DDMCE : recherche de cliques maximales dans des graphes dynamiques de grande taille

Ovidiu Şerban[*,**], **Alexandre Pauchet**[*], **Alexandrina Rogozan**[*], **Jean-Pierre Pécuchet**[*]

[*] *Laboratoire LITIS - EA 4108, INSA de Rouen, Avenue de l'Université - BP 8*
76801 Saint-Étienne-du-Rouvray Cedex, France
nom.prenom@insa-rouen.fr
[**] *Faculty of Mathematics and Computer Science, "Babeş-Bolyai" University*
1, Mihail Kogălniceanu St., RO-400084 Cluj-Napoca, Romania

RÉSUMÉ. *L'objectif de cet article est de présenter un algorithme de découverte de cliques maximales permettant l'exploration de grands graphes dynamiques. Un ensemble d'expérimentations a été conduit sur des graphes de petite taille mais très denses et sur des graphes très grands mais creux, tous générés aléatoirement. Notre algorithme DDMCE (Dynamic Distributable Maximal Clique Exploration), y obtient des résultats encourageants laissant penser qu'il supporte bien le passage à l'échelle.*

ABSTRACT. *Our main objective is to design an algorithm for maximal clique exploration that scales well with large and dynamic data. We tested multiple experimental setups: small data with low and high density, and very sparse large data, all randomly generated. An experiment was conducted with dynamic generated data. Our proposition, the Dynamic Distributable Maximal Clique Exploration (DDMCE) Algorithm, looks very promising in all these experiments because it scales well considering data size and number of processors involved in the computation.*

MOTS-CLÉS : *Recherche de cliques maximales, Grands graphes, Graphes dynamiques*

KEYWORDS: *Maximal clique exploration, Large graphs, Dynamic graphs*

1. Introduction

Une clique est un sous-graphe complet, *i.e.* un sous-graphe pour lequel tous les noeuds sont interconnectés. Elle est dite maximale lorsqu'elle est le plus grand sous-graphe complet contenant tous ses noeuds. La suite de cet article se concentre sur la recherche de cliques maximales (MCE : Maximal Clique Exploration) dans un graphe. La MCE concerne le plus souvent deux types de graphe particuliers : 1) des graphes de petite taille (contenant moins de 1 000 noeuds) mais très denses (densité supérieure à 0,75 (75%)) ; la plupart des graphes issus du second challenge DIMACS ([JOH 96]), appartiennent à cette catégorie. 2) des graphes de grande taille (contenant plus de 1 000 noeuds), très creux (densité inférieure à 0,01 (1%)) ([MOH 10]) ; il s'agit la plupart du temps de graphes "réels", ayant une signification sémantique (ex : graphes de réseaux sociaux). Par ailleurs, pour de nombreux graphes comme ceux représentant des réseaux sociaux, le caractère dynamique (ajouts et suppressions de noeuds et d'arcs), nécessite d'être traité.

La plupart des algorithmes de MCE existants ont été conçus pour des graphes de petite taille et supportent mal le passage à l'échelle pour des graphes de grande taille ([MOH 10]). Malgré certains progrès récents dans la gestion de graphes de grande taille, comme par exemple le système Google Pregel ([MAL 10]), le problème de la MCE n'est toujours pas résolu. De plus, le caractère dynamique reste peu abordé à ce jour pour la recherche de cliques maximales ([STI 04]).

L'objectif de cet article est de présenter DDMCE (Dynamic Distributable Maximal Clique Exploration), un algorithme permettant de rechercher les cliques maximales dans un graphe de grande taille et dynamique, c'est-à-dire pour lequel la structure du graphe peut être modifiée durant le fonctionnement de l'algorithme.

2. Représentation des solutions

Contrairement aux approches proposées par ([BRO 73, TOM 06, CAZ 08]) basées sur des ensembles, nous avons adopté une représentation sous forme d'arbre pour stocker l'ensemble des cliques maximales découvertes lors de l'exploration du graphe. Cette représentation maintient un historique des cliques détectées et permet ainsi de ne pas recommencer la recherche au début en cas de modification dynamique du graphe. Dans cette représentation, chaque chemin depuis la racine jusqu'à une feuille représente une clique maximale en encodant l'exploration (voir figure 1). DDMCE initialise la construction de cet arbre avec comme racine le noeud abstrait $\{*\}$.

Chaque étape de l'exploration se fait à partir d'un élément racine appelé pivot. Le pivot est un élément central de DDMCE, comme pour de nombreux autres algorithmes de découverte de cliques ([CAZ 08]). Par ses propriétés locales, il permet de réduire la taille de l'arbre d'exploration. DDMCE inclut comme stratégie de sélection de pivot celle proposée par [TOM 06] : à chaque étape du processus d'exploration, l'élément sélectionné comme pivot est celui avec le plus grand voisinage. Ce choix permet de ré-

duire significativement l'ensemble des noeuds encore à explorer. La figure 1 présente pour chaque noeud le pivot correspondant encadré en rouge.

