Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-3 (2016) 347–352

Investigation of hidden markov model for the tuning of metaheuristics in airline scheduling problems

Oussama Aoun* Malek Sarhani**
Abdellatif El Afia***

**ENSIAS - Mohammed V University, Rabat, Morocco*
oussama.aoun@um5s.net.ma
***ENSIAS - Mohammed V University, Rabat, Morocco*
malek.sarhani@um5s.net.ma
****ENSIAS - Mohammed V University, Rabat, Morocco*
a.elafia@um5s.net.ma

Abstract: The tuning approach consists in finding the most suitable configuration of an algorithm for solving a given problem. Machine learning methods are usually used to automate this process. They may enable to construct robust autonomous artifacts whose behavior becomes increasingly expert. This paper focuses on the restriction of this general problem to the field of air planning and more specifically the crew scheduling problem. Metaheuristics are widely used to solve this problem. Our approach consists of using hidden markov model to find the best configuration of the algorithm based on the estimation of the most likely state. The experiment consists of finding the best parameter values of the particle swarm optimization algorithm for the crew scheduling problem. Our approach has shown that it can be a promising solution for automatic optimization of airline scheduling problems.

Keywords: Tuning metaheuristics, hidden markov model, airline scheduling, particle swarm optimization, machine learning.

1. INTRODUCTION

Nowadays, air transport has experienced increased competition. Therefore, the companies have given much attention to the optimization of airline scheduling problems. In many cases, this problem is NP-hard and considered computationally intractable and then the importance of using metaheuristics for solving it.

The configuration of algorithms is one of the main challenges within metaheuristics; the reason lies in the fact that the algorithm performance depends heavily on the chosen parameter values. Concerning metaheuristics, their configuration can be done in two ways Hamadi (2013). The off-line configuration involves the adjustment of parameters before running the algorithm while the online configuration consists of adjusting the algorithm parameters while solving the problem. The first problem is known as the ‘parameter tuning problem’ and the second one as the ‘parameter control problem’.

The problem of parameter tuning in metaheuristics is as old as metaheuristics themselves. However, it has been done in most cases “manually” following the conventional propositions proposed in the literature. For instance, concerning particle swarm optimization (PSO) Kennedy & Eberhart (1997), most papers follow conventions to define the inertia weight and the acceleration factors such as Shi & Eberhart (1997) and Clerc (2011). However, these parameters depend on the problem type (unimodal, multi-modal...). That is, each problem has its own specificity and then the necessity of automating this pre-processing step. Even if it is known that the tuning is a necessary step in all metaheuristics, little effort is spent on the automation of this step.

The aim of the tuning approaches is to search for a suitable algorithm in the space of available configurations. More specifically, its aim is to choose the configuration which can lead to the best performance of the algorithm on an instance corresponding to the measure. This problem could be viewed as a general design whose components are selected according to the problem to be solved.

The tuning allows automatic configuration adjustment; this can be viewed as an expert system which can learn from its own experience based on previous results in order to improve its performance. Machine learning algorithms are frequently used for this purpose. They work at a high level in order to automate the metaheuristic applicability to each problem. This automation is very useful to build optimization software with enhanced performance (see for instance Bonesa software Smit & Eiben (2011)).

In this paper, the tuning problem to be considered is an airline planning problem (crew scheduling Barnhart et al. (1997) and the metaheuristic to be tuned is particle swarm optimization (PSO). The PSO parameters are the inertia weight (w), the acceleration factors (cognitive attraction $c_1$and social attraction c_2) and the population size. Also, to be able to use PSO for these binary problems. The canonical binary version of PSO proposed in Kennedy & Eberhart (1997) has been adopted.

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.058

The remainder of the paper is organized as follows. Section 2 provides a literature review. Section 3 presents the formulation of the tuning problem based on hidden markov model (HMM). Section 4 describes the experimental results and Section 5 is dedicated to conclusion.

2. LITERATURE REVIEW

In this section, we start by mentioning that the definition of the algorithm parameters can be done with three main approaches which are deterministic, adaptive and self-adaptive (see Eiben et al. (2007) for more details). By analyzing the literature on algorithm configurations, we can affirm that self-adaptive approaches are based on the on-line control of algorithms and the adaptive one are based on the off-line tuning, even if some papers have confused adaptive and self-adaptive approaches.

Thereby, we give in this section a brief review on deterministic and self-adaptive approaches used especially in particle swarm optimization (PSO), and then we focus on the off-line methods in general.

