Expert Systems with Applications 41 (2014) 6876–6889

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A unified hyper-heuristic framework for solving bin packing problems

Eunice López-Camacho[a], Hugo Terashima-Marin[a,*], Peter Ross[b], Gabriela Ochoa[c]

[a] Tecnológico de Monterrey, Av. E. Garza Sada 2501, Monterrey, NL 64849, Mexico
[b] School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, UK
[c] Computing Science and Mathematics, University of Stirling, Scotland, UK

ARTICLE INFO

Article history:
Available online 10 May 2014

Keywords:
Bin packing problems
Evolutionary computation
Hyper-heuristics
Heuristics
Optimization

ABSTRACT

One- and two-dimensional packing and cutting problems occur in many commercial contexts, and it is often important to be able to get good-quality solutions quickly. Fairly simple deterministic heuristics are often used for this purpose, but such heuristics typically find excellent solutions for some problems and only mediocre ones for others. Trying several different heuristics on a problem adds to the cost. This paper describes a hyper-heuristic methodology that can generate a fast, deterministic algorithm capable of producing results comparable to that of using the best problem-specific heuristic, and sometimes even better, but without the cost of trying all the heuristics. The generated algorithm handles both one- and two-dimensional problems, including two-dimensional problems that involve irregular concave polygons. The approach is validated using a large set of 1417 such problems, including a new benchmark set of 480 problems that include concave polygons.

1. Introduction

Finding an arrangement of pieces to cut or pack inside larger objects is known as the cutting and packing problem. Besides the academic interest in this NP-hard problem, there are numerous industrial applications of its many variants. The one-dimensional (1D) and two-dimensional (2D) bin packing problems (BPPs) are particular cases of the cutting and packing problem. The 1D BPP can be applied, for example, to the assignment of commercial breaks on television and for copying a collection of files to disks (Bhatia, Hazra, & Basu, 2009). For the 2D BPP, the case of rectangular pieces is the most widely studied. However, the irregular case is seen in a number of industries where parts with irregular shapes are cut from rectangular materials. For instance, in the shipbuilding industry, plate parts with free-form shapes for use in the inner frameworks of ships are cut from rectangular steel plates, and in the garment industry, parts of clothes and shoes are cut from fabric or leather (Okano, 2002). Other applications include the optimization of layouts within the wood, sheet metal, plastics, and glass industries (Burke, Hellier, Kendall, & Whitwell, 2006). In these industries, improvements of the arrangement can result in a large saving of material (Hu-yao & Yuan-jun, 2006).

Hyper-heuristics aim at automating the design of heuristic methods to solve difficult search problems and providing a more general procedure for optimization (Burke et al., 2003; Pillay, 2012; Burke, Gendreau, Hyde, Kendall, & Ochoa, 2013). Hyper-heuristics differ from the widely-used term meta-heuristic: instead of searching within the space of solutions, they explore the space of heuristics (Vázquez-Rodríguez, Petrovic, & Salhi, 2007; Pappa et al., 2013). The idea is to use a variety of methods to discover algorithms, based on single heuristics, that have good worst-case performance across a range of problems and are fast in execution (Ross, 2014). There are two main types of hyper-heuristic: selection hyper-heuristics, which are methods for choosing or selecting existing heuristics, and generation hyper-heuristics which focus on generating new heuristics from components of existing heuristics (Burke et al., 2013; Burke et al., 2010a). The approach presented in this paper is of the first type.

Over the last few years, one trend in combinatorial optimization has been to find more general solvers capable of extending to other types of problems within a domain and even crossing domain boundaries. For example, Burke et al. (2010b) conducted an empirical study that ran the same hyper-heuristic strategy in three different domains: 1D bin packing, permutation flow shop and personnel scheduling. Burke, Hyde, Kendall, and Woodward (2012) presented a genetic programming system to automatically generate a good quality heuristic for each instance of the one-, two-, and three-dimensional knapsack and bin packing problems with rectilinear pieces; however, because the generated heuristics

* Corresponding author. Tel.: +52 8181582045.
E-mail addresses: eunice.lopez@itesm.mx (E. López-Camacho), terashima@itesm.mx (H. Terashima-Marin), P.Ross@napier.ac.uk (P. Ross), gabriela.ochoa@cs.stir.ac.uk (G. Ochoa).

http://dx.doi.org/10.1016/j.eswa.2014.04.043
0957-4174/

are instance-specific, the computational costs involved are non-trivial. Ochoa et al. (2012) proposed a software framework named HyFlex (Hyper-heuristic Flexible framework) for developing cross-domain search methodologies along six different optimization problems.

In this paper, we introduce an evolutionary hyper-heuristic framework for solving 1D and 2D BPPs (rectangular, convex and concave shapes) that automatically chooses which heuristic to apply at each step in building a good solution. The approach described in this paper is a development of earlier work on solving the 1D BPP (Ross, Marín-Blázquez, Schulenburg, & Hart, 2003), the 2D regular packing problem (Terashima-Marín, Farías-Zárate, Ross, & Valenzuela-Rendón, 2006) and the 2D irregular (convex only) packing problem (Terashima-Marín, Ross, Farías-Zárate, López-Camacho, & Valenzuela-Rendón, 2010). In that earlier work, the solution construction process used an ad-hoc simplification of the current problem state when deciding what to do next and, in the 2D cases, a large set of possible basic heuristics.

