Abstract of the Dissertation

Analyses and Design of Efficient Graph Partitioning Methods

Robert Preis

Graph partitioning is a standard problem in graph theory with a wide range of applications. This thesis made some progress in the analyzes of the bisection width of regular graphs and designed provable good algorithms for the coarsening and the local improvement phase of the multilevel graph partitioning approach.

The work in this thesis contributed new upper bounds on the bisection width of 3- and 4-regular graphs. Only a small gap to known lower bounds of certain 3- and 4-regular graphs remains. Furthermore, we showed new lower bounds on the bisection width of 3- and 4-regular Ramanujan graphs. These lower bounds are the highest lower bounds for explicitly constructible 3- and 4-regular graphs. The different lower bounds are compared for some graph classes. Furthermore, small regular graphs with the highest bisection width are derived experimentally.

The multilevel graph partitioning paradigm has been proven to be a very powerful approach to efficient graph-partitioning. However, the quality of the used algorithms have only been proven on an experimental basis. In this thesis, we developed efficient methods to be used in the multilevel context. For the coarsening part we developed a new approximation algorithm for maximum weighted matching in general edge-weighted graphs. The algorithm calculates a matching with an edge weight of at least $\frac{1}{2}$ of the edge weight of a maximum weighted matching in linear time. For the local improvement we use the Helpful-Set method which is derived from a constructive proof of upper bounds on the bisection width of regular graphs. Overall, the combination of analytical methods for the two parts of the multilevel approach lead to an efficient graph-partitioning concept.

The algorithms designed in this thesis were implemented in the PARTY graph partitioning library. It provides efficient implementations of many different methods and offers a flexible and universal interface, in order to be easily integratable in applications.