An Approach to the Automatic Configuration of a Generalized Metaheuristic Structures

Marie-Eléonore Marmion, Franco Mascia
Manuel López-Ibáñez, Thomas Stützle
IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgique
{marie-eleonore.marmion;fmascia;manuel.lopez-ibanez;stuetzle}@ulb.ac.be

1 Motivation

We propose a generalized metaheuristics structure that unifies different metaheuristics such as iterated local search, simulated annealing, variable neighborhood search, iterated greedy and greedy randomized adaptive search procedures. From this metaheuristic structure, we can instantiate each of the above mentioned metaheuristics, but we also can generate novel combinations of the algorithmic components of those metaheuristics and embed metaheuristics within others. In fact, the possibility of embedding metaheuristics into others (or itself) leads to the ability to recursively call one metaheuristic within another one.

This generalized metaheuristics structure is implemented through a context-free grammar. The grammar is composed of a set of production rules and each production rule can be seen as encoding the set of possibilities to instantiate one specific component of the metaheuristics [4]. With the production rules, we can therefore encode discrete alternative choices for an algorithmic component. As an example, we can choose the type of modification (say, neighborhood) in the perturbation of an iterated local search algorithm. But we also can define through this grammar, the numerical values of parameters, for example, the strength of a perturbation. Thus, from the grammar we can derive structurally different stochastic local search algorithms that are fully instantiated, that is, where all numerical parameters have defined values. The grammar defines a large search space that is the set of the possible configurations. We then use an automatic algorithm configuration tool to instantiate algorithms from the grammar.

2 Framework

The grammar is implemented in Python and gives the set of possible algorithm configurations. From this grammar, we can instantiate a specific configuration in our generalized metaheuristic structure that is output in the form of a program that can be compiled with ParadisEO. ParadisEO is a C++ white-box framework dedicated to the reusable design of metaheuristics [1]. ParadisEO is based on a conceptual separation of the problem modeling and the solving methods. A large set of implemented classes is available that represents components of stochastic local search [2] such as hill climber methods, simulated annealing and iterated local search. When solving a problem, only the instance parser, the objective function and the neighborhood are to be implemented to use ParadisEO. The iterated racing procedure from the Irace package [3] is applied as an automatic algorithm configuration tool. This procedure aims at configuring the algorithmic components and setting the numerical parameters in order to obtain an efficient algorithm.

3 Experimental Results

Designing an efficient metaheuristic for a problem is a hard and time-consuming task. Many greedy heuristics are found for the classical problems of the literature. Here, we apply our framework to two classical NP-hard problems to prove the interest of the proposed framework. The first one is a bit string problem called unconstrained binary quadratic programming. Various heuristics are known and used to solve this problem [5]. The second problem is the permutation flowshop scheduling problem with weighted tardiness. The iterated racing procedure will be used to give for each problem the best configuration. Then, the performance of the best configuration is compared to the state-of-the-art algorithm implemented in the framework. Experiments are on going and results will be presented at the conference.

References

[1] Sébastien Cahon, Nouredine Melab, and El-Ghazali Talbi. Paradiseo: A framework for the reusable design of parallel and distributed metaheuristics. *Journal of Heuristics*, 10:357–380, 2004.

[2] Jérémie Humeau, Arnaud Liefooghe, El-Ghazali Talbi, and Sébastien Verel. ParadisEO-MO: From Fitness Landscape Analysis to Efficient Local Search Algorithms. Rapport de recherche RR-7871, INRIA, 2012.

[3] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Thomas Stützle, and Mauro Birattari. The irace package, iterated race for automatic algorithm configuration. (TR/IRIDIA/2011-004), 2011.

[4] Franco Mascia, Manuel López-Ibáñez, Jérémie Dubois-Lacoste, and Thomas Stützle. From grammars to parameters: Automatic iterated greedy design for the permutation flow-shop problem with weighted tardiness. In *accepted for publication in the post-conference proceedings of LION 7*, 2013.

[5] Peter Merz and Bernd Freisleben. Greedy and local search heuristics for unconstrained binary quadratic programming. *Journal of Heuristics*, 8:197–213, 2002.