Improvements to the $\mathcal{O}^*(2^k)$ path finding algorithm for colored graphs

Joren Verspeurt

March 2014

1 Introduction

For this assignment I examined the randomized simple k-path algorithm proposed by Ryan Williams in [Williams,2009]. The assignment was to find possible optimizations for colored graphs, specifically the cases where the path to be found consists of $c * k'$ nodes with c the amount of colors and k' the amount of nodes per color and the case where some colors have only 1 node.

2 Trivial improvements

As a vertex with a certain color can only be mapped to a vertex of the same color there will be zero-elements in the matrix implied by the dynamic programming algorithm for the evaluation of the polynomials. More specifically with c colors there will be only 1/c non-zero elements in the implicit matrix and this will reduce the amount of work even for an unmodified version of the algorithm. This is, however, not a significant improvement. It does reduce the amount of operations but not the size of each operation, dependent on the size of the elements of $\mathbb{Z}_2^k$ which is the determining factor in this case. The complexity of the algorithm remains $\mathcal{O}^*(2^k)$.

3 Nodes evenly distributed among c colors

There are multiple more specific cases that are interesting to consider:

- The case where colors only appear in a strict sequence and the path is dividable in sections that contain only 1 color.
- The case where colors do not appear in a strict sequence but still only appear in a limited region in the path.
- The general case where colors an appear anywhere.

For the first case the following optimization can be made: the problem can be divided into smaller subproblems. In general the full problem can be split into c subproblems (one per color) and $c - 1$ "interface" problems. The subproblems per color only consider the part of the graph in that color and a path of length k'. The interface problems consider only paths consisting of 2 vertices in 2 neighboring colors from the original path and the part of the graph that consists of vertices of either color that have edges to vertices of the other color (this does not need to be computed explicitly). If the result of a subproblem is that a path can't be found then that result counts for the full problem. If one or more paths exist for a subproblem, information can be retained about which vertices can be used as endpoints for a path of that color and this information can be used in the interface problem, which then determines which vertices of the next color can be used as endpoints for a path of that color. This results in a further reduction of the complexity per subproblem. Each of the subproblems then has a complexity of (at most) $\mathcal{O}^*(2^{k/c})$.

Similarly for the case where the colors appear in a limited region in the path this division can be performed but there the complexity of the interface problems can begin to dominate and negate the benefits of the subdivision.

4 Some colors only have 1 vertex in the path

For this case a subdivision is also possible: the problem can be split around the unique vertices. The interface problem there consists of finding a path with 3 vertices. The complexity of the full problem can then be reduced (at least) to $\mathcal{O}^*(2^{k-c'})$.

5 Time information

The "Trivial Improvements" section was completed before the first contact moment, the other sections where inspired by that contact moment.