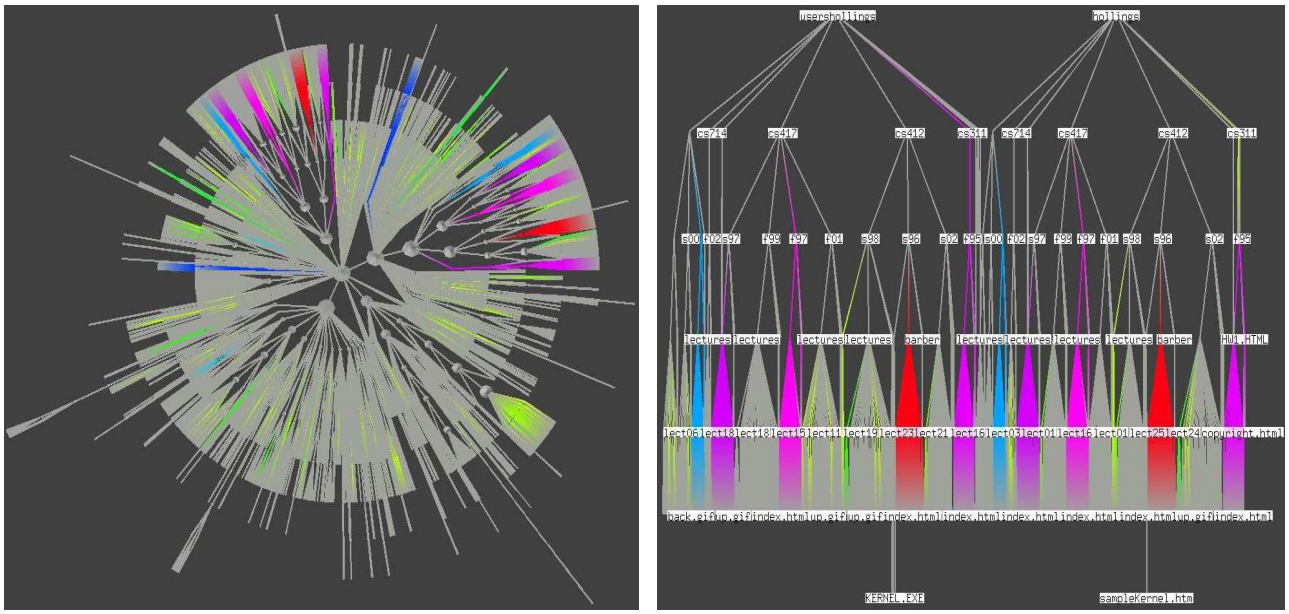
Résumé. Nous décrivons un algorithme basé sur des métriques intrinsèques de graphes permettant de découvrir des motifs communs et similaires entre plusieurs graphes. Nous montrons des applications à la recherche d'image dans une collection et à l'interprétation de données géographiques.

1 Introduction

Il existe un très large panel de méthodes et algorithmes d'analyse, manipulation, navigation, dessin, visualisation pour l'étude et la représentation des graphes. L'une des tâches les plus importantes liées à l'utilisation des graphes est leur comparaison. Ainsi, le problème de mise en correspondance de graphes et de sous-graphes fait l'objet d'une recherche intensive. Le problème d'isomorphisme de graphes et de sous-graphes étant coûteux, un grand nombre de techniques proposées exploitent les spécificités des données du domaine d'application afin d'optimiser les traitements en limitant les comparaisons. D'autre part, selon l'application visée, il est souvent pertinent de détecter dans des structures de graphes des éléments qui se ressemblent fortement, sans pour autant être identiques. En effet, la détection de motifs ressemblants que l'on appellera *motifs quasi-similaires* peut se révéler une source d'information importante. En vision par ordinateur, Lladós et al. (2001) ont proposé une méthode de reconnaissance de symboles graphiques ou de diagrammes manuscrits. Dans leur approche, un graphe d'adjacence des régions est associé à chaque symbole. La reconnaissance s'effectue par comparaison de graphes en utilisant une méthode de coût d'édition. En chimie, on peut citer les travaux de Yan et al. (2006) pour la recherche de sous-structures dans des molécules chimiques. Chaque molécule est dans un premier temps associée à un graphe. Ce graphe est ensuite décomposé en un ensemble de motifs élémentaires prédéfinis. Ces motifs servent d'index pour retrouver une molécule dans une base de données. Dans le domaine du multimédia, de nombreux travaux sont basés sur les techniques de mise en correspondance de graphes. On peut citer Gomila et Meyer (2003) qui utilisent des méthodes de relaxation probabiliste pour le suivi d'objets segmentés dans la vidéo. Demirci et al. (2006) proposent une mise en correspondance des sommets telle que plusieurs régions d'une image requête (associées à plusieurs sommets dans le graphe d'adjacence des régions), peuvent être mises en correspondance avec une seule et même région (donc un unique sommet dans le graphe d'adjacence) de l'image cible. En maintenance de code informatique, les arbres syntaxiques correspondant au code source sont utilisés pour la détection de duplication de code. Baxter et al. (1998) ont proposé

une méthode basée sur la classification des sommets par similarité (au vu de la composition du sous-arbre). Cette classification implique que deux sommets classés dans la même catégorie correspondent à deux sommets racines de sous-arbres similaires.