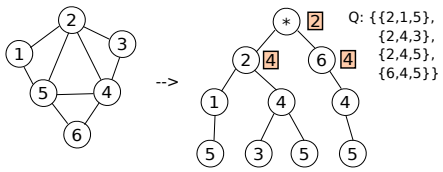
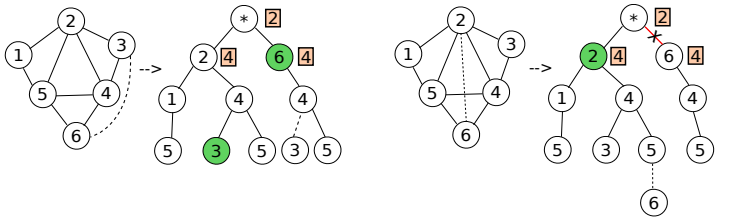


Figure 1 – Décomposition du graphe sous forme d'un arbre d'exploration. Q est l'ensemble des cliques maximales découvertes. Pour chaque étape les pivots sont encadrés en rouge.

Les figures 2a et 2b illustrent le caractère dynamique du problème : lorsque le graphe à explorer est modifié, la représentation sous forme d'arbre des solutions est mise à jour. Lors de l'ajout ou de la suppression d'un arc, tous les sous-arbres affectés par ce changement doivent être recherchés et modifiés. Si un pivot est impliqué dans la modification, un sous-arbre complet est alors supprimé et exploré à nouveau à partir du pivot. Figure 2a, un arc est ajouté entre les noeuds $\{2, 6\}$, n'entraînant que l'ajout d'une feuille à l'arbre d'exploration. Figure 2b, ajouter un arc entre les noeuds $\{2, 6\}$, alors que $\{2\}$ est pivot pour le noeud racine, conduit à supprimer le sous-arbre du noeud $\{6\}$ au premier niveau. À l'ajout d'un noeud, dans le cas le moins favorable, l'ensemble des cliques maximales découvertes peut devoir être recalculé.



(a) Modification dynamique du graphe : ajout d'un arc

(b) Modification du graphe : ajout d'un arc (influence du pivot)

Figure 2 – Modification du graphe entraînant une mise à jour de l'arbre d'exploration. Les noeuds verts sont réactivés comme racines des sous-arbres à recalculer.

DDMCE est implémenté avec une architecture à base de messages afin de permettre de distribuer l'exploration. La liste des cliques découvertes reste cependant centralisée pour éviter les problèmes de synchronisation. Afin de limiter l'espace mémoire nécessaire au stockage des solutions, un système de cache est utilisé (fichier texte ou base de données).

3. Expérimentations

L'évaluation des performances de DDMCE a été effectuée sur un serveur de type Xeon Intel (4 coeurs, 1.6 Ghz/coeur), sur Linux Ubuntu64. La JVM utilisée est Oracle Java, version 1.7.0 pour Linux64. Le générateur de graphes aléatoires est un générateur d'arcs basé sur un échantillonnage linéaire permettant de créer un voisinage de noeuds, prenant en paramètre le nombre de noeuds à générer et la densité. Il construit un graphe uniformément distribué (2% de variance pour la densité). Le même algorithme a été utilisé pour générer dynamiquement de nouveaux noeuds et de nouveaux arcs, en conservant la densité ρ du graphe initial.

La table 1 présente les temps de traitement en s. obtenus avec DDMCE, en situation dynamique et statique. Les résultats présentés dans le cas statique sont les temps d'exploration des graphes dans certaines configurations (n est le nombre de noeuds, ρ la densité et c le nombre de coeurs). Dans le cas dynamique, le résultat présente le temps pour trouver les cliques après l'ajout d'un noeud. Le ratio du temps de calcul obtenu avec notre approche dynamique par rapport à une approche statique est précisé. Ce rapport est largement inférieur à 1,00 pour des graphes de grandes tailles (10 000) et avec une densité allant de 0,01 (1%) à 0,30 (30%).