Firstly, a number of paper interested in defining deterministic configurations of PSO. One the one hand, as an example of the population size, Clerc (2011) proposed the following formula to define the swarm size.

One the other hand, concerning the values that have to be affected to the inertia weight and acceleration parameters. The classical way to define the inertia weight was proposed by Shi & Eberhart (1997), it consists of linearly decreasing w with the iterative generations. The definition of acceleration parameters has been done in many ways. For instance, Bratton & Kennedy (2007) affirmed that the convergence would be quick and guaranteed for $c_1+c_2> 4$. However, according to Perez & Behdinan (2007), $c_1+c_2 < 4$ is a condition that may enable the algorithm to be stable. Secondly, in Aoun et al. (2015), we have presented a brief review on the online configuration of PSO parameters.

Thirdly, the autonomous search book edited by Hamadi (2013) has surveyed the main approaches which can be used for parameters tuning of optimization algorithms. In Eiben et al. (2007), the tuning approaches have been classified into four categories depending the number of parameters and the number of functions. Hoos (2012) has presented the three most used procedures for tuning algorithms which are: Racing Procedures, ParamILS and Sequential Model-Based Optimization. In Epstein & Petrovic (2012), the authors showed how the training sample can be classified into positive and negative examples. This classification may enable us to use a supervised machine learning method.

Moreover, metaheuristics have been used for tuning metaheuristics. Indeed, the off-line configuration of the algorithm can be formulated as an optimization which aims to minimize the objective function. This idea has been introduced for evolutionary algorithms as meta-evolutionary algorithms. In other terms, an evolutionary algorithm is used to configure another one. A similar idea of the meta-evolutionary has been proposed in Yang et al. (2013). That is, the algorithm that has to be tuned can be used to tune the algorithm itself. The specificity of the paper is that the authors proposed to use a multi-objective approach. The firefly algorithm has been used to examine the proposed framework.

The tuning of metaheuristics is related to the generic notion of hyper-heuristics which consists of finding the most suitable configuration of heuristic algorithms such as local searches (simulated annealing, tabou search...). Machine learning has been used also to deal with hyper-heuristics Swan et al. (2014). Furthermore, Hoos (2012) proposed a hyper-heuristic solver based on a choice function which combines various numbers of strategies to learn the weighted mixture of heuristics for a given problem class. Also, Crawford et al. (2013) proposed another choice function which tends to rank the heuristics according to their ability to properly solve an instance of the problem and PSO has been used then for the tuning of the choice function parameters.

The use of machine learning for the tuning problem has been popularized by Birattari (2006). It consists of learning from problem instances. In particular, HMM has been successfully applied in a number of problems which have similarity with the tuning problem. For example, in character recognition, HMM can identify characters from a stream of observation sequences. In the same manner, at the evaluation phase of tuning, HMM will be used to identify the best configuration from the stream of execution data. Furthermore, HMM has been used as a tool to define autonomous systems as presented in a number of volume of the intelligent autonomous systems (IAS) book (see for instance Kojo et al. (2006) and therefore, it may be useful for autonomous search as presented in Hamadi (2013), in both offline and online configurations.

This feature of HMM has just been investigated for the online configuration (control problem) in Aoun et al. (2015). The proposed approach has given better results than the majority of the state of the art of PSO improvement in terms of both solution accuracy and convergence speed. In this paper, we examine it for the tuning problem. On the other hand, even if Birattari (2006) has affirmed that HMM is an adequate for the generation of instances in the tuning problem, it is difficult to estimate state-transition probabilities (by Baum-Welch or expectation maximization algorithms) and to system state given the estimated state-transition probabilities (by the Viterbi algorithm for instance). This is why he defined the racing approaches. We can see from the literature that the racing approaches have been well investigated, and a number of variants have been proposed (sampling F-Race, iterative F-Race, tNO-Race...). However, HMM has not been yet examined for this purpose.

Concerning air planning problems, the binary PSO has been successfully used to solve the crew problem. That is, it has given competitive results to the genetic algorithm (GA) in the crew pairing problem Ezzinbi et al. (2014).

At the end of this section, we can conclude on the one hand that the automation of the tuning of is not yet well developed especially in the practical aspect even if it importance is affirmed in many papers. One the other hand, hidden markov model is an adequate method that has to be investigated for tuning metaheuristics in order to build an autonomous optimization system.