La méthode, que nous détaillons ici, s'appuie sur une heuristique utilisant les caractéristiques structurelles des sommets pour la mise en correspondance. Le résultat de la mise en correspondance de motifs quasi-similaires est présenté comme une coloration des sommets telle que deux motifs de la même couleur sont des motifs quasi-similaires. A l'origine, cette méthode a été développée pour répondre aux tâches concernant la comparaison d'arborescences de fichiers. L'un des objectifs fixés par le concours Infovis (Fekete et Plaisant (2003)) était de proposer une méthode de détection visuelle des similarités et/ou changements entre deux versions différentes d'une même arborescence. La solution logicielle EVAT de Auber et al. (2003) a été proposée dans ce cadre. L'interface de visualisation permet à l'utilisateur d'identifier visuellement, par la coloration des sommets et arêtes des arborescences (ici en blanc), si des changements ont eu lieu entre deux versions d'un système arborescent de fichiers. Une même couleur suggère des motifs quasi-similaires. Sur la figure 1.a, on remarque la présence de deux sous-arbres proches [1]. En montrant le détail de ces deux sous-arbres (Fig. 1.b) on constate en effet que les répertoires nommés `hollings` et `usershollings` sont très proches en terme de structure.



(a) (b)

FIG. 1 – *EVAT : Visualisation de motifs communs dans un système de fichiers.*

[1]Pour plus de lisibilité, on recommande vivement au lecteur de consulter la version couleur de l'article en ligne à l'adresse http ://www.labri.fr/publications/mabiovis/. Dans cette version, des motifs rouges, roses et bleus sont visibles dans les deux sous-arbres

Par la suite, Auber et al. (2006) ont repris la méthode dans un algorithme pour le dessin de structures secondaires d'ARN. Le principe est d'identifier le plus grand motif similaire entre deux structures. Ce motif est alors dessiné à une même place et avec la même orientation dans chacune des fenêtres dédiées à la représentation des ARNs. Ainsi, le motif similaire permet à l'utilisateur d'avoir un point de référence pour la comparaison. Enfin, dans le domaine de la visualisation de l'information, Chevalier et al. (2007a) ont proposé une technique de visualisation permettant d'identifier visuellement les changements au cours du temps des blocs de codes en génie logiciel.

Nous proposons dans cet article une formalisation de cette heuristique généralisée aux graphes. L'heuristique procède en deux phases. Dans un premier temps, les sommets des graphes sont classifiés en différentes familles de sommets partageant les mêmes caractéristiques. La deuxième phase concerne la construction des motifs quasi-similaires : l'algorithme se base sur la comparaison des étiquetages des voisinages des sommets à comparer.

Nous avons introduit l'arbre de couverture de Beygelzimer et al. (2006) pour optimiser la phase de construction des familles de sommets aux mêmes propriétés. D'autre part, nous définissons ici deux variantes pour la reconnaissance de motifs quasi-similaires. Ces variantes se distinguent par la stratégie de comparaison du voisinage. L'ensemble des sommets considérés pour la comparaison de motifs, selon la variante, est constitué de :

– une partie du voisinage direct des sommets
– l'intégralité du voisinage direct des sommets

Dans ce qui suit nous présentons tout d'abord la classification des sommets puis dans la section suivante la contruction des motifs. La section 4 est consacré aux applications. Enfin, nous proposons une conclusion en section 5.

2 Classification des sommets

La première étape de la méthode est consacrée à la classification des sommets des graphes selon une mesure de similarité basée sur leurs propriétés structurelles. Les caractéristiques structurelles considérées sont liées au type d'application. Cette classification est une première étape à la comparaison des sous-structures des graphes : deux sous-graphes dont les sommets racines ont des caractéristiques structurelles (locales et globales) proches sont intéressants à comparer.

La classification des sommets en familles distinctes s'appuie sur un ensemble de caractéristiques associées à chaque sommet.

Les métriques utilisées ici pour évaluer la similarité entre deux sommets correspondent à des métriques intrinsèques, calculées sur les sommet d'un graphe, indépendamment des informations propres aux données étudiées. En effet nous pensons que quelle que soit l'application visée, la structure représente une caractéristique pertinente qu'il est important de considérer pour la comparaison. Parmi les caractéristiques structurelles, on peut citer, pour un sommet u, le degré, le nombre de sommets à distance n de u, le nombre de cliques de taille k qui contiennent u ; si l'on considère des arbres, la distance à la racine, la taille du sous-arbre enraciné en u, le nombre de strahler (Auber et al. (2004)), *etc*.

Soit G (resp. G') un graphe d'ensemble de sommets V (resp. V') et d'arêtes E (resp. E'). Pour une métrique μ, la valeur associée à un sommet $u \in V \cup V'$, correspond à la valeur

normalisée $\tilde{\mu}(u)$ de la métrique μ selon la formule :

$$\tilde{\mu}(u) = \frac{\mu(u) - \mu_{min}}{\mu_{max} - \mu_{min}} \tag{1}$$

avec $\mu_{min} = \min_{v \in V \cup V'} \mu(v)$ et $\mu_{max} = \max_{v \in V \cup V'} \mu(v)$.

Ainsi, pour un ensemble de n métriques $\{\mu_i\}_{i \in \{1, \cdots, n\}}$, nous calculons pour chaque sommet $u \in V \cup V'$, le vecteur caractéristique associé $(\tilde{\mu}_1(u), \tilde{\mu}_2(u), \cdots, \tilde{\mu}_n(u))$.

A cette étape, nous construisons les différentes familles $\mathcal{F}$ contenant des sommets aux propriétés similaires. Nous définissons un étiquetage λ des sommets tel que si deux sommets u et v appartiennent à une même famille $\mathcal{F}$, alors $\lambda(u) = \lambda(v)$.

