Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in DISCRETE MATHEMATICS
Electronic Notes in Discrete Mathematics 58 (2017) 167–174 www.elsevier.com/locate/endm

Exploring variable neighborhood search for automatic algorithm configuration

Leslie Pérez Cáceres [1] Thomas Stützle [2]

IRIDIA, Université Libre de Bruxelles,
CP 194/6, Av. F. Roosevelt 50, B-1050, Belgium

Abstract

Methods for automatic algorithm configuration integrate some search mechanism for generating candidate algorithm configurations and mechanisms for handling the stochasticity of the algorithm configuration problem. One popular algorithm configurator is ParamILS, which searches the configuration space using an iterated local search algorithm. In our research, we explore variable neighborhood search mechanisms as an alternative for the one-exchange neighborhood that is searched in the local search phase of ParamILS. In this article, we explore a reduced variable neighborhood search for automatic configuration. Our experimental results are promising and indicate directions for extending our work.

Keywords: Variable neighborhood search, automatic algorithm configuration, ParamILS

1 Introduction

The performance of exact and (meta)heuristic algorithms does strongly depend on the settings given to their parameters and typically these must be

[1] Email: `leslie.perez.caceres@ulb.ac.be`
[2] Email: `stuetzle@ulb.ac.be`

http://dx.doi.org/10.1016/j.endm.2017.03.022
1571-0653/

adapted to the problem at hand. Algorithm configuration is the problem of finding performance-optimizing parameter settings for search algorithms (target algorithm). Automatic algorithm configuration aims at finding high-performance parameter settings by means of another algorithm, conveniently in that context called configurator, that search the parameter space of the target algorithm to identify settings that perform best for an available set of training instances for the problem being tackled [2]. The impact of automatic algorithm configuration is actually larger than just finding the setting of specific parameters of a fully developed algorithm. It has the potential to transform the way exact and heuristic optimization algorithms are designed [3,9]. This is the case as algorithm design decisions can often be conveniently encoded by categorical or ordinal parameters and algorithm configurators such as ParamILS [5], irace [7] or SMAC [4] are able to handle these efficiently.

In this article, we focus on extending ParamILS [5], which is a configurator based on iterated local search (ILS) [8]. ParamILS requires all algorithm parameters to be defined as categorical parameters. This is simply done by discretizing numerical parameters, that is, integer parameters (such as tabu list length or population sizes) and continuous parameters (such as mutation rates or initial temperature) and treating the discrete values without specific ordering between them. ParamILS starts the search from the best configuration between a user specified default configuration and $n_{initial}$ randomly generated candidate configurations. It follows an iterative improvement process that searches a one-exchange neighborhood, where configurations differ in the value given to exactly one parameter. Once a locally optimal configuration is determined, the main loop of the ILS is entered consisting of an initial perturbation step that changes randomly the values given to three parameters, an execution of the iterative improvement process and an acceptance criterion that decides from which configuration to continue the search process. ParamILS implements a method that increases the number of instances to which the best configuration is applied (a way to handle the stochasticity of the configuration problem). Most applications of ParamILS are to configuration tasks where the goal is to minimize the computation time taken by an algorithm, e.g. to find an optimal solution and prove its optimality. For this task, ParamILS is using an *adaptive capping* mechanism, which bounds the execution time of configurations according to the observed performance of the current best configuration. Adaptive capping performs some form of early pruning of poor performing configurations. For more details about ParamILS we refer to [5].

We study the integration of some ideas from variable neighborhood search