

Abstract:

Many industrial problems in various fields, such as logistics, process management, or product design, can be formalized and expressed as optimization problems in order to make them solvable by optimization algorithms. However, solvers that guarantee the finding of optimal solutions (complete) can in practice be unacceptably slow. This is one of the reasons why approximative (incomplete) algorithms, producing near-optimal solutions under restrictions (most dominant time), are of vital importance. Those approximative algorithms go under the umbrella term metaheuristics, each of which is more or less suitable for particular optimization problems. These algorithms are flexible solvers that only require a representation for solutions and an evaluation function when searching the solution space for optimality.

What all metaheuristics have in common is that their search is guided by certain control parameters. These parameters have to be manually set by the user and are generally problem and inter-dependent: A setting producing near-optimal results for one problem is likely to perform worse for another. Automating the parameter setting process in a sophisticated, computationally cheap, and statistically reliable way is challenging and a significant amount of attention in the artificial intelligence and operational research communities. This activity has not yet produced any major breakthroughs concerning the utilization of problem instance knowledge or the employment of dynamic algorithm configuration.

The thesis promotes automated parameter optimization with reference to the inverse impact of problem instance diversity on the quality of parameter settings with respect to instance-algorithm pairs. It further emphasizes the similarities between static and dynamic algorithm configuration and related problems in order to show how they relate to each other. It further proposes two frameworks for instance-based algorithm configuration and evaluates the experimental results. The first is a recommender system for static configurations, combining experimental design and machine learning. The second framework can be used for static or dynamic configuration, taking advantage of the iterative nature of population-based algorithms, which is a very important sub-class of metaheuristics.

A straightforward implementation of framework one did not result in the expected improvements, supposedly because of pre-stabilization issues. The second approach shows competitive results in the scenario when compared to a state-of-the-art model-free configurator, reducing the training time by in excess of two orders of magnitude.