Définition 1 (Sommets ϵ-similaires) *Soient μ_i, $i \in \{1, \cdots, n\}$ l'ensemble des métriques considérées et soit le seuil de similarité $\epsilon \in [0, 1]$. On dit que deux sommets u et v sont ϵ-similaires s'ils vérifient :*

$$dist(u, v) < \epsilon \tag{2}$$

Dans cet article, $dist(u, v)$ correspond à la distance euclidienne entre les vecteurs. La construction des familles de sommets ϵ-similaires se base donc sur une comparaison deux à deux des sommets des graphes à laquelle nous intégrons une stratégie de partitionnement plus efficace pour la construction des familles. L'algorithme est basé sur une variante de la méthode d'*arbre de couverture* décrite dans Beygelzimer et al. (2006). Pour définir l'arbre de couverture, il est nécessaire d'introduire dans un premier temps les deux définitions de *séparabilité* et de *couverture* suivantes :

Définition 2 (Séparabilité) *Soit un ensemble S d'éléments définis dans un espace métrique. Soit* dist *une distance sur cet espace et soit θ un réel.*

On dit que l'ensemble S vérifie la condition de séparabilité de seuil θ si, pour tout $u, v \in S$, on a $dist(u, v) > \theta$.

Définition 3 (Couverture) *Soient un ensemble S d'éléments définis dans un espace métrique et S' un sous-ensemble de S. Soit* dist *une distance sur cet espace et soit θ un réel.*

On dit que l'ensemble S' vérifie la condition de couverture (de seuil θ) de l'ensemble S si, pour tout $u \in S$, il existe $v \in S'$ tel que $dist(u, v) \leq \theta$.

Un arbre de couverture T défini pour un ensemble d'éléments S est un arbre multi-niveaux. Le niveau 0 correspond aux feuilles de l'arbre, et le niveau le plus élevé (dénoté i_{max}) correspond à la racine. A chaque niveau i, est associé un réel θ_i tel que $\theta_i < \theta_{i+1}$. Soit S_i l'ensemble des éléments au niveau i.

Définition 4 (Arbre de couverture) *L'arbre de couverture T défini pour l'ensemble d'éléments S vérifie les propriétés suivantes pour tout i :*

- *$S_i \subset S_{i-1}$*
- *S_i est une couverture de l'ensemble S_{i-1}*
- *S_i vérifie la propriété de séparabilté*

L'arbre de couverture n'est pas nécessairement unique : le contenu des familles peut varier en fonction de l'ordre dans lequel les sommets sont traités car la construction de l'arbre de couverture dépend de l'ordre d'insertion. Auber (2003) propose un algorithme de dessin de graphes incrémental pour la visualisation progressive de la structure à afficher. Le principe est de calculer un ordre sur les sommets afin d'afficher progressivement (en proposant à l'utilisateur le rendu à différentes étapes de calcul du dessin) la structure du graphe. Dans ce contexte, il est important de faire apparaître dès les premiers calculs la forme générale du graphe afin d'offrir à l'utilisateur une vision globale de la structure qu'il souhaite visualiser. Pour ce faire, il utilise un ordre sur les sommets : en considérant les sommets par ordre inverse de leur nombre de Strahler, il semble que la vue schématique de l'image finale (moins de 10% des éléments affichés) donne une bonne représentation de l'allure générale du graphe. Nous avons donc choisi, pour construire l'arbre de couverture d'un graphe, de traiter les sommets dans l'ordre inverse de leur nombre de Strahler, positionnant ainsi les sommets les plus importants comme représentants des familles de sommets.

La procédure d'étiquetage des sommets est décrite par l'algorithme 1,

- T_u désigne le sous-arbre enraciné en u,
- $prof_T(u)$ désigne la distance du sommet u à la racine de l'arbre T
- $\sigma(u)$ correspond au nombre de Strahler du sommet u

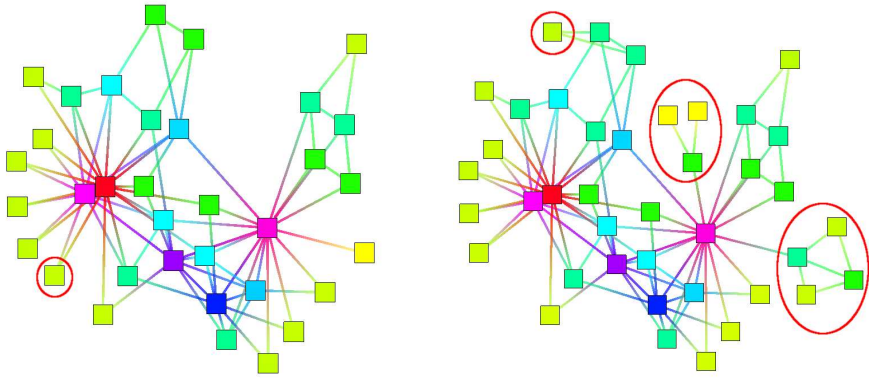
```
1  S := V ∪ V'
2  trier  S par σ décroissant;
3  T := arbre de couverture de l'ensemble trié  S
4  R := {u ∈ T | prof_T(u) = i}
5  Entier  L := 1;
6  Pour u ∈ R Faire
7      U := v ∈ T_u
8      Pour v ∈ U Faire
9          λ(v) := L
10     FinPour
11     L := L + 1;
12 FinPour
```

Algorithme 1 – *Procédure d'étiquetage des familles*

Nous n'avons pas abordé dans ces travaux le problème de l'évolution des graphes. La modification des graphes nécessite de recalculer la classification des sommets. Cependant, seuls les sommets dont le vecteur de caractéristiques a été altéré (et leurs successeurs dans l'arbre de couverture) doivent être reconsidérés pour la classification. Ainsi, il est possible d'adapter la méthode à des graphes dynamiques en reconstruisant partiellement l'arbre de couverture pour la classification des sommets.

