UNIVERSITÉ Clermont Auvergne

CLERMONT AUVERGNE INP

ÉCOLE DOCTORALE DES SCIENCES POUR L'INGÉNIEUR

LIMOS – MAAD - AGC

Thesis supervisor: Vincent Limouzy; vincent.limouzy@uca.fr ; Professor
Co-supervisor : Pierre Bergé ; pierre.berge@uca.fr ; Associate Professor

Title of PhD subject: Efficient enumeration of potential maximal cliques in a graph.

Summary

Context

In our context, solving an enumeration problem means finding all feasible solutions of the problem. Thus the goal is to find an efficient algorithm which lists all the solutions exactly once. For instance, finding all the items in some database that match a query is an enumeration problem, or also finding all the maximal cliques in a graph. However, the number of solutions might be exponential in the size of the input, for instance it is proven that the number of maximal cliques in a n-vertex graph can be 3^{n/3}. Thus the classical notion of complexity is no longer relevant in our setting. To account this particular context, Johnson and his co-authors Johnson & Papadimitriou '88 introduced a novel notion of complexity, here we express the complexity in the size of the input plus the size of the output, with that definition, an algorithm that is polynomial in both the size of the input and the output is called **output polynomial**.

In that setting, it could be the case that all the solutions are produced at the very end of the execution of the algorithm. As a consequence we would wait a time exponential in the size of the input before a first solution is produced, but still obtain an output polynomial algorithm. Therefore the notion a **polynomial delay** algorithms was naturally introduced. Here the goal is to be able to find solutions regularly, and the delay between the production of two solutions should be at most polynomial in the size of the input. For the latter notion, it is not always possible to obtain such a complexity.

Goals

A lot of enumeration problems were studied on graphs and other binary structures. An overview can be seen in compendium paper by Wasa '16. But, concerning graph decomposition, many results have yet to be found. Graph decompositions turn out to be very useful when one needs to design efficient algorithms to solve a combinatorial problem on graphs. Unfortunately most of the interesting decompositions cannot be obtained in polynomial time (unless P=P), at best for some of them FPT algorithms are known.The tree decomposition and its associated parameter **treewidth** play an important role in algorithmic graph theory. For the last thirty years a lot of energy has been devoted to find efficient algorithms to obtain a good tree decomposition. Among the techniques developed along the way, some key concepts were introduced. One of them, called **potential maximal cliques** turns out to be very useful and interesting.

A *potential maximal clique* of graph G is a subset of vertices X such that there exist a minimal chordal completion H where the graph induced by X in H is a clique. This concept was introduced by *Bouchitté and*
Todinca '02 in order to compute efficiently the treewidth of a graph. The number of potential maximal cliques could be exponential in the size of the graph. In some classes of graph, they managed to prove that their number could be polynomial. In their seminal paper they developed an output polynomial algorithm to find all the potential maximal cliques. The complexity of their algorithm displays an $O(N^2)$ time complexity where N denotes the number of solutions. This notion of maximal cliques is more and more popular and now widely used to obtain polynomial time algorithms for some combinatorial problems
Obtaining an algorithm with an improved running time would automatically decrease the complexity of many graph algorithms, as this part is the bottleneck of the algorithm. In the best case one could

hope to obtain an O(N) algorithm. This question has been open for the last twenty years. In order to attack this problem we will consider special graph classes. Among the enumeration techniques we could use to solve this problem is the newly developed technique called **Proximity search with canonical path** successfully applied to obtain a polynomial delay and polynomial space algorithm Brosse et al. '22.