Engineering Applications of Artificial Intelligence 76 (2018) 108–118

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Tuners review: How crucial are set-up values to find effective parameter values?

Elizabeth Montero *, María-Cristina Riff, Nicolás Rojas-Morales

Department of Computer Science, Universidad Técnica Federico Santa María, Avenida España 1680, Valparaíso, Chile

ARTICLE INFO

Keywords:
Parameter setting
Tuning methods
Parameter search space definition

ABSTRACT

ParamILS, I-Race and Evoca are well-known tuning methods designed to search quality parameter calibrations for metaheuristic algorithms. The set-up of parameter search space can strongly affect the performance of tuning methods. In this work we study how the parameter search definitions affect the quality of parameter calibrations delivered by these tuners. An experimental evaluation using two well known metaheuristic algorithms and a real life case is presented. We also provide some guidelines to consider when defining parameters search spaces according to the tuner used in order to obtain the best performance they can find.

1. Introduction

Tuning methods have shown to be effective strategies to improve metaheuristics performance finding appropriate values for their parameters. A set of values for each parameter a metaheuristic defines is usually called a parameter calibration. A typical genetic algorithm usually defines a set of at least three parameters: population size, crossover rate and mutation rate. Moreover, most times these parameters are coupled with some design decisions that drive the designer to define proper selection, mutation and crossover operators. In fact, parameter values can be divided into two categories: categorical and numerical. Categorical parameters are those that have a finite domain with no distance metric or ordering between values, while numerical parameters are those whose domains are subsets of $\mathbb{N}$ or $\mathbb{R}$. Popular recommendations to set numerical values can be found in literature for genetic algorithms (De Jong, 1975; Grefenstette, 1986), but setting categorical parameters can become a harder task that requires the definition of proper "values" for these parameters.

Moreover, when tuning a categorical parameter, most tuning methods search all possible values for the parameter. On the other hand, when tuning a numerical parameter, a finite number of values belonging to its range of values is selected to perform the tuning. The selection of these sets of values is an experience-dependent process that can strongly affect the quality of the tuning process. We consider as quality/performance of a tuning process the quality of solutions obtained by the tuned metaheuristic algorithm when using the calibration obtained by the tuning process and the computational effort involved on the tuning process. In this work we study the performance of three well-known tuning methods considering complex tuning scenarios in order to analyze the influence of parameter search space definitions.

Several tuning methods have been proposed during the last years. These methods can be categorized in four main areas (Eiben and Smit, 2011): Sampling methods, model-based methods, screening methods and meta-evolutionary algorithms. Section 2 gives an overview of state-of-the-art tuning techniques and algorithms. In this work, we focus our attention on three well-known tuners: ParamILS (Hutter et al., 2009), I-Race (Birattari et al., 2010) and Evoca (Riff and Montero, 2013). ParamILS corresponds to an iterated local search algorithm that starting from a default parameter calibration searches its neighborhood looking for parameter calibrations of better quality. The process is iterated until a fixed stopping criterion is met. I-Race follows the framework of iterative screening methods. It constructs candidate calibrations based on a probability model, evaluates the most promising ones and updates the probability model to bias the next sample. Evoca works with a population of parameter calibrations that undergo selection, crossover and mutation operations to improve their quality.

Related to the definition of parameter search spaces ParamILS requires the definition of a discrete set of values for each parameter being tuned in order to describe finite neighborhoods. I-Race also requires a finite set of possible values for each categorical parameter being tuned and a range of values for each numerical parameter. Moreover, I-Race can also use a user-defined set of parameter calibrations to sample its first iteration. Evoca requires the definition of a set of possible values for each categorical parameter and a range of values for each numerical

* Corresponding author.
E-mail address: elizabeth.montero@usm.cl (E. Montero).

https://doi.org/10.1016/j.engappai.2018.09.001
Received 12 February 2018; Received in revised form 4 August 2018; Accepted 6 September 2018
0952-1976/

parameter being tuned. These tuning methods are described in detail in Section 3.

We have realized that using ParamILS is not an easy task for users/designers because they are asked to explicitly define the parameter search space. This definition requires, in most cases, the selection of a small subset of values each numerical parameter can take. For categorical parameters all possible values are usually considered. On the other hand, the performance of I-Race has shown being strongly dependent of a proper definition of the initial setup of parameter space used. A complex initial setup may discourage some users who find it difficult to select adequate settings for these methods. Unlike these algorithms, Evoca works well with simpler definitions of parameter search spaces. We deepen the motivation of our research in Section 4. Moreover, a performance analysis of ParamILS (FocusedILS), I-Race and Evoca tuners using different setup of parameters space is presented in Section 5. We analyze the difference in the quality of parameter calibrations obtained using various experimental setups. Section 6 presents a summary of main recommendations on the definition of parameter spaces for tuning methods including the analysis of a real-life case. Moreover, we also included some preliminary results on collaborative tuning scenarios between the tuning methods studied. Our conclusions and future work are presented in Section 7.

2. Related work

Tuning methods have shown being effective strategies to find good quality parameter calibrations for metaheuristics algorithms. Several tuning methods have being proposed, starting with Grefenstette' Meta-GA (Grefenstette, 1986) until nowadays where four categories of tuners can be identified (Eiben and Smit, 2011): Sampling methods, model-based methods, screening methods and meta-evolutionary algorithms.