Par la suite, on note $\mathcal{F}_l$ la famille de sommets u d'étiquette $\lambda(u) = l$. La figure 2 montre un exemple de calcul de familles de sommets ϵ-équivalents sur l'ensemble des sommets de deux versions du classique graphe "club de karaté" de Zachary (1977). Les formes texturées représentent les différentes valeurs d'étiquettes, une même forme texturée correspondant à une même étiquette. Le premier (Figure 2.a) correspond aux données originales, que nous avons légèrement modifié manuellement en ajoutant des sommets et des arêtes (les éléments

qui diffèrent sont détourés dans les figures) pour obtenir un second graphe (Fig. 2.b). Nous pouvons d'ores et déjà remarquer des ressemblances au niveau du contenu [2].



(a) (b)

FIG. 2 – *Exemple de classification en familles de sommets ϵ-équivalent. (a) le graphe original du club de karaté et (b) le même graphe après modifications.*

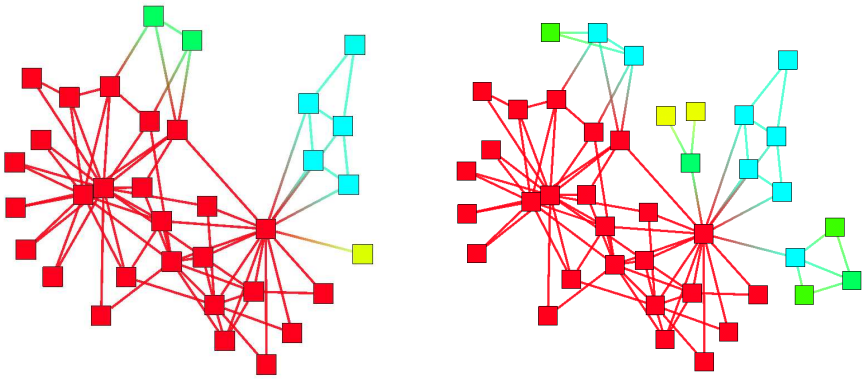
Complexité Il a été prouvé par Don (2006) que l'algorithme de construction de l'arbre de couverture d'un ensemble de points S défini sur un espace métrique E a pour complexité $O(2^{2dd} n \log A)$, avec dd la dimension doublante de E associée à l'arbre, et $A = \frac{d_{max}}{d_{min}}$ l'*aspect ratio* de l'ensemble S (d_{min} et d_{max} étant les distances minimale et maximale respectivement entre les paires d'éléments distincts de S).

La phase de classification des sommets des graphes de l'exemple de la figure 6 (135 et 170 sommets) des réseaux de migrations alternantes 40 ms sur un processeur de 1,8 GHz. Pour deux graphes représentant le réseau aérien mondial de 1542 et 1781 sommets respectivement, le temps de calcul est de 820 ms.

3 Construction des motifs similaires

Si l'on considère $u \in G$ et $u' \in G'$ tels que $\lambda(u) = \lambda(u')$, alors les caractéristiques structurelles de u sont proches de celles de u' car les sommets sont dits ϵ-similaires. Il est probable que les sous-graphes G_u issu de u et $G_{u'}$ issu de u' soient similaires. La procédure de reconnaissance de motifs considère donc de telles paires de sommets $\{u, u'\}$. La construction du motif similaire se fait par propagation : l'algorithme traite les sommets depuis les sommets considérés u et u' vers les voisins identifiés comme appartenant au motif, et récursivement sur les voisins des voisins. Les sommets considérés au départ sont les sommets appartenant à la famille dont le représentant a la plus forte valeur de Strahler. On note que toute décision est définitive : si un ensemble de sommets est considéré comme appartenant à un motif, alors on ne reviendra pas sur cette décision. La figure 3 montre un exemple de propagation. Le principe général de l'heuristique consiste, pour chaque paire de sommets ϵ-similaires, à évaluer

[2] Le lecteur est invité à consulter la version couleur de l'article pour plus de lisibilité.



(a) (b)

FIG. 3 – *Exemple de reconnaissance des motifs correspondant à la classification de la figure 2*

la *similarité* de leur voisinage. On utilise pour cela une mesure de dissimilarité $D(C, C')$ entre deux ensembles de sommets C et C' définie, pour chacune des deux variantes. Cette mesure de dissimilarité se base sur la distribution des étiquettes des sommets de C et C'. Intuitivement, elle peut être assimilée à une distance d'édition entre les étiquetages des sommets de C et C' en considérant les opérations d'ajout et de suppression uniquement.

Si $D(C, C')$ est inférieure à un seuil de similarité τ fixé, alors on considère que les deux étiquetages sont cohérents et les ensembles de sommets sont dits τ-similaires : C et C' correspondent à un même motif. On énonce la définition de motifs τ-similaires comme suit :

Définition 5 (Motifs τ-similaires) *On dit que deux ensembles de sommets étiquetés C et C' sont τ-similaires si la mesure de dissimilarité $D(C, C')$ vérifie :*

$$D(C, C') \leq \tau$$

La procédure de reconnaissance est appliquée récursivement sur les sommets nouvellement inclus dans le motif.

Nous proposons ici deux versions de l'heuristique pour l'identification de motifs par propagation, correspondant à deux différentes stratégies de comparaison sur la similarité des sous-structures. Une troisième a été proposée par Chevalier et al. (2007a) dans le cadre des applications au géniel logiciel. Cette troisième version correspond à la figure 4.d.

