UNIVERSITÉ Clermont Auvergne

CLERMONT AUVERGNE INP — ÉCOLE DOCTORALE DES SCIENCES POUR L'INGÉNIEUR

LIMOS – MAAD - AGC

Directeur de thèse :Vincent Limouzy; vincent.limouzy@uca.fr ; Professeur – HDR
Co-encadrant : Pierre Bergé ; pierre.berge@uca.fr ; Maître de conférences

Titre du sujet de thèse : Algorithmes efficaces d'énumérations de cliques maximales potentielles dans des graphes

Résumé du sujet de thèse :

Contexte

Dans notre contexte, résoudre un problème d'énumération signifie trouver toutes les solutions possibles au problème. Le but est donc de trouver un algorithme efficace qui liste toutes les solutions exactement une fois. Par exemple, trouver tous les éléments d'une base de données qui correspondent à une requête est un problème d'énumération, ou également trouver toutes les cliques maximales dans un graphe. Cependant, le nombre de solutions peut être exponentiel en la taille de l'entrée, par exemple il est prouvé que le nombre de cliques maximales dans un graphe à n sommets peut être 3^{n/3}. Ainsi, la notion classique de complexité n'est plus pertinente dans notre contexte. Pour tenir compte de ce contexte particulier, Johnson et ses co-auteurs Johnson & Papadimitriou '88 ont introduit une nouvelle notion de complexité, nous exprimons ici la complexité dans la taille de l'entrée plus la taille de la sortie, avec cette définition, un algorithme qui est polynomial en termes de taille de l'entrée et de la sortie est appelé **output polynomial**.

Dans ce cas, il se pourrait que toutes les solutions soient produites à la toute fin de l'exécution de l'algorithme. En conséquence, nous attendrions un temps exponentiel dans la taille de l'entrée avant de produire une première solution, tout en obtenant un algorithme polynomial de sortie. C'est pourquoi la notion d'algorithme à **délai polynomial** a été naturellement introduite. Ici le but est de pouvoir trouver des solutions régulièrement, et le délai entre la production de deux solutions doit être au plus polynomial en la taille de l'entrée. Pour cette dernière notion, il n'est pas toujours possible d'obtenir une telle complexité.

Objectifs

De nombreux problèmes d'énumération ont été étudiés sur des graphes et d'autres structures binaires. Un aperçu peut être consulté dans le compendium de Wasa '16. Mais concernant la décomposition des graphes, de nombreux résultats restent encore à trouver. Les décompositions de graphes s'avèrent très utiles lorsqu'il faut concevoir des algorithmes efficaces pour résoudre un problème combinatoire sur des graphes. Malheureusement, la plupart des décompositions intéressantes ne peuvent pas être obtenues en temps polynomial (à moins que P=NP), au mieux pour certaines d'entre elles, les algorithmes FPT sont connus. La décomposition arborescente et son paramètre associé, la largeur d'arbre, jouent un rôle important dans la théorie algorithmique des graphes. Depuis trente ans, beaucoup d'énergie a été consacrée à trouver des algorithmes efficaces pour obtenir une bonne décomposition des arbres. Parmi les techniques développées tout au long du parcours, quelques concepts clés ont été introduits. L'une d'elles, appelée **cliques maximales potentielles**, s'avère très utile et intéressante.

Une *clique maximale potentielle* d'un graphe G est un sous-ensemble de sommets X tel qu'il existe une complétion cordale minimale H où le graphe induit par X dans H est une clique. Ce concept a été introduit par Bouchitté et Todinca '02 afin de calculer efficacement la largeur arborescente d'un graphe. Le nombre de cliques maximales potentielles pourrait être exponentiel en la taille du graphe. Dans certaines classes de graphes, ils ont réussi à prouver que leur nombre pouvait être polynomial. Dans leur article fondateur, ils ont développé un algorithme polynomial de sortie pour trouver toutes les cliques maximales potentielles. La complexité de leur algorithme affiche une complexité temporelle $O(N^2)$ où N désigne le nombre de solutions. Cette notion de cliques maximales est de plus en plus populaire et désormais largement utilisée pour obtenir des algorithmes en temps polynomial pour certains problèmes combinatoires. L'obtention d'un algorithme avec un temps d'exécution amélioré diminuerait automatiquement la complexité de nombreux algorithmes de graphes, car cette partie constitue le goulot d'étranglement de ces algorithmes. Dans le meilleur des cas on pourrait espérer obtenir un algorithme en O(N). Cette question reste ouverte depuis vingt ans. Afin d'attaquer ce problème, nous considérerons des classes de graphes spéciales. Parmi les techniques d'énumération que nous pourrions utiliser pour résoudre ce problème se trouve la technique nouvellement développée appelée **Proximity search with canonical path** appliquée avec succès pour obtenir un retard polynomial et un algorithme d'espace polynomial Brosse et al. '22.