Information Sciences 299 (2015) 412–432

Contents lists available at ScienceDirect

Information Sciences

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

A tensor-based selection hyper-heuristic for cross-domain heuristic search

Shahriar Asta *, Ender Özcan

ASAP Research Group, School of Computer Science, University of Nottingham, NG8 1BB Nottingham, UK

ARTICLE INFO

Article history:
Received 29 May 2014
Received in revised form 1 November 2014
Accepted 8 December 2014
Available online 17 December 2014

Keywords:
Hyper-heuristic
Data science
Machine learning
Move acceptance
Tensor analysis
Algorithm selection

ABSTRACT

Hyper-heuristics have emerged as automated high level search methodologies that manage a set of low level heuristics for solving computationally hard problems. A generic selection hyper-heuristic combines heuristic selection and move acceptance methods under an iterative single point-based search framework. At each step, the solution in hand is modified after applying a selected heuristic and a decision is made whether the new solution is accepted or not. In this study, we represent the trail of a hyper-heuristic as a third order tensor. Factorization of such a tensor reveals the latent relationships between the low level heuristics and the hyper-heuristic itself. The proposed learning approach partitions the set of low level heuristics into two subsets where heuristics in each subset are associated with a separate move acceptance method. Then a multi-stage hyper-heuristic is formed and while solving a given problem instance, heuristics are allowed to operate only in conjunction with the associated acceptance method at each stage. To the best of our knowledge, this is the first time tensor analysis of the space of heuristics is used as a data science approach to improve the performance of a hyper-heuristic in the prescribed manner. The empirical results across six different problem domains from a benchmark indeed indicate the success of the proposed approach.

1. Introduction

Hyper-heuristics have emerged as effective and efficient methodologies for solving hard computational problems. They perform search over the space formed by a set of low level heuristics, rather than solutions directly [49]. Burke et al. [13] defined a hyper-heuristic as *a search method or learning mechanism for selecting or generating heuristics to solve computational search problems*. Hyper-heuristics are not allowed to access problem domain specific information. It is assumed that there is a conceptual *barrier* between the hyper-heuristic level and problem domain where the low level heuristics, solution representation, etc. reside. This specific feature gives hyper-heuristics an advantage of being more general than the existing search methods, since the same hyper-heuristic methodology can be reused for solving problem instances even from different domains. More on hyper-heuristics can be found in [12,49,16]. The focus of this paper is on selection hyper-heuristics which, often operate under a single point based search framework by improving an initially created solution iteratively, exploiting the strengths of multiple low level heuristics. At each step, a complete solution is updated forming a new solution using a selected heuristic and this new solution is considered for use in the next step via a move acceptance method.

* Corresponding author.
E-mail addresses: sba@cs.nott.ac.uk (S. Asta), Ender.Ozcan@nottingham.ac.uk (E. Özcan).

http://dx.doi.org/10.1016/j.ins.2014.12.020
0020-0255/

There are some recent studies indicating the effectiveness and potential of mixing multiple move acceptance methods under a selection hyper-heuristic framework. Kheiri and Özcan [28] described a bi-stage hyper-heuristic which allows improving and equal moves only in the first stage while a naive move acceptance method allows worsening moves in the following stage. Özcan et al. [47] recently combined different move acceptance methods and tested different group decision making strategies as a part of selection hyper-heuristics. One of the rare theoretical studies reveal that mixing move acceptance within a selection hyper-heuristic framework could yield a better running time on some benchmark functions [36]. Machine learning techniques, such as reinforcement learning and learning classifier systems have been used as a component of selection hyper-heuristics since the early ideas have emerged [23]. In this study, we propose a multi-stage selection hyper-heuristic, hybridizing two simple move acceptance methods, which is significantly improved by the use of a machine learning technique, namely tensor analysis [40].

In the proposed approach, we represent the trail of a selection hyper-heuristic as a 3rd order tensor. Tensor analysis is performed during the search process to detect the latent relationships between the low level heuristics and the hyper-heuristic itself. The feedback is used to partition the set of low level heuristics into two subsets where heuristics in each subset are associated with a separate move acceptance method. Then a multi-stage hyper-heuristic combining a random heuristic selection with two simple move acceptance methods is formed. While solving a given problem instance, heuristics are allowed to operate only in conjunction with the corresponding move acceptance method at each alternating stage. This overall search process can be considered as a generalized and a non-standard version of the iterated local search [38] approach in which the search process switches back and forth between diversification and intensification stages. More importantly, the heuristics (operators) used at each stage are fixed before each run on a given problem instance via the use of tensors. To the best of our knowledge, this is the first time tensor analysis of the space of heuristics is used as a data science approach to improve the performance of a selection hyper-heuristic in the prescribed manner. The empirical results across six different problem domains from a benchmark indicate the success of the proposed hyper-heuristic mixing different acceptance methods.

This paper is organized as follows. An overview of hyper-heuristics, together with the description of the benchmark framework used in this paper is given in Section 2. Section 3 includes a description of the data analysis method we have used in our study. Section 4 discusses a detailed account of our framework while experimental design issues as well as the results are discussed in Section 5. Finally, conclusion is provided in Section 6.

2. Selection hyper-heuristics

A hyper-heuristic either selects from a set of available low level heuristics or generates new heuristics from components of existing low level heuristics to solve a problem, leading to a distinction between *selection* and *generation* hyper-heuristic, respectively [13]. Also, depending on the availability of feedback from the search process, hyper-heuristics can be categorized as *learning* and *no-learning*. Learning hyper-heuristics can be further categorized into online and offline methodologies depending on the nature of the feedback. Online hyper-heuristics learn *while* solving a problem whereas offline hyper-heuristics process collected data gathered from training instances prior to solving the problem. The framework proposed in this paper is a single point based search algorithm which fits best in the online learning selection hyper-heuristic category.

A selection hyper-heuristic has two main components: *heuristic selection* and *move acceptance* methods. While the task of the heuristic selection is to choose a low level heuristic at each decision point, the move acceptance method accepts or rejects the resultant solution produced after the application of the chosen heuristic to the solution in hand. This decision requires measurement of the quality of a given solution using an *objective* (evaluation, fitness, cost, or penalty) function. Over the years, many heuristic selection and move acceptance methods have been proposed. A survey on hyper-heuristics including their components can be found in [12,49].

2.1. Heuristic selection and move acceptance methodologies

In this section, we describe some of the basic and well known heuristic selection approaches. [20,21] are the earliest studies testing simple heuristic selection methods as a selection hyper-heuristic component. One of the most basic and preliminary approaches to select low level heuristics is the Simple Random (SR) approach requiring no learning at all. In SR, heuristics are chosen and applied (once) at random. Alternatively, when the randomly selected heuristic is applied repeatedly until the point in which no improvement is achieved, the heuristic selection mechanism is Random Gradient. Also, when all low level heuristics are applied and the one producing the best result is chosen at each iteration, the selection mechanism is said to be greedy. The heuristic selection mechanisms discussed so far do not employ learning. There are also many selection mechanisms which incorporate learning mechanisms. Choice Function (CF) [20,22,51] is one of the learning heuristic selection mechanisms which has been shown to perform well. This method is a score based approach in which heuristics are adaptively ranked based on a composite score. The composite score itself is based on few criteria such as: the individual performance profile of the heuristic, the performance profile of the heuristic combined with other heuristics and the time elapsed since the last call to the heuristic. The first two components of the scoring system emphasize on the recent performance while the last component has a diversifying effect on the search.