Sampling methods reduce the search effort by cutting the number of parameter calibrations evaluated with respect to a full factorial design. Most sampling methods are used as starting points for model-based methods or as initialization methods. Calibra (Adenso-Diaz and Laguna, 2006) is an example of *iterative* sampling method. After each step, *iterative* sampling methods refine the parameter search area from which new parameter calibrations will be sampled.

Model-based methods construct models of utility landscapes of parameter search spaces. In these methods the total number of experiments is reduced by replacing some of the "real" executions by estimations of the proposed model. SPO (Bartz-Beielstein et al., 2012) is a well-known *iterative* model-based method. It performs a multi-stage procedure, that iteratively generates a set of new parameter calibrations and predicts their quality using the proposed model. Best performing calibrations are tested to prove their quality and then used to update the current model. SMAC (Hutter et al., 2011) method is an improved model-based tuning method. It uses a specially designed comparison mechanism and random forests to model response surfaces. It is able to work with categorical and numerical parameters, and also, with multiple problem instances that show different features.

Screening methods try to identify the best parameter calibrations from a given set performing a fixed number of tests/executions. *Iterative* screening methods, determine a subset of parameter calibrations that deserve further investigation at each step. F-Race (Birattari et al., 2002) is the most known screening tuning method. I-Race (Balaprakash et al., 2007) is an extension of F-Race that at each iteration starts an F-Race procedure with a sampling of parameter calibrations in a large region of the parameter search space. At each step, I-Race uses a multi-variate normal distribution to fit the F-Race surviving parameter calibrations. These calibrations are then used to sample calibrations for a new iteration. This screening and generating procedure is repeated until the computational budget for tuning is reached.

Search-based methods face the tuning problem as an optimization problem, hence a good strategy to find high quality calibrations can be to use a metaheuristic algorithm. ParamILS (Hutter et al., 2009) is an iterated local search algorithm that works searching neighborhoods of parameter calibrations. A version of ParamILS (FocusedILS) uses a comparison method able to increase the number of executions for comparing parameter calibrations when required. FocusedILS version of ParamILS was used in our experiments. Evoca (Riff and Montero, 2013) is an evolutionary-based algorithm that works with a population of calibrations. In Evoca, the population size is computed considering the number of parameters and their domain sizes. The key idea is to include a set of well distributed values for each parameter on its first population. Evoca implements two transformation operators: a wheel crossover and a hill climbing first improvement mutation operator.

Tuning methods have become an important tool in metaheuristic research area. Their use has been extended to metaheuristic design, i.e., to determine not only the value of numerical parameters of the metaheuristic, but also, of its categorical parameters. In Montero and Riff (2014a) authors use I-Race and Evoca to select the transformation operators of a standard genetic algorithm and the mutation operator of a multi-objective immune-based algorithm. In Bezerra et al. (2014) authors use a large set of both numerical and categorical parameters to perform the component selection of a generic Multi-objective Evolutionary Algorithm (MOEA). Here, I-Race is used to select the population size, population type, offspring size, external archive type and size, selection scheme, removal set-partitioning quality indicator and diversity operators. Moreover, in Radulescu et al. (2013) authors use I-Race to tune the bi-objective problem of finding anytime algorithms. Anytime algorithms are those able to find quality solutions at any time of their execution. Hypervolume indicator is used to define a single-objective tuning problem that searches for an approximation of the Pareto front of quality of solutions versus time. Moreover, they study the anytime behavior of multi-objective evolutionary algorithms where the quality of solutions obtained is also evaluated using the hypervolume indicator.

Nowadays the main focus of tuning methods is related to the optimization of several criteria, also known as multi-objective problems (Augusto et al., 2006). The bi-objective problem that considers the performance over time of metaheuristics is typically considered. Several multi-criterion approaches proposed in literature are based on single-criteria tuning methods as SPRINT-Race (Zhang et al., 2015) that takes inspiration from the *-Race approaches, MO-ParamILS (Blot et al., 2016) that is based on ParamILS. On the other hand, some of these new proposals are inspired in well known MOEAs as M-FETA (Smit et al., 2010) and EMOPaT (Ugolotti and Cagnoni, 2014). Furthermore, the tuning of multi-criterion algorithms optimizing a set of typical multi-objective indicators has attracted the attention of researchers in the area lately (Blot et al., 2017).

3. Tuning methods

This section introduces the three tuning methods: ParamILS, I-Race and Evoca. Here, we explain the main components of tuning processes performed by each tuning method studied in this work. We have selected these three methods because they have been extensively used in the tuning research area and they have implementations available online.

3.1. Parameter iterated local search

The Parameter Iterated Local Search (ParamILS) method (Hutter et al., 2009) works as an iterated local search algorithm. Algorithm 1 shows the structure of ParamILS method. Starting with a user defined parameter calibration, R tries searching randomly for a better calibration are performed. The resulting calibration undergoes local search through the *IterativeFirstImprovement*(c) function in line 8.

At each iteration, lines 9 to 21, ParamILS performs s random perturbations and the best calibration is then improved by the local search process. Its performance is then compared to the best parameter calibration found so far. There is also, a restart probability $p_{restart}$, which allows the method to escape from local optimum.