Expert Systems With Applications 63 (2016) 208–221

Contents lists available at ScienceDirect

Expert Systems With Applications

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

CHAMP: Creating heuristics via many parameters for online bin packing

Shahriar Asta, Ender Özcan*, Andrew J. Parkes

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

ARTICLE INFO

Article history:
Received 12 November 2015
Revised 12 June 2016
Accepted 2 July 2016
Available online 4 July 2016

Keywords:
Genetic algorithms
Heuristics
Packing
Decision support systems
Learning systems
Noisy optimization

ABSTRACT

The online bin packing problem is a well-known bin packing variant and which requires immediate decisions to be made for the placement of a lengthy sequence of arriving items of various sizes one at a time into fixed capacity bins without any overflow. The overall goal is maximising the average bin fullness. We investigate a 'policy matrix' representation, which assigns a score for each decision option independently and the option with the highest value is chosen, for one-dimensional online bin packing. A policy matrix might also be considered as a heuristic with many parameters, where each parameter value is a score. We hence effectively investigate a framework which can be used for creating heuristics via many parameters. The proposed framework combines a Genetic Algorithm optimiser, which searches the space of heuristics in policy matrix form, and an online bin packing simulator, which acts as the evaluation function. The empirical results indicate the success of the proposed approach, providing the best solutions for almost all item sequence generators used during the experiments. We also present a novel fitness landscape analysis on the search space of policies. This study hence gives evidence of the potential for automated discovery by intelligent systems of powerful heuristics for online problems; reducing the need for expensive use of human expertise.

1. Introduction

There are many problems in which decisions have to be made without fully knowing their future implications, hence, the use of a heuristic 'dispatch policy' is common in such situations. Heuristics are usually designed by an expert through a trial and error process particularly for solving the problems from a specific domain. Given the time-consuming nature of the overall process, automated generation of heuristics has been of interest to both academics and practitioners (Burke et al., 2009; Chakhlevitch & Cowling, 2008; Ross, 2005).

The previous work of Özcan and Parkes (2011) provided an approach for the automatic production of heuristics. The authors formulated the whole process as a special type of parameter tuning (Smit & Eiben, 2009; Yarimcam, Asta, Özcan, & Parkes, 2014) in which the number of parameters is much larger than usually considered. This growth in the number of parameters is due to the use of a matrix encoding as a heuristic which represents various potential decisions. In a way, a 'policy matrix' defines an 'index policy' (e.g., Gittins (1979)) covering the possible observed states. A potential decision is given a score independently of other decisions and the decision with the largest score based on a given state is selected.

The online bin packing problem (Coffman, Garey, & Johnson, 1997; Csirik & Woeginger, 1998) requires immediate decisions to be made for the packing of a lengthy sequence of arriving items of various sizes one at a time into fixed capacity bins without any overflow. In this work, we particularly study the well-known online bin-packing problem, creating a policy that is based on using a (large) matrix[1] of 'heuristic scores'. An observed state for online bin packing considers the item to be packed and each open bin in which that item can fit; it then gets a score from the policy matrix based on its remaining space and the highest value option (open bin) is chosen for packing. The policy matrix can be viewed as a heuristic with many parameters requiring optimization yielding scores which can achieve the highest bin fullness in the long run. We describe a framework which allows the use of an optimiser for 'Creating Heuristics via Many Parameters' (CHAMP) and online bin packing simulator that can be used as an evaluation function

* Corresponding author.
E-mail addresses: shahriarasta@gmail.com (S. Asta), ender.ozcan@nottingham.ac.uk (E. Özcan), andrew.parkes@nottingham.ac.uk (A.J. Parkes).

[1] Here, the term 'matrix' is used as a convenience for a 2-d array; there is no implication of it being used for matrix/linear algebra.

http://dx.doi.org/10.1016/j.eswa.2016.07.005

for a given policy on a given problem instance. Packing problem instances are specified in terms of a specified bin capacity and a stochastically generated sequence of item sizes taken from a specified range. For specific instance generators, good policies are found using a Genetic Algorithm (GA) as the optimiser under the CHAMP framework to search the space of matrices, with the matrix-based policies being evaluated directly by packing a (large) number of items. The proposed approach can also be considered as an intelligent expert system which captures the decision-making ability of a human expert as a policy matrix and uses a GA to automatically create packing heuristics for online bin packing with the goal of producing better performing packing heuristics than the human designed ones.

The primary result (reported in Özcan and Parkes (2011) and Asta, Özcan, and Parkes (2013a)) is that the GA finds matrices for the specific packing problems that perform significantly better than the standard general-purpose heuristics such as first and best fit (Coffman, Galambos, Martello, & Vigo, 1999; Rhee & Talagrand, 1993). Yarimcam et al. (2014) showed that applying a parameter tuning approach does not match the performance of GA in the overall. Asta, Özcan, Parkes, and Etaner-Uyar (2013b) showed that k-means clustering can be used to generalise the behaviour of GA for solving a given online bin packing problem, however, GA still performed the best. Asta and Özcan (2015) embedded a tensor-based learning approach into GA to adapt the mutation rate for each locus. In this paper, we firstly clarify some potentially confusing and counter-intuitive issues in our search methodology and so our experimental design which is different than the previous studies. We then investigate various parameter settings for the proposed policy matrices, including integer and binary settings, as well as two different initialisation schemes.