Dans la première version, nous cherchons à mettre en correspondance les portions des voisinages directs par famille. Dans l'exemple de la figure 4.a, on considère qu'il existe un trop grand écart de cardinalité entre les ensembles de voisins de u et de v étiquetés par un rond. Ainsi, ces sommets voisins ne sont pas inclus dans le motif représenté par la zone de la figure 4.b. Par contre, les voisinages étiquetés d'un triangle sont assez proches en terme de cardinalité pour être inclus dans le motif. Ainsi nous avons :

Définition 6 *Soient $C_l(u)$ et $C_l(u')$ les ensembles des voisins de u et u' respectivement d'étiquette l. La distance d'édition d'étiquetage D_1 se calcule comme suit :*

$$D_1(C_l(u), C_l(u')) = abs(|C_l(u)| - |C_l(u')|)$$

Dans la deuxième version, on examine les ensembles des sommets voisins dans leur intégralité : il est nécessaire que les voisinages complets soient proches. Dans cette version, la construction des motifs similaires se fait “par niveaux” : on propage toujours sur l'ensemble des voisinages si ces derniers se correspondent. Ainsi, pour une paire de sommets, soit tous les voisins sont inclus dans le motif, soit aucun. Dans l'exemple de la figure 4.a, les ensembles des voisins de u et de v sont trop différents pour être considérés comme appartenant à des motifs similaires. Par contre, les compositions des voisinages de u_3 et u'_5 sont suffisamment proches pour constituer des motifs similaires (voir Fig. 4.c).

On utilise alors la mesure de dissimilarité suivante

Définition 7 *Soient $C(u)$ et $C(u')$ les ensembles des voisins de u et u' respectivement, quelle que soit leur étiquette. La mesure de dissimilarité D_2 est définie par*

$$D_2(C(u), C(u')) = \sum_{l \in [1, \cdots, L]} D_1(C_l(u), C_l(u'))$$

Complexité Nous n'avons pas prouvé la complexité globale de l'étape de propagation des sommets. Nous donnons cependant quelques indications.

Soient $G = (V, E)$ et $G' = (V', E')$ deux graphes et soit L le nombre de familles de sommets ϵ-similaires définies sur G et G'. Le nombre de comparaisons de sommets pour la phase de propagation est au pire $\sum_{i \in [1,L]} |\mathcal{F}_i \cap V||\mathcal{F}_i \cap V'|$. Dans ce cas, toutes les paires de sommets sont testées, ce qui implique que la propagation échoue systématiquement. Dans la pratique, ce cas extrême est très peu probable, ainsi, dès qu'une étape de propagation est effectuée, l'ensemble des sommets appartenant au motif nouvellement identifié sont exclus du processus de propagation par la suite. Ce qui a pour effet de réduire le nombre de comparaisons.

Pour une étape de propagation de motifs issus des sommets u et u', on effectue au pire L comparaisons (on compare le nombre de successeurs de u et de v appartenant à chacune des classes). Si la propagation est validée, alors le processus est récursif sur les sommets nouvellement aggrégés au motif, mais ces sommets ne seront plus considérés pour la propagation par la suite.

La phase de reconnaissance de motifs par propagation dans les graphes de l'exemple de la figure 6 (135 et 170 sommets) a été effectuée en 1s sur un processeur de 1,8 GHz. Pour deux graphes correspondant au réseau aérien mondial (1542 et 1781 sommets), le temps de calcul est de 8s.

Dans la section qui suit, nous présentons quelques applications de la méthode que nous venons de présenter.

4 Exemples d'application

L'heuristique de reconnaissance de motifs similaires présentée dans ce papier est générique et peut être adaptée à de nombreux domaines d'applications. Cette méthode a été utilisée pour la reconnaissance d'objets extraits de flux vidéo à très basse résolution (Chevalier et al. (2007b)). Le principe de l'application est de retourner à l'utilisateur les objets de la base de données qui correspondent le mieux à un objet requête. Un exemple de l'application sur deux requêtes (vignettes détourées) est montré sur la figure 5. Dans cet exemple, les 5 objets les plus

