Ann Math Artif Intell (2011) 61:87–104
DOI 10.1007/s10472-011-9230-5

Discovering the suitability of optimisation algorithms by learning from evolved instances

Kate Smith-Miles · Jano van Hemert

Published online: 19 April 2011

Abstract The suitability of an optimisation algorithm selected from within an algorithm portfolio depends upon the features of the particular instance to be solved. Understanding the relative strengths and weaknesses of different algorithms in the portfolio is crucial for effective performance prediction, automated algorithm selection, and to generate knowledge about the ideal conditions for each algorithm to influence better algorithm design. Relying on well-studied benchmark instances, or randomly generated instances, limits our ability to truly challenge each of the algorithms in a portfolio and determine these ideal conditions. Instead we use an evolutionary algorithm to evolve instances that are uniquely easy or hard for each algorithm, thus providing a more direct method for studying the relative strengths and weaknesses of each algorithm. The proposed methodology ensures that the meta-data is sufficient to be able to learn the features of the instances that uniquely characterise the ideal conditions for each algorithm. A case study is presented based on a comprehensive study of the performance of two heuristics on the Travelling Salesman Problem. The results show that prediction of search effort as well as the best performing algorithm for a given instance can be achieved with high accuracy.

Keywords Algorithm selection · Combinatorial optimization · Travelling salesman problem · Hardness prediction · Phase transition · Instance difficulty

Mathematics Subject Classifications (2010) 49-04 · 68Q25 · 68Q87 · 68T05 · 68T20 · 90B99

K. Smith-Miles (✉)
School of Mathematical Sciences, Monash University, Victoria 3800, Australia
e-mail: kate.smith-miles@sci.monash.edu.au

J. van Hemert
School of Informatics, University of Edinburgh, EH8 9AB Edinburgh, UK
e-mail: j.vanhemert@ed.ac.uk

1 Introduction

While many optimisation problems are classified as NP-hard based on their theoretical worst-case analysis, there are many instances of such problems that are quite easy for any algorithm to solve. Consider the simple example of the Travelling Salesman Problem with all cities arranged in a circle. Clearly, few algorithms would be challenged by this instance of the problem and the search effort to find the optimal solution would be minimal. If the nature of the instance changes however, and the cities are arranged in tight clusters of variable sizes with outlier cities that do not belong to any clusters, some algorithms would require more search effort than others to arrive at an optimal solution. A generic answer to the question of what makes an optimisation problem hard has been elusive, since it is clear that problem specific characteristics determine the intrinsic difficulty of a particular instance of an optimisation problem for a particular algorithm [23]. Consequently, there has been increasing interest in measuring the characteristics of instances for a given optimisation problem to gain insights into which features of an instance make a particular algorithm perform well or poorly. Examples of such studies include Nudelman et al. [24] and Xu et al. [40] for SAT problems, Cho et al. [6] and Hall and Posner [11] for knapsack problems, Smith-Miles [31] for the QAP, Leyton-Brown et al. [19, 20] for combinatorial auctions, and Smith-Miles et al. for job shop scheduling [33]. The methods used to predict and gain insights come from the statistical and machine learning communities, and the task is to learn to predict algorithm performance based only on the features of an instance. Performance prediction is a key component in the broader goals of automated algorithm selection and generating knowledge about the ideal conditions for each algorithm which, in turn, is knowledge that can be exploited to influence better algorithm design.

The algorithm selection framework of Rice [27] provides a convenient representation of this task, as discussed in a recent survey paper [31], and enables us to generalize the important components of the methodology. The algorithm selection problem requires a complete set of meta-data from which to learn, comprising: a large set of diverse instances, suitable features with which to characterise the instances, a diverse portfolio of algorithms, and a performance metric for the algorithms. Once we have sufficient meta-data, we can apply machine learning methods to learn the relationships within the meta-data.

One of the key challenges to successful learning from the meta-data though, is to ensure sufficient diversity of the instances in both feature space and algorithm performance space. There is nothing meaningful that will be learned from the meta-data if all instances map to the same region in feature space (in which case the features are not discriminating enough to model sub-classes of instances), or if all algorithms perform similarly on all instances (in which case the instances are not discriminating enough to test algorithm performance). The instance generation method, and the choice of features are clearly critical to the success of any subsequent analysis of the meta-data.

The most common approach to generating problem instances for meta-data is to randomly generate instances based on statistically varying features known to relate to instance difficulty. The performance of algorithms can then be measured on these instances, and we hope that the instance generation process has created sufficient diversity in the feature space that some meaningful relationships can be inferred. The