La simulation relaxée de graphes pour la recherche de motifs

IEEE/ACM ASONAM 2018, August 28-31, 2018, Barcelona, Spain

Abdelmalek Habi*, Brice Effantin*, Hamamache Kheddouci*

**Univ Lyon, Université Claude Bernard Lyon 1, CNRS, LIRIS, F-69622*
Lyon, France
{abdelmalek.habi, brice.effantin-dit-toussaint, hamamache.kheddouci}@univ-lyon1.fr

Résumé. La recherche de motifs de graphe est l'une des opérations principales de la recherche des correspondances d'une requête dans un graphe donné. Dans ce contexte, trouver des solutions garantissant l'optimalité en termes de précision et de temps de calcul est un problème de recherche difficile et d'actualité. Différents modèles ainsi que leurs algorithmes appropriés ont été proposés pour la recherche de motifs dans les graphes de données. Cependant, l'inconvénient majeur est leur limitation à trouver des réponses significatives entraînant le problème des réponses vides. Dans cet article nous introduisons un nouveau modèle pour la recherche de motifs de graphe permettant un certain type d'assouplissement de requêtes afin d'éviter ce problème. Ensuite nous développons un algorithme efficace basé sur des techniques d'optimisation pour trouver les k-meilleurs réponses selon notre modèle. Nos expérimentations sur quatre ensembles de données réelles démontrent l'efficacité de notre approche.

1 Introduction

Les graphes sont des structures mathématiques constituant un outil de modélisation et de représentation universel utilisé dans une large gamme d'applications réelles. La recherche de motifs de graphes (*RMG*) est l'une des opérations fondamentales sur laquelle reposent la recherche et l'analyse des graphes de données. Soit $G(V, E, l, \Sigma)$ un graphe de données et $Q(V_q, E_q, f_v)$ un motif de graphe (requête) où : V (V_q), E (E_q), l (f_v) et Σ représentent respectivement l'ensemble de nœuds, l'ensembles des arêtes, la fonction des étiquettes et l'univers des étiquettes dans le graphe (la requête). Le problème *RMG* consiste à trouver toutes les correspondances de Q dans G, notées par $M(Q, G)$. Typiquement, ce problème est défini en termes de :

— l'isomorphisme de sous-graphes (Ullmann, 1976) : $M(Q, G)$ est constitué de tous les sous-graphes G' de G auxquels Q est isomorphe, *i.e.,* il existe une fonction bijective $h : V_q \longmapsto V$ telle que $(u, u') \in Q$ si et seulement si $(h(u), h(u')) \in G'$; ou

— la simulation de graphes (Henzinger et al., 1995) : $M(Q, G)$ est une relation binaire $R \subseteq Vq \times V$ qui vérifie : (1) $\forall u \in V_q$, $\exists v \in V \mid (u, v) \in R$, et (2) $\forall (u, v) \in R$ et $\forall (u, u') \in Q$, $\exists (v, v') \in G \mid (u', v') \in R$.

