MID-NOVEMBER STATUS REPORT

Peter Tatkowski, Esref Özdemir, Sebastian Leisinger, Tobi

Department of Computer Science
ETH Zürich
Zürich, Switzerland

1. SUBGRAPH ISOMORPHISM

Subgraph Isomorphism is a problem that asks: Given a data graph G and a query graph q, how many isomorphisms of q can be found in G? This is a known NP-Hard problem, but several papers have tried to give several heuristics to solve this problem faster in general situations. Our group is studying three of these algorithms: Ullman [**?**], TurboISO [**?**], and VF3 [**?**]. There exist sequential versions of Ullman and TurboISO, but as of right now, there are no parallel implementations that were found.

Our project is going to consist of benchmarking and parallelizing Ullman, TurboISO and VF3, and comparing different parallelization techniques. On top of this, we want to standardize the library that each of these algorithms are using, so we are intergrating these algorithms into the GAPBS Framework. The work up to now has consisted of finding sequential implementations of these algorithms, intergrating them into the framework, and running some simple benchmarking on them.

2. WORK DIVISION

Here is how the work was divided between the project members:

Peter.

1. Implementing the existing implementation of TurboISO [**?**] into the GAPBS Framework
   (a) Major code refactoring on the found TurboISO implementation
   (b) Integrating the graph structures into GAPBS interface
2. Creating benchmark suites so that the benchmarks can be ran more easily.

Esref.

1. Implementing the existing implementation of TurboISO into the GAPBS Framework
   (a) Major code refactoring on the found TurboISO implementation
   (b) Integrating the graph structures into GAPBS interface
2. Creating test suites for the transformed algorithms including TurboISO code to ensure correctness
3. Researching standard datasets to use for benchmarking and transforming them into a standard format.

Sebastian.

1. Implementing existing implementation of the Ullman algorithm [**?**] into the GAPBS Framework
   (a) Code refactoring to integrate Ullman algorithm with the Graph datastructure of GAPBS
   (b) Some refactoring of the algorithm to avoid stack-overflow on large graphs.

Tobi.

1. Implementing existing implementation of the VF3 algorithm [**?**] into the GAPBS Framework
   (a) Code refactoring to integrate the existing sequential [**?**] and parallel [**?**] VF3 algorithm with the Graph datastructure of GAPBS

3. NEXT STEPS

- Create parallel versions of Ullman, TurboISO and extend benchmark suite to scaling.
- The parallel version of VF3 leaves a lot of room for improvement. We plan to refactor it and improve performance.

4. INITIAL RESULTS

We show initial results of triangle finding in complete graphs on different algorithms, where we tested performance by increasing the number of nodes in the underlying data graph.