The main contributions of this paper are:

- A unified framework that handles 1D, 2D regular (rectangles), and 2D irregular (convex and non-convex polygons) packing problems, together with an empirical analysis of its performance on a large unseen set of such problems.
- An experiment-based methodology for deciding which heuristics should be included in the framework.
- A data-mining methodology for choosing the problem-state representation to be used.
- The creation of a new, large benchmark set of 2D problems that include some non-convex polygons.

2. Background and related work

Many heuristics have been developed for specific problems but none of them seems able to provide good-quality results for all instances. Certain problems may contain features that enable a particular heuristic to work well, but those features may not be present in other problems and so might lower that heuristic's performance. Research in hyper-heuristics has developed algorithms with some claims to more generality, but there is interest in seeing whether even more general architectures can be developed, that are capable of solving many different kinds of problem efficiently. Recent work by Ochoa et al. (2012), introduced a software framework called HyFlex for the development of cross-domain search methodologies. The framework provides a common interface for treating different combinatorial problems and provides the problem-specific algorithm components. Hyflex can be seen as a benchmark framework for developing, testing and comparing the generality of algorithms such as selection hyper-heuristics. HyFlex has served to test algorithms in different domains like maximum satisfiability, one dimensional bin packing, permutation flow shop, personnel scheduling, traveling salesman and vehicle routing. Other interesting investigations have been motivated by the HyFlex system, see for example the work by Burke et al. (2010b) where several hyper-heuristics combining two heuristic selection and three acceptance approaches were compared, and other extensions are given in Burke, Gendreau, Ochoa, and Walker (2011). In a related study, Burke et al. (2012) proposed a general packing methodology that includes 1D, 2D (orthogonal) and 3D (orthogonal) bin packing and knapsack packing. They presented a genetic programming system to automatically generate a good quality heuristic for each instance among the different problems considered although at a non-trivial cost per instance. HyFlex has also served as a framework for the CHeSC 2011 algorithm competition, won by Misir, Verbeeck, Causmaecker, and Berghe (2011) with an algorithm which provides an intelligent way of selecting heuristics, pairing heuristics and adapting the parameters of heuristics online. They later extended this (Misir, Verbeeck, Causmaecker, & Berghe, 2013) by focusing on the single heuristic sets involved and on the distinct experimental limits. Other recent research in selection hyper-heuristics was introduced by Kalender, Kheiri, Özcan, and Burke (2013) in which a simulated annealing-based move acceptance method is combined with a learning heuristic selection algorithm to manage the single heuristics.

HyFlex and related systems use a selection hyper-heuristic approach which operates on complete candidate solutions, perturbing them to try to improve their quality. As such, solving an instance typically involves some search, although usually limited. The work presented in this paper instead uses a selection hyper-heuristic approach that constructs a solution incrementally, each step of which could be expressed as a simple lookup of what to do next. The approach uses significant search effort to create such an incremental solution-builder, but the amortized cost of generating solutions to unseen problems is then much lower than for HyFlex-type methods. This framework has also been used for solving Constraint Satisfaction Problems (Terashima-Marín, Ortiz-Bayliss, Ross, & Valenzuela-Rendón, 2008).

One of the possible limitations of this approach, as stated by Sim and Hart (2013), is that if the nature of the unseen problems changes over time, the system may need periodic re-training.

Other heuristic-selection mechanisms have been used, for example Cowling, Kendall, and Soubeiga (2000) used a choice function based on the performance of single and pairs of heuristics. Burke, Petrovic, and Qu (2006) employed a case-based reasoning approach to tackle timetabling problems, while Bai, Blazewicz, Burke, Kendall, and McCollum (2012) proposed a learning approach by updating the heuristic selection weights depending on the heuristic performance after each learning period. Walker, Ochoa, Gendreau, and Burke (2012) used HyFlex to tackle a large set of instances within the domain of Vehicle Routing Problem by using two adaptive variants of a multiple neighborhood iterated local search algorithm.

3. The bin packing problem

The cutting and packing problem has been studied since 1939 (Kantorovich, 1960), even though a more intensive research started after the middle of the twentieth century. In 2007, Wäscher, Hausner, and Schumann (2007) suggested a complete problem typology which is an extension of Dychoff (1990). In that work, authors state that, in general terms, cutting and packing have a common identical structure given by a set of large objects that are to be filled and a set of items with which to do the filling, without overlapping other items or the edges of the objects.

In this paper, we consider the following problem types in Wäscher et al. typology: (a) the 1D single bin size bin packing problem, (b) the 2D regular single bin size bin packing problem as well as (c) the 2D irregular single bin size bin packing problem.

In the 1D BPP, there is an unlimited supply of bins, each with capacity $c > 0$, and a set of n items (each one of size $s_i < c$) is to be packed into the bins. The aim is to minimize the total number of bins used. In the 2D BPP, there is a set $L = (a_1, a_2, \ldots, a_n)$ of pieces to pack and an infinite set of identical rectangular *objects* into which the pieces are to be packed. The aim is to minimise the number of objects needed. A *problem instance* $I = (L, x_0, y_0)$ consists of a list of elements L and object dimensions x_0 and y_0. The term '2D regular BPP' is mainly used when all pieces are rectangular and the term '2D irregular BPP' refers to the case where pieces can be polygonal, not just rectangular. We deal only with the off-line BPP, in which the list of pieces to be packed is given in advance.