

AUTOMATIC DESIGN OF ALGORITHMS

Thomas Stützle, Manuel López-Ibáñez

Most optimization algorithms, including evolutionary algorithms and metaheuristics, and general-purpose solvers for integer or constraint programming, have often many parameters that need to be properly designed and tuned for obtaining the best results on a particular problem. Automatic (offline) algorithm design methods help algorithm users to determine the parameter settings that optimize the performance of the algorithm before the algorithm is actually deployed. Moreover, automatic offline algorithm design methods may potentially lead to a paradigm shift in algorithm design because they enable algorithm designers to explore much larger design spaces than by traditional trial-and-error and experimental design procedures. Thus, algorithm designers can focus on inventing new algorithmic components, combine them in flexible algorithm frameworks, and let final algorithm design decisions be taken by automatic algorithm design techniques for specific application contexts.

This tutorial is structured into two main parts. In the first part, we will give an overview of the algorithm design and tuning problem, review recent methods for automatic algorithm design, and illustrate the potential of these techniques using recent, notable applications from the presenters' and other researchers work. In the second part of the tutorial will focus on a detailed discussion of more complex scenarios, including multi-objective problems, anytime algorithms, heterogeneous problem instances, and the automatic generation of algorithms from algorithm frameworks. The focus of this second part of the tutorial is, hence, on practical but challenging applications of automatic algorithm design. The second part of the tutorial will demonstrate how to tackle algorithm design tasks using our irace software (<http://iridia.ulb.ac.be/irace>), which implements the iterated racing procedure for automatic algorithm design. We will provide a practical step-by-step guide on using irace for the typical algorithm design scenario.