We also conducted, and present here, a *fitness landscape analysis* (Tavares, Pereira, & Costa, 2008; Wright, 1932) to better understand the performance of the proposed algorithm in relation with the policy matrix representation. To the best of our knowledge, this is the first study on fitness landscape analysis on the search space of online bin-packing policies. A fitness landscape is obtained by associating a fitness value indicating the quality of a solution with each point in the search space, on which a neighbourhood is defined. In a standard landscape analysis it is the direct solution space that is considered; the novelty of the study here is that instead it is in the space of the heuristics that are used to construct the solutions. It is generally not practical to cover all points in the search space, since the number of points often grows exponentially with respect to the size of a problem. Hence, except for the instance generator with 'small' capacity, sampling is used. Naturally, as standard for fitness landscape analysis, the results are based on a particular neighbourhood (which depends mainly on the solution representation), as well as the move operators employed.

We emphasise that we are aiming to optimise the performance of the heuristic when averaged over many sequences; this is quite different from heuristics (for example, 'Harmonic heuristics', Richey (1991)) designed to be approximation algorithms and so optimise the worst performance achieved over all possible sequences. Also, standard methods for policy creation in stochastic processes (Markov chains, reinforcement learning, etc.) are likely to be able to generate good policies in some simpler cases (e.g., Gittins (1979)). However, our driving motivation is to form the basis for evolutionary and other relevant search methods to aid in the generation of heuristics and heuristic policies for complex situations (and out of the reach of analytical methods). For example, situations when sequences are finite (though long) rather than the infinite limit case usually considered in stochastic processes theory (Gittins, Glazebrook, & Weber, 2011), or complex non-Markovian time-varying distributions, etc. Such complex situations might include practical combinatorial optimisation problems using constructive heuristics, or queuing networks.

It is important to note we will be creating heuristics for specific (stochastic) generators of item sequences. That is, each usage of CHAMP-GA will be with some specific method of generating sequences of items, the fitness of a heuristic will be with respect to that generation method, and so the resulting automatically generated heuristic (policy matrix) will be targeting such a generator — though of course the generator is stochastic and so each time the generator is used it will give a different sequence of item sizes, and the policy matrix should work well on all the different sequences.

Our aim is precisely to develop methods that can adapt to specific properties of the sets of sequences of items and exploit their properties. Emphasising again, that policies are evolved to learn from and hence work for (sequences produced by) a generator, and *not* for only one specific sequence. Heuristics tailored to specific generators will generally be expected to perform better than ones aiming to work on all generators. We believe that such creation of tailored or specialised heuristics is practically useful because users are likely to have input problems with particular characteristics and so would like to be able to develop heuristics that are best able to exploit those characteristics, but would also like to be able to do so automatically, without the need to hire an expert in heuristics. This is particularly relevant if the characteristics of their problems should change over time; in which case, a system should be able to cheaply re-tune the heuristics without having to go back to a human expert. That is, a goal is to develop systems for the creation of heuristics that can reliably, and automatically, adapt to many different situations, without the need for external interventions by experts.

The structure of the paper is as follows: Section 2 gives a brief review and pointers into the literature of existing work on firstly bin-packing and also computer-based methods to help design heuristics. Section 3.1 gives basic definitions of the bin-packing problem, the instance generators that we use, and the existing standard heuristics. Section 3 specifies the policy matrix-based CHAMP framework we use in order to define the packing heuristics. Section 3.4 describes the GA search method we use to find good policies. The experimental methodology is explained in Section 4. Section 5 gives the main results on examples that are large enough to allow policy improvements, but still small enough that the structure of the resulting policies can be (partially) understood. Moreover, a fitness landscape analysis on the search space of policies for online bin packing is provided. Section 6 summarises our results and their implications, and then discusses future plans.

2. Related work

We now briefly cover the bin packing problem and relevant issues, followed by existing methods for automated discovery of heuristics from the scientific literature.

2.1. Bin packing

One dimensional (offline) bin packing is a well known NP-hard (Garey & Johnson, 1990) combinatorial optimisation problem. Solving this problem requires packing of a number of pieces with given sizes into a minimal number of fixed capacity bins. This process can also be considered as partitioning (grouping) a set of integer values into subsets (bins) in such a way that the sum of the integers within a subset does not exceed the fixed capacity. The complete information, including the number of pieces and size of each piece, is known to the solution approaches for the generic one dimensional offline bin packing.

A range of solution approaches from approximation algorithms to metaheuristic optimisation methods have been proposed for