Expert Systems with Applications 42 (2015) 3297–3305

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Two-dimensional residual-space-maximized packing

Yanchao Wang, Lujie Chen *

Singapore University of Technology and Design, 20 Dover Drive, Singapore

ARTICLE INFO

Article history:
Available online 20 December 2014

Keywords:
Two-dimensional packing
Heuristics
Residual space
Residual-space-maximized packing

ABSTRACT

Many approaches exist for solving two dimensional rectangle-packing problems. Some rely on multiple heuristic policies to detect suitable packing positions. Others resort to searching for a sound packing sequence from a great number of variations. This paper describes a heuristic algorithm with only a single policy: maximize the residual space during packing, which ensures that rectangles to be packed will fit into the space with maximum likelihood. An efficient implementation is proposed to realize the policy. Experimental results based on openly available datasets demonstrate that the proposed algorithm is comparable to most state-of-the-art algorithms in space efficiency while is significantly faster in data processing. In the case of large-scale problems tested, it is the best by both evaluation metrics.

1. Introduction

Cutting and Packing (C&P) is a class of optimization problems with a wide range of applications in resource management (Dyckhoff, 1990). Over years of research, several subdivisions of C&P problems have been studied scholarly and their solutions have been implemented in the industry. Although C&P is known to be NP-hard (Non-deterministic Polynomial-time hard) in general (Garey & Johnson, 1979), existing methods can find near-optimal solutions to various problems; nevertheless, new approaches and algorithms are still emerging in the research field. Some offer even better solutions in broader scenarios, while others reduce the time and space overhead in implementation.

This paper pertains to two-dimensional (2D) rectangle packing, focusing on bin packing – single bin size bin packing problem (SBSBPP) and strip packing – open dimension problem (ODP) in the typology of C&P (Wäscher, Haußner, & Schumann, 2007). The objective of the former is to minimize the number of fixed-size bins and that of the latter is to minimize the overall height of a fixed-width, infinitely long bin. Specifically, the problems under investigation belong to the RF (Rotated, Free cutting) subtype as classified by Lodi, Martello, and Vigo (1999), where input rectangles can be rotated and packing is not constrained by guillotine cutting.

A 2D rectangle-packing algorithm is developed based on a single policy: the residual space (RS), i.e. the unused region, should be maximized during each step of packing. It will be shown that an efficient implementation exist for such a residual-space-maximized packing (RSMP) policy. RSMP can produce comparable results to existing algorithms but runs considerably faster. The related work to this study is reviewed in Section 2. The principle of RSMP is described in Section 3. Comparisons with existing algorithms based on benchmark datasets are presented in Section 4. Conclusion and future work of the proposed approach are presented in Section 5.

2. Related work

One of the early approaches to 2D packing is the so-called exact method based on integer programming formulation (Beasley, 1985; Martello & Vigo, 1998). It attempts to find the packing positions of rectangles by solving a set of linear equations, constructed to optimize certain criteria. The number of equations to solve is related to the number of rectangles to pack; thereby the computational cost is relatively high.

A different approach, known as the heuristic approach, produces solutions with relatively less computational overhead. Heuristic algorithms are motivated by intuitive experience, e.g. rectangles are piled to the bottom-left (BL) corner of a bin by the BL heuristic (Baker, Coffman, & Rivest, 1980). Performance of an algorithm depends on the efficacy of the heuristic. Many heuristic algorithms consist of a placement method, used to place a rectangle at a specific location in a bin, and a sequence-generation strategy, to create various sequences by changing the order of the input rectangles. Bottom-left (BL) (Baker et al., 1980), bottom-left-fill (BLF) (Chazelle, 1983), improved bottom-left (IBL) (Liu & Teng, 1999) and difference process DP (Lai & Chan, 1997) are widely used

* Corresponding author.
E-mail addresses: wangyanchao1988@gmail.com (Y. Wang), chenlujie@sutd.edu.sg (L. Chen).

http://dx.doi.org/10.1016/j.eswa.2014.12.021
0957-4174/

placement methods. Methods such as floor-ceiling (FC) and touching perimeter (TP) have also been reported (Lodi et al., 1999).

Sequence generation is motivated by the fact that a placement method would produce different results if the order of the input rectangles is varied. When different sequences are tested, the best result can be treated as the solution. To date, lots of strategies have been proposed, such as simulated annealing (SA) (Dowsland, 1993) and genetic algorithm (GA) (Jakobs, 1996; Liu & Teng, 1999), artificial neural network (Dagli & Poshyanonda, 1997), tabu search (TS) (Lodi et al., 1999), unified tabu search (UTS) (Bennell, Lee, & Potts, 2013; Lodi, Martello, & Vigo, 2004), greedy randomized adaptive search procedure (GRASP) (Alvarez-Valdés, Parreño, & Tamarit, 2005), sequential value correction (SVC) (Belov, Scheithauer, & Mukhacheva, 2008), iterative maximal area (IMA) (Hayek, Moukrim, & Negre, 2008), iterative doubling binary search (IDBS) (Wei, Oon, Zhu, & Lim, 2011), single- and multi-crossover genetic algorithm (SGA and MXGA) (Bennell et al., 2013). Some of these algorithms have achieved near-optimal solutions in certain scenarios; however, the drawback is the prolonged computation time due to many trials of different sequences.

There are alternatives to sequence generation relatively independent from a placement method. For example, Burke, Kendall, and Whitwell (2004) proposed a best-fit (BF) heuristic algorithm that could actively select a suitable sized rectangle to pack, which makes it a placement method that generates a sequence during packing. BF can also be combined with a sequence-generation strategy to improve the performance (Burke, Kendall, & Whitwell, 2006). Variations of BF have been proposed, such as the bidirectional best-fit (BBF) (Aşık & Özcan, 2009) and modified bidirectional best-fit (BBFM) (Özcan, Kai, & Drake, 2013) algorithms. Among these, BBFM is the best in terms of space-efficiency, owing to a large number of (6912) policy combinations incorporated. The supposedly most suitable policy is applied at each phase of packing but the computational cost is heavy due to multiple levels of nested comparison.

Some recent publications suggested that effective algorithms could be based on only a few policies. The fast heuristic (FH) algorithm (Leung & Zhang, 2011) applied a scoring policy and the binary search heuristic algorithm (BSHA) (Zhang, Wei, Leung, & Chen, 2013) incorporated a primary policy that ensures the smoothness of the unused region's envelop. Better results than most existing algorithms were achieved and the processing time was shorter in some scenarios than the algorithms of many policies (Aşık & Özcan, 2009; Özcan et al., 2013). However, the processing time was not proportional to the number of rectangles: some small-scale tasks might take longer to process than large-scale problems.

3. Principle

The heuristic policy of the proposed residual-space-maximized packing (RSMP) algorithm is motivated by an observation: the most problematic issue of packing is where to put big rectangles. If not properly handled, they may occupy a great number of bins (in bin packing) or reach considerable piled height (in strip packing) with many small spaces unused. Hence, it is reasonable to hypothesize that the best packing position of a rectangle is the one that maximizes the residual space (RS). In this paper, a RS is defined in the same way as in Lai and Chan (1997): it is the largest rectangular area that can be obtained in a free area, where the two areas have at least one mutual edge. In so doing, the chance that subsequent rectangles can fit into the space is maximized. The hypothesis also suggests that it is reasonable to sort the input rectangles in an order of big to small (Hopper & Turton, 2001): pack the big ones before the small ones.

Guided by the hypothesis, RSMP first orients each input