Overview
In computer science, engineering, and related fields, graph partitioning is a common technique. For example, in parallel computing good partitionings of unstructured graphs are very valuable. In this area, graph partitioning is mostly used to partition the underlying graph model of computation and communication. Roughly speaking, nodes in this graph denote computation units, and edges represent communication. This graph needs to be partitioned such that there are few edges between the blocks (pieces). In particular, if we want to use $k$ processors we want to partition the graph into $k$ blocks of about equal size. Most graph partitioners try to minimize the total cut size, i.e. the number of edges that run between blocks.

It is, however, well known that there are more realistic objective functions such as the total communication volume or the maximum communication volume:

$$MCV(V_1, \ldots, V_k) := \max_p \sum_{v \in V_p} |\{V_i | \exists \{u, v\} \in E, u \in V_i \neq V_p\}|.$$

Tasks
Existing algorithms neglect the maximum communication volume metric, i.e. FM-based local search algorithms mostly optimize the number edges cut by a partition. There are algorithms that can combine many different local searches by employing a negative cycle detection algorithm. The main task of this thesis is to improve the maximum communication volume of partitions. Therefore the student should engineer new local search algorithms. In particular, heuristic algorithms that optimize the maximum communication volume efficiently even if the graphs has to be partitioned into more than two blocks while keeping other objective functions at a given threshold.

Requirements
- Interest in algorithms and data structures
- Good programming skills in C++
- Basic parallel programming skills

Application deadline 15th April 2015