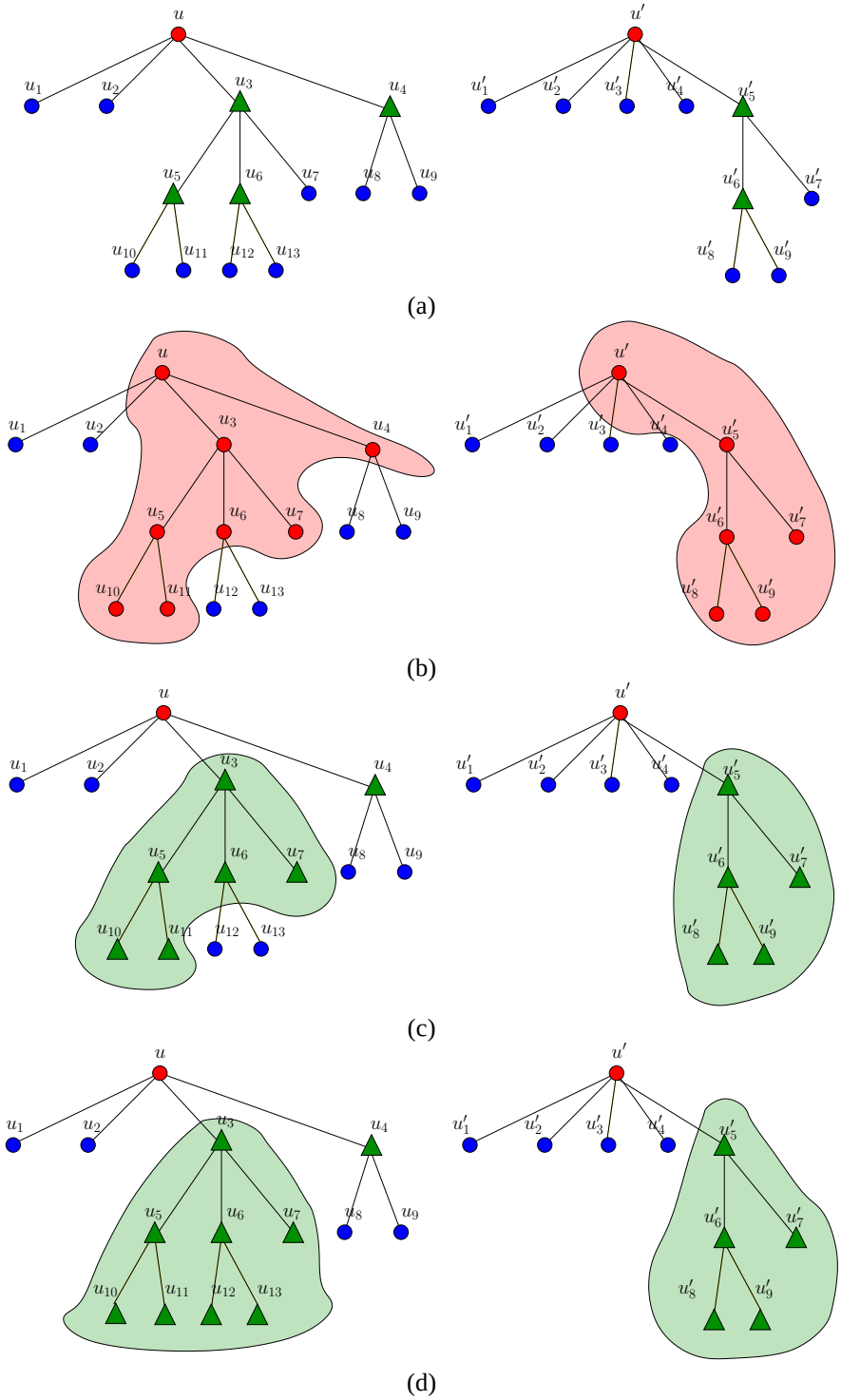


FIG. 4 – *Les différentes versions de propagation d'étiquette. (a) Deux graphes étiquetés en familles de sommets ϵ-similaires et (b) Version 1 : propagation sur les voisins par classe, (c) Version 2 : propagation sur l'ensemble des voisins et (d) Version 3 : propagation sur le sous-DAG (ou sous-arbre).*

similaires sont proposés à l'utilisateur. Dans cette application, on construit les graphes par segmentation à partir des objets de la vidéo et on applique la deuxième variante de l'heuristique pour déterminer les portions communes entre deux objets.

FIG. 5 – *Exemple d'une requête d'objet dans une base de données vidéo.*

Dans le cadre du projet ANR SPANGEO, nous avons appliqué cette méthode à l'analyse des migrations professionnelles en 1975 et en 1982 dans la région Provence Alpes Côte d'Azur. Les graphes sont orientés et les arêtes sont dirigées de la ville de résidence vers la ville de travail. Nous n'avons conservé que les arêtes du graphe telles que l'effectif dépasse 8 personnes. La première variante de la méthode est utilisée. La figure 6 montre le résultat de la reconnaissance de motifs quasi-similaires entre les années 1975 (Fig. 6.a) et 1982 (Fig. 6.b). On peut par exemple remarquer la présence d'un large motif commun entre les deux réseaux. Sur la figure 6, les sommets appartenant à un motif quasi-similaire sont représentés en foncé. Sur la figure 7, nous avons isolé le plus large motif quasi-similaire (en violet dans la version couleur). Dans cet exemple, 50 % des sommets sont communs aux deux motifs, avec notamment des villes pivots comme Le Cannet, Saint-Laurent-du-Var, Carros et Villeneuve-Loubet. Les géographes du projet SPANGEO ont considéré les résultats comme étant pertinents et utiles à leur analyse.

5 Conclusion et perspectives

Nous avons présenté dans cet article une méthode pour la reconnaissance de motifs similaires dans des graphes. En proposant deux variantes correspondant à deux différentes stratégies de tolérance sur la notion de similarité de structure, nous avons défini une méthode générique qui peut être adaptée à de nombreuses applications dans divers domaines tels que la biologie (Auber et al. (2006)) ou le multimédia (Chevalier et al. (2007b)).

Les applications basées sur la méthode ayant donné lieu à des publications concernent dans tous les cas des structures orientées sans cycles (DAGs ou arbres) dans lesquelles un parcours hiérarchique peut être effectué. Nous envisageons, en perspective de ces travaux, d'approfondir la méthode pour des applications nécessitant la comparaison de graphes non orientés tels que les réseaux géographiques, ou les réseaux sociaux, et l'optimisation des algorithmes pour permettre de traiter efficacement les graphes dynamiques en limitant les calculs nécessaires à l'introduction de nouvelles données.

Un autre point important à soulever dans le cadre de cette étude concerne l'évaluation et la validation de la méthode. En effet, si sur des applications particulières les résultats des comparaisons permettent de déterminer les performances de la méthode dans chaque cas précis,

il reste difficile de valider l'efficacité de l'algorithme dans le cas général de la comparaison de graphes. En effet, il n'existe pas à notre connaissance de jeu de données de référence pour lequel les motifs à retrouver sont exhaustivement définis, si ce n'est dans le cadre d'une application précise dans un domaine d'application particulier. Nous envisageons de constituer un tel jeu de données de référence pour permettre une évaluation précise des performances et limitations de la méthode proposée, ainsi que des ses avantages et inconvénients par rapport aux méthodes existantes.