n	ρ	c	Static	Dynamic	Dynamic/Static	n	ρ	c	Static	Dynamic	Dynamic/Static
1,000	.01	1	.342	.029	.084	5,000	.01	1	1.687	.125	.074
	.01	2	.250	.021	.086		.01	2	1.266	.099	.078
	.01	4	.240	.022	.092		.01	4	1.324	.100	.076
	.03	1	.598	.061	.102		.03	1	9.010	.541	.060
	.03	2	.557	.043	.077		.03	2	5.673	.449	.079
	.03	4	.543	.043	.078		.03	4	5.363	.432	.081
	.05	1	.776	.119	.154		.05	1	30.805	1.267	.041
	.05	2	.836	.098	.117		.05	2	18.585	1.103	.059
	.05	4	.996	.101	.101		.05	4	17.828	1.102	.062
	.07	1	.992	.210	.212		.07	1	90.080	3.854	.043
	.07	2	.886	.164	.185		.07	2	54.499	2.804	.051
	.07	4	1.071	.160	.149		.07	4	52.718	2.746	.052
	.10	1	1.617	.483	.298		.10	1	408.919	19.324	.047
	.10	2	1.215	.388	.319		.10	2	245.741	11.875	.048
	.10	4	1.433	.365	.255		.10	4	235.444	11.580	.049
	.15	1	4.683	1.037	.221	10,000	.01	1	13.114	.292	.022
	.15	2	3.217	.906	.281		.01	2	9.522	.228	.024
	.15	4	3.266	1.015	.311		.01	4	9.246	.194	.021
	.20	1	16.819	2.360	.140		.03	1	99.625	1.788	.018
	.20	2	9.527	1.852	.194		.03	2	62.055	1.405	.023
	.20	4	9.781	2.069	.211		.03	4	57.786	1.593	.028
	.25	1	67.404	8.501	.126		.05	1	467.644	11.138	.024
	.25	2	37.192	5.381	.145		.05	2	286.653	7.247	.025
	.25	4	35.882	5.665	.158		.05	4	269.613	6.687	.025
	.30	1	281.932	40.488	.144		.07	1	1,768.328	59.519	.034
	.30	2	160.485	22.151	.138		.07	2	1,075.273	37.237	.035
	.30	4	149.339	21.711	.145		.07	4	1,015.318	34.676	.034
							.10	1	10,550.762	491.325	.047
							.10	2	6,379.845	301.977	.047
							.10	4	6,068.725	282.626	.047

Tableau 1 – Temps de traitement de DDMCE (en s) pour des graphes statiques et dynamiques. n représente le nombre de noeuds, ρ la densité et c le nombre de coeurs utilisés pour le traitement. $Dynamic/Static$ représente le rapport des temps de traitement en dynamique et en statique.

DDMCE présente un gain de performance en contexte dynamique. Après l'ajout d'un noeud, le calcul dynamique des cliques ne prend que 20 % du temps (comparé

à l'approche statique) pour des petits graphes. Ce rapport descend même à 5 % sur des données de grande dimension. En moyenne, l'approche dynamique est 9 fois plus rapide que l'approche statique.

4. Conclusion

DDMCE est un algorithme original conçu pour la recherche de cliques maximales dans des graphes dynamiques de grande taille. La représentation sous forme d'arbres de l'exploration du graphe adoptée dans DDMCE lui permet : 1) d'être robuste aux changements dynamiques, 2) de réduire l'espace d'exploration, et 3) de pouvoir réutiliser des résultats précédents même en cas d'erreur, en ne recalculant que les noeuds erronés.

Cette étude s'est concentrée sur la comparaison entre le comportement statique et dynamique de notre algorithme DDMCE. La suite de ces travaux consiste à démontrer ces résultats et à comparer notre algorithme avec ceux de la littérature, notamment dans le cas dynamique.

5. Bibliographie

[BRO 73] BRON C., KERBOSCH J., « Algorithm 457 : finding all cliques of an undirected graph », *Communications of the ACM*, vol. 16, n° 9, 1973, p. 575–577, ACM.

[CAZ 08] CAZALS F., KARANDE C., « A note on the problem of reporting maximal cliques », *Theoretical Computer Science*, vol. 407, n° 1-3, 2008, p. 564–568, Elsevier.

[JOH 96] JOHNSON D., TRICK M., *Cliques, coloring, and satisfiability : second DIMACS implementation challenge, October 11-13, 1993*, vol. 26, Amer Mathematical Society, 1996.

[MAL 10] MALEWICZ G., AUSTERN M., BIK A., DEHNERT J., HORN I., LEISER N., CZAJKOWSKI G., « Pregel : a system for large-scale graph processing », *Proceedings of the 2010 international conference on Management of data*, ACM, 2010, p. 135–146.

[MOH 10] MOHAISEN A., YUN A., KIM Y., « Measuring the mixing time of social graphs », *Proceedings of the 10th annual conference on Internet measurement*, ACM, 2010, p. 383–389.

[STI 04] STIX V., « Finding all maximal cliques in dynamic graphs », *Computational Optimization and applications*, vol. 27, n° 2, 2004, p. 173–186, Springer.

[TOM 06] TOMITA E., TANAKA A., TAKAHASHI H., « The worst-case time complexity for generating all maximal cliques and computational experiments », *Theoretical Computer Science*, vol. 363, n° 1, 2006, p. 28–42, Elsevier.