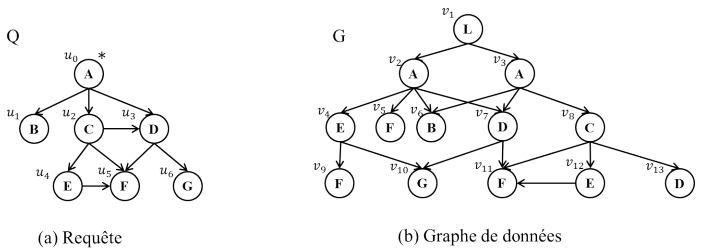


FIG. 1: Un graphe de données et un graphe requête

Avec la taille grandissante des graphes de données, le nombre des correspondances d'une requête peut être excessivement important. Inspecter tous les résultats est une tâche ardue, en plus du fait que les utilisateurs ne portent d'intérêt qu'aux meilleures réponses. De plus, dans plusieurs applications, les algorithmes d'appariement utilisent des requêtes ciblées qui visent à trouver des correspondances d'un nœud de sortie au lieu de l'appariement entier (Fan et al., 2013). Ce genre d'applications ne cherche pas les correspondances exactes, et pour cette raison plusieurs approches à base de simulation de graphes ont été proposées. Malgré les relaxations posées par la simulation de graphes et ses variantes d'une part, et le fait qu'il est presque impossible de connaitre la structure de graphe d'une autre part, nous avons constaté que ces approches sont aussi restrictives car elles n'acceptent pas l'appariement avec des nœuds manquants (sans affecter la qualité des résultats). Dans plusieurs applications réelles, ce type de relaxation est très utile car il permet d'éviter le problème de réponses vides.

Ceux-ci mettent en évidence le besoin de trouver les k-meilleures (top-k) réponses d'un nœud de sortie en permettant la relaxation en termes de nœuds manquants.

Exemple : Un réseau de collaboration est représenté par G dans la figure 1b. Dans ce graphe, un nœud v_i représente une personne avec son travail (étiquette du nœud) et une arête (v_i, v_j) indique une relation de supervision. Exemple (v_3, v_7) indique que la personne v_3 avec le travail A est le superviseur de la personne v_7 avec le travail D. Une compagnie émet la requête Q (figure 1a) pour trouver des correspondances dans G. Dans cet exemple, u_0^* est le nœud de sortie de Q. Cela signifie que seules les correspondances de ce nœud sont demandées.

Dans cet exemple, l'isomorphisme de sous-graphes ne parvient pas à identifier des correspondances pour la requête Q. Avec la simulation de graphes, on peut vérifier que $M(Q, G) = \{(u_0, v_3), (u_1, v_6), (u_2, v_8), (u_3, v_7), (u_4, v_{12}), (u_5, v_{11}), (u_6, v_{10}), (u_4, v_4), (u_5, v_9)\}$ est le résultat d'appariement de Q dans G. Cependant, le résultat d'une recherche utilisant la requête ciblée ne contient que le nœud v_3. En outre, on peut vérifier que, dans un tel cas, le nœud v_2 peut être considéré comme un résultat potentiel, car il est possible que le superviseur d'une personne occupant le poste C puisse également être un superviseur des personnes supervisées par C. Mais la simulation classique ne parvient pas à identifier ce type de relation.

Les approches à base de la simulation relaxée de graphes peuvent bénéficier d'une notion plus générale de la simulation de graphes. Mais elles risquent de perdre de leur efficacité en terme de temps de recherche.

Dans cet article, nous étudions les défis ci-dessus et nous proposons un nouveau modèle,

appelé *la simulation relaxée de graphes* (*SRG*), afin d'éviter le problème des réponses vides. Nous proposons des algorithmes qui permettent de trouver les top-k réponses tout en réduisant le coût de la recherche grâce à des techniques d'optimisation. De plus, nous menons des expérimentations approfondies pour attester l'efficacité de l'approche proposée.

2 État de l'art

Plusieurs travaux ont été proposés pour la recherche de motifs de graphes. Ce problème a été traité par l'isomorphisme de sous-graphes et la simulation de graphes.

L'isomorphisme de sous-graphes consiste à énumérer toutes les occurrences exactes d'une requête dans un graphe. Ce problème est NP-complet et il est largement étudié dans la littérature. Récemment, plusieurs travaux ont été proposés pour faire face aux limites de l'isomorphisme de sous-graphes. (Zhang et al., 2010) ont étudié l'appariement approximatif en utilisant une distance d'édition bornée par la requête. La simulation de graphe présente une alternative efficace à l'isomorphisme de sous-graphes en permettant un certains nombre d'assouplissements sur les correspondances. (Fan et al., 2010) introduisent un modèle de simulation bornée en terme de nombre de sauts dans le graphe. Le modèle, *simulation forte*, proposé par (Ma et al., 2014) étend la simulation classique en imposant deux conditions supplémentaires (dualité et localité). Une autre étude récente, réalisée par (Gao et al., 2016), étend la simulation de graphes en permettant l'absence des nœuds à un saut. Cependant, elle perd la notion de simulation et affecte la qualité des résultats pour les requêtes avec des nœuds feuilles. (Li et al., 2017) proposent la combinaison de la taxonomie des étiquettes avec la simulation de graphes.