Références

Auber, D. (2003). *Graph drawing software*, Chapter Tulip - A Huge Graph Visualization framework, pp. 105–126. Verlag.

Auber, D., M. Delest, J. Domenger, et S. Dulucq (2006). Efficient drawing and comparison of rna secondary structure. *Journal of Graph Algorithms and Applications 10*(2), 329–351.

Auber, D., M. Delest, J. Domenger, P. Ferraro, et R. Strandh (2003). EVAT : Environment for visualization and analysis of trees. In *IEEE Symposition on Information Visualisation Contest*, Volume www.cs.umd.edu/hcil/iv03contest/, pp. 124–126.

Auber, D., M. Delest, J. Fédou, J. Domenger, et P. Duchon (2004). New strahler numbers for rooted plane trees. In M. Drmota, P. Flajolet, D. Gardy, et B. Gittenberger (Eds.), *Third Colloquium on Mathematics and Computer Science, Algorithms, Trees, Combinatorics and Probabilities*, Trends in Mathematics, pp. 203–215. Vienna University of Technology : Birkhauser.

Baxter, I., A. Yahin, L. M. ans M. Sant'Anna, et L. Bier (1998). Clone detection using abstract syntax trees. In *IEEE International Conference on Software Maintenance*, Bethesda, MD, USA, pp. 368–377.

Beygelzimer, A., S. Kakade, et J. Langford (2006). Cover trees for nearest neighbor. In *ACM International Conference Proceeding Series, Proceedings of the 23rd international conference on Machine learning*, Volume 148, Pittsburgh, Pennsylvania, pp. 97–104. ACM Press.

Chevalier, F., D. Auber, et A. Telea (2007a). Structural analysis and visualization of c++ code evolution using syntax trees. In *9th International Workshop on Principles of Software Evolution (IWPSE'07)*.

Chevalier, F., M. Delest, et J. Domenger (2007b). A heuristic for the retrieval of objects in video in the framework of the rough indexing paradigm. *Signal Processing : Image Communication on Content-Based Multimedia Indexing (SPIC) 22*, 622–634.

Demirci, M. F., A. Shokoufandeh, Y. Keselman, L. Bretzner, et S. Dickinson (2006). Object recognition as many-to-many feature matching. *International Journal of Computer Vision 69*(2), 203–222.

Don, A. (2006). *Indexation et navigation dans les contenus visuels : approches basÃl'es sur les graphes*. Ph. D. thesis, University of Bordeaux 1.

Fekete, J. et C. Plaisant (2003). Infovis contest 2003 - visualization and pair wise comparison of trees. In IEEE (Ed.), *IEEE Symposium on Information Visualization*, Volume www.cs.umd.edu/hcil/iv03contest/.

Gomila, C. et F. Meyer (2003). Graph-based object tracking. In *IEEE International Conference on Image Processing (ICIP)*, Volume 2, pp. 41–44.

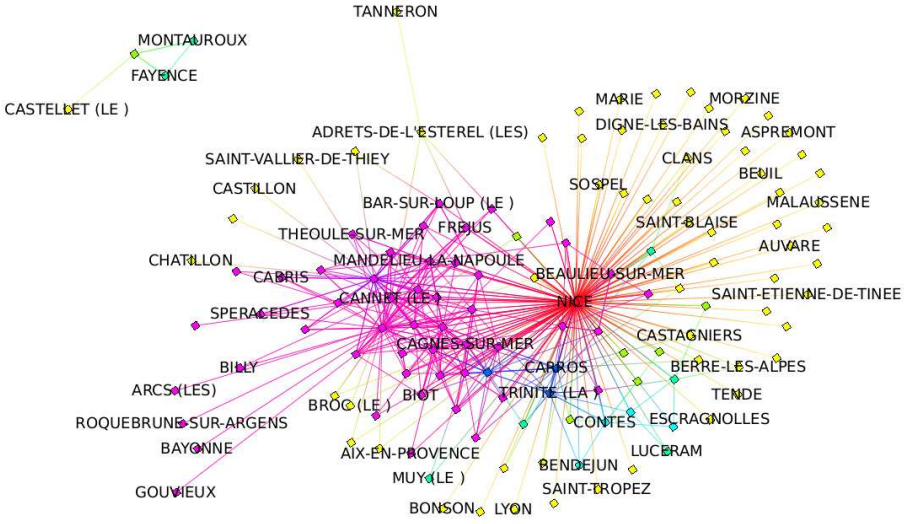
Lladós, J., E. Martí, et J. Villanueva (2001). Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*(10), 1137–1143.

Yan, X., F. Zhu, J. Han, et P. Yu (2006). Searching substructures with superimposed distance. In *ICDE '06. Proceedings of the 22nd International Conference on Data Engineering*, pp. 88–97.

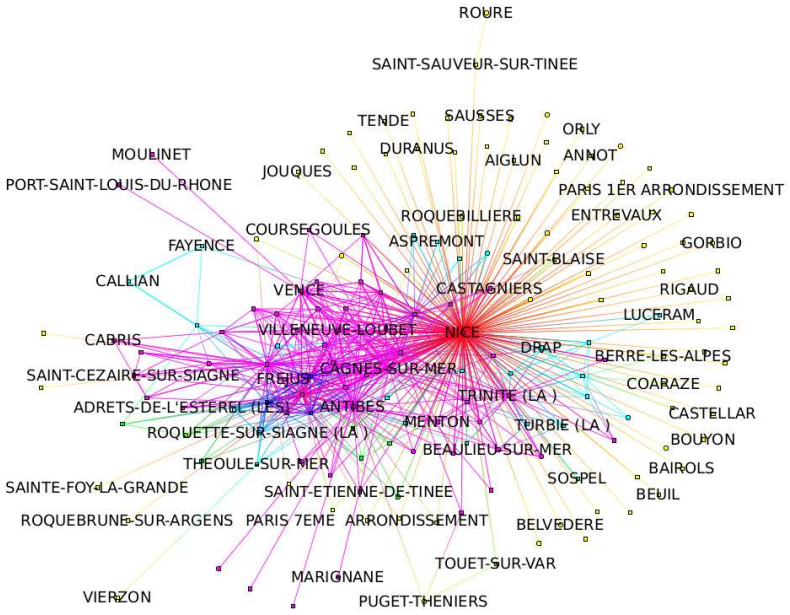
Zachary, W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research 33*, 452–473.

