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: In the past, this was fine since most of the networks that needed partitioning had structures in which cut size and communication volume are highly correlated. With a shifting focus towards partitioning social networks, this is no longer the case. Hence, we directly want to optimize the “right” objective function like maximum communication volume:

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

Other applications, like the preprocessing technique in customizable route planning for shortest-path computations, need to minimize even more complicated objective functions in order to work well.

Tasks

The main task of this thesis is to engineer local search algorithms for other objective functions such as the ones listed above. In the best case, we derive algorithms that only need a definition of a local gain function.

Requirements

- Interest in algorithms and data structures
- Good programming skills in C++
- Ability to think for yourself

Application deadline 31th October 2016