Le problème de top-k a beaucoup été étudié pour toutes les représentations de données. (Ilyas et al., 2008) présentent un état de l'art de ce problème dans les systèmes de bases de données relationnelles. Ce problème a été également étudié pour les requêtes XML, (Guo et al., 2003), et les graphes de données, (Fan et al., 2013).

3 La simulation relaxée de graphes

Dans cette section nous présentons notre modèle qui permet d'éviter le problème des réponses vides et nous décrivons notre approche pour la recherche des motifs.

Bien que les résultats des approches à base de simulation de graphes soient intéressants en terme de temps de recherche, nous avons constaté qu'elles sont incapables de capturer des correspondances significatives ce qui entraine un certain nombre de réponses vides. Nous avons donc proposé un nouveau modèle, appelé la *simulation relaxée de graphes* (*SRG*), qui permet d'identifier des correspondances avec un certain type de relaxation.

SRG est basé sur la notion d'*ensemble de satisfaction*. Un ensemble de satisfaction d'un nœud $u \in Q$ (Sat_u) est un ensemble d'ensembles d'étiquettes qui permet de vérifier si un nœud $v \in G$ correspond au nœud u. Selon le besoin dans *SRG*, l'ensemble de satisfaction peut être calculé comme suit : *(1)* le premier ensemble dans Sat_u contient les étiquettes des fils du nœud u, *(2)* les autres ensembles dans Sat_u sont construits à partir de toutes les combinaisons possibles en remplaçant chaque élément dans le premier ensemble par les étiquettes des fils du nœud correspondant, s'ils existent.

La simulation relaxée de graphes. Soient $G = (V, E, l, \Sigma)$ un graphe de données et $Q = (V_q, E_q, f_v)$ un motif de graphe. Le graphe G correspond au motif Q par ***SRG*** s'il existe une relation binaire $R \subseteq Vq \times V$ qui vérifie : Si $(u, v) \in R$ alors : *(1)* $l(u) = f_v(v)$ et *(2)* $\exists S_{u_i} \subseteq Sat_u$, tel que $S_{u_i} \subseteq L_v$, où L_v est la liste des étiquettes des fils de v.

En général, les graphes de données sont très grands, ce qui rend l'ensemble de résultats excessivement large. En revanche, l'intérêt de l'utilisateur peut être résumé ou regroupé dans les top-k réponses du nœud souhaité u_* (Ilyas et al., 2008). Dans ce but, nous avons défini une fonction de classement basée sur les ensembles de satisfaction où l'importance d'un nœud $v \in G$ est donnée par le score $\gamma(S_{u_i})$ de l'ensemble de satisfaction couvert par v :

$$\gamma(S_{u_i}) = deg(u) - \alpha \cdot b_i.$$

avec $0 \leq \alpha \leq 1$ représente un facteur de pénalité et $0 \leq b_i \leq deg(u)$ (degré du nœud u) représente le nombre de substitutions dans $S_{u_i} \subseteq Sat_u$, ***i.e.,*** le nombre des nœuds manquants remplacés par leurs nœuds fils. Ensuite, nous utilisons $\gamma()$ pour calculer la pertinence $\delta()$ d'un nœud v_* qui correspond au nœud de sortie u_* :

$$\delta(v_*) = \frac{\sum\limits_{S_i \subseteq S_{best}} \gamma(S_i)}{|E_q|}.$$

avec S_{best} représente les meilleurs ensembles de satisfaction couverts lors du processus de recherche des correspondances, et $|E_q|$ représente le nombre des arêtes dans la requête Q.

La recherche à base de ***SRG*** permet d'avoir plus de résultats en évitant le problème des réponses vides. Ce gain en qualité influe sur l'efficacité en terme de temps de recherche. Pour cette raison, nous avons conçu des algorithmes qui visent à trouver les top-k réponses et réduire le coût de recherche. Notre approche se déroule en deux phases. La première vise à encoder les informations de voisinage des nœuds d'un graphe en utilisant la structure de données probabiliste ***"filtre de Cuckoo"*** (Fan et al., 2014). Cette dernière est très efficace pour les tests d'appartenance à un ensemble et elle permet de réduire de façon significative la complexité en temps. La seconde phase vise à trouver les k meilleures correspondances selon le modèle ***SRG***. Pour qu'un nœud $v \in G$ corresponde à un nœud $u \in Q$, les trois conditions suivantes doivent être vérifiées : *(1)* $l(u) = f_v(v)$, *(2)* $deg(v) \geq deg(u)$ et *(3)* la liste des étiquettes des fils de v couvre au moins un ensemble parmi les ensembles de Sat_u. Nous utilisons le filtre de Cuckoo pour vérifier la dernière condition. L'algorithme de recherche choisit des correspondances potentielles du nœud de sortie u_* et effectue une vérification de correspondance des nœuds de sous graphe induit par le nœud choisi. le résultat final est une liste des k meilleures correspondances trouvées (voir Habi et al., 2018).

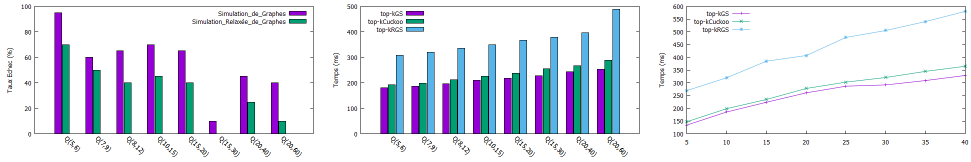
L'analyse de la complexité en temps de notre approche a donné $O(|V|.D+|V|.(|V|+|E|))$.

4 Résultats

Dans cette section, nous décrivons et discutons les résultats expérimentaux afin d'évaluer nos méthodes.

Nos expérimentations portent sur quatre graphes (*Epinions, Amazon, Google, et LiveJournal*) issus de *Stanford Large Network Dataset Collection* [1] (nous ne présenterons ici que les

1. http ://snap.stanford.edu



(a) Taux d'échec (LiveJournal) (b) Temps (Variant $|Q|$ - LiveJournal) (c) Temps (Variant k - LiveJournal)

FIG. 2: Évaluation des performances

résultats du graphe *LiveJournal*, les autres étant similaires). Nous considérons deux critères d'évaluation. Le premier est le taux d'échec qui représente le rapport entre le nombre de réponses vides et le nombre de requêtes utilisées. Ce critère permet d'évaluer l'efficacité des algorithmes. Le deuxième critère est l'efficacité en terme de temps de recherche.

Trois séries d'expérimentations sont menées : la première est la simulation de graphes améliorée par la terminaison anticipée (*top-kGS*) (Fan et al., 2013), la deuxième est la simulation relaxée de graphes (*top-kRGS*), et finalement la simulation relaxée de graphe améliorée par l'utilisation du filtre de Cuckoo (*top-kCuckoo*).

4.1 Discussions

Taux d'échec : La figure 2a montre les résultats du taux d'échec des deux modèles (simulation de graphes et simulation relaxée de graphes). Dans cette expérience, nous avons fixé $k = 10$ et nous avons fait varier $(|V_q|, |E_q|)$ de $(5, 6)$ à $(20, 60)$. Nous observons que la simulation relaxée réduit efficacement le taux d'échec (dans toutes les expérimentations).

Temps de recherche : Les figures 2b et 2c montrent le temps moyen de recherche en fonction respectivement de la taille de la requête et de k. Les résultats montrent que les trois algorithmes sont sensibles à la variation de k. En outre, *top-kGS* et *top-kCuckoo* surperforment toujours *top-kRGS*. *Top-kCuckoo* et *top-kGS* ont un temps de recherche presque similaire pour les requêtes de petites tailles mais *top-kCuckoo* prend plus de temps pour les grandes requêtes, ce qui s'explique par le fait qu'il identifie plus de correspondances.

Ces expérimentations montrent les performances de notre approche en terme de qualité avec un temps de recherche quasi similaire au *top-kGS*.

5 Conclusion

Dans cet article, nous avons abordé le problème de réponse vide dans le contexte de la recherche de motifs de graphes. Nous avons proposé un nouveau modèle appelé la simulation relaxée de graphes (*SRG*) basé sur la simulation de graphes. Notre modèle permet de prendre en compte les nœuds manquants sans affecter la qualité des résultats et de fournir une bonne flexibilité pour plusieurs applications. En outre, nous avons également développé un algorithme efficace utilisant le filtre de Cuckoo pour calculer les k meilleures réponses. Par conséquent, notre approche convient très bien aux grands graphes. Nos expérimentations valident l'efficacité de cette approche.

Références

Fan, B., D. G. Andersen, M. Kaminsky, et M. D. Mitzenmacher (2014). Cuckoo filter : Practically better than bloom. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pp. 75–88. ACM.

Fan, W., J. Li, S. Ma, N. Tang, Y. Wu, et Y. Wu (2010). Graph pattern matching : from intractable to polynomial time. *Proceedings of the VLDB Endowment 3*(1-2), 264–275.

Fan, W., X. Wang, et Y. Wu (2013). Diversified top-k graph pattern matching. *Proceedings of the VLDB Endowment 6*(13), 1510–1521.

Gao, J., P. Liu, X. Kang, L. Zhang, et J. Wang (2016). Prs : Parallel relaxation simulation for massive graphs. *The Computer Journal 59*(6), 848–860.

Guo, L., F. Shao, C. Botev, et J. Shanmugasundaram (2003). Xrank : Ranked keyword search over xml documents. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pp. 16–27. ACM.

Habi, A., B. Effantin, et H. Kheddouci (2018). Fast top-k search with relaxed graph simulation. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 495–502. IEEE.

Henzinger, M. R., T. A. Henzinger, et P. W. Kopke (1995). Computing simulations on finite and infinite graphs. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pp. 453–462. IEEE.

Ilyas, I. F., G. Beskales, et M. A. Soliman (2008). A survey of top-k query processing techniques in relational database systems. *ACM Computing Surveys (CSUR) 40*(4), 11.

Li, J., Y. Cao, et S. Ma (2017). Relaxing graph pattern matching with explanations. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1677–1686. ACM.

Ma, S., Y. Cao, W. Fan, J. Huai, et T. Wo (2014). Strong simulation : Capturing topology in graph pattern matching. *ACM Transactions on Database Systems (TODS) 39*(1), 4.

Ullmann, J. R. (1976). An algorithm for subgraph isomorphism. *Journal of the ACM (JACM) 23*(1), 31–42.

Zhang, S., J. Yang, et W. Jin (2010). Sapper : subgraph indexing and approximate matching in large graphs. *Proceedings of the VLDB Endowment 3*(1-2), 1185–1194.

Summary

Graph pattern matching has been widely used in large spectrum of real applications. In this context, different models along with their appropriate algorithms have been proposed. However, a major drawback on existing models is their limitation to find meaningful matches resulting in a number of failing queries. In this paper we introduce a new model for graph pattern matching allowing the relaxation of queries in order to avoid the empty-answer problem. Then we develop an efficient algorithm based on optimization strategies for computing the top-k matches according to our model. Our experimental evaluation on four real datasets demonstrates both the effectiveness and the efficiency of our approach.