Summary

Detecting similar patterns in graphs is known to be a difficult task. We tackle this problem by designing a heuristic based on intrinsic metrics for graphs, looking for patterns where metric values are coupled with topology. Our algorithm allows us to detect similar patterns common to a series of graphs. We give applications to geographical data and to feature detection in video data.

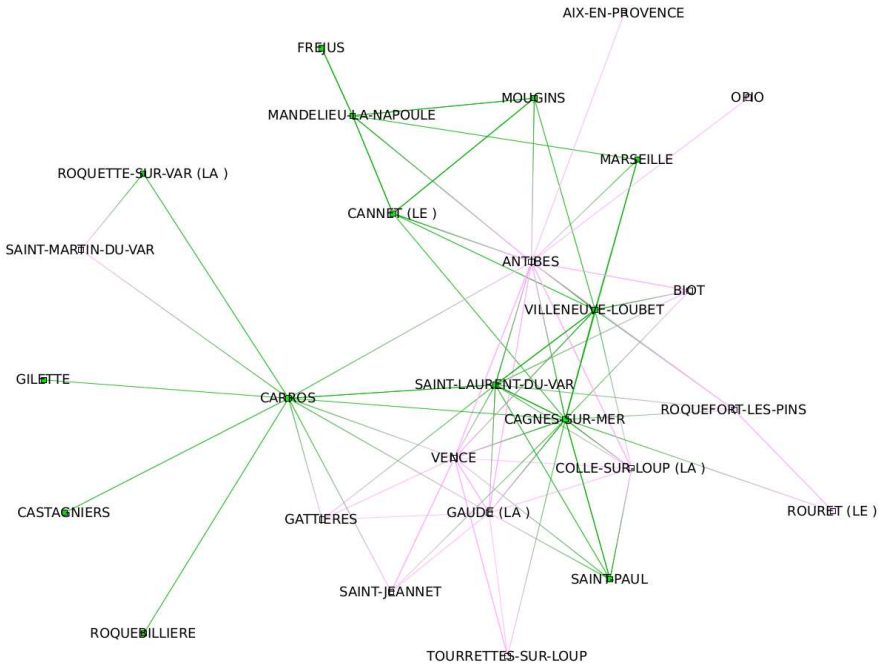


(a) 1975

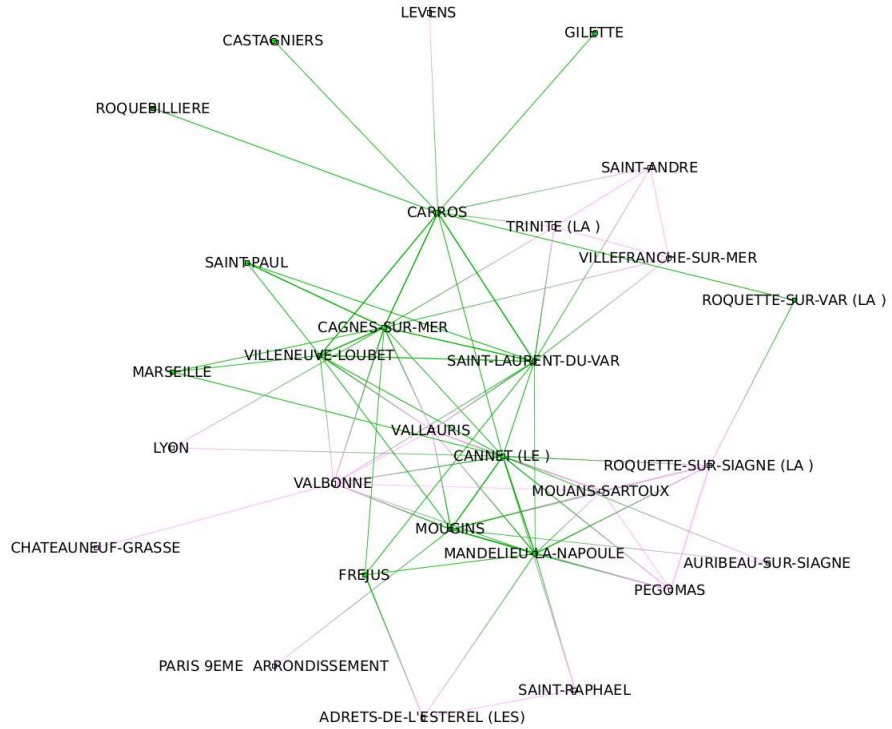


(a) 1982

FIG. 6 – *Réseaux de migrations professionnelles : reconnaissance de motifs par propagation d'étiquette. Les motifs similaires sont représentés en foncé (dans la version couleur, une même couleur représente un motif quasi-similaire).*



(a) 1975



(a) 1982

FIG. 7 – *Réseaux de migrations professionnelles : un motif quasi-similaire. Les sommets communs aux deux graphes sont représentés en foncé (en vert dans la version couleur).*