

Bachelor-/Masterthesis

Algorithm Configuration / Algorithm Selection

Overview

Algorithms for hard computational problems, for example based on local search are often highly parameterized. Typical parameters in local search include neighbourhoods, tabu tenure, percentage of random walk steps, and perturbation and acceptance criteria in iterated local search.

Automated procedures for solving this algorithm configuration problem are useful in a variety of contexts. Their most prominent use case is to optimize parameters on a training set of instances from some application ("offline" as part of algorithm development) in order to improve performance when using the algorithm in practice ("online"). Algorithm configuration thus trades human time for machine time and automates a task that would otherwise be performed manually.

The algorithm configuration problem can be formally stated as follows: given a parameterized algorithm \mathcal{A} (the target algorithm), a set (or distribution) of problem instances \mathcal{I} and a cost metric c , find parameter settings of \mathcal{A} that minimize c on \mathcal{I} . The cost metric c is often based on the runtime required to solve a problem instance, or, in the case of optimization problems, on the solution quality achieved within a given time budget.

$$\theta^* \in \arg \min_{\theta \in \Theta} c(\theta)$$

Tasks

Existing algorithms neglect that many algorithms, e.g. local search algorithms, can trade solution quality for running time. The main task is to modify existing algorithms so that Pareto optimal configurations in the parameter space are derived instead of a single configuration that only looks at one objective. This has to be done on different levels of granularity. Advanced algorithms compute a performance model and select good algorithm configurations based on instance features and a choice of the user. The graph partitioning tool KaHIP will be used as a case study.

Requirements

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

Application deadline 15th April 2015

