Thesis abstract

Mohammed Amin Tahraoui

In this thesis we invetsigate some problems in graph theory, namely the graph coloring problem, the graph packing problem and tree pattern matching for XML query processing. The common point between these problems is that they use labeled graphs.

In the first part, we study a new coloring parameter of graphs called the *gap vertex-distinguishing edge coloring*. It consists in an edge-coloring of a graph G which induces a vertex distinguishing labeling of G such that the label of each vertex is given by the difference between the highest and the lowest colors of its adjacent edges. The minimum number of colors required for a gap vertex-distinguishing edge coloring of G is called the *gap chromatic number* of G and is denoted by $gap(G)$. We will compute this parameter for a large set of graphs G of order n and we even prove that $gap(G) \in \{n-1, n, n+1\}$.

In the second part, we focus on graph packing problems, which is an area of graph theory that has grown significantly over the past several years. However, the majority of existing works focuses on unlabeled graphs. In this thesis, we introduce for the first time the packing problem for a vertex labeled graph. Roughly speaking, it consists of graph packing which preserves the labels of the vertices. We study the corresponding optimization parameter on several classes of graphs, as well as finding general bounds and characterizations.

The last part deal with the query processing of a core subset of XML query languages: XML twig queries. An XML twig query, represented as a small query tree, is essentially a complex selection on the structure of an XML document. Matching a twig query means finding all the occurrences of the query tree embedded in the XML data tree. Many holistic twig join algorithms have been proposed to match XML twig pattern. Most of these algorithms find twig pattern matching in two steps. In the first one, a query tree is decomposed into smaller pieces, and solutions against these pieces are found. In the second step, all of these partial solutions are joined together to generate the final solutions. In this part, we propose a novel holistic twig join algorithm, called *TwigStack++*, which features two main improvements in the decomposition and matching phase. The proposed solutions are shown to be efficient and scalable, and should be helpful for the future research on efficient query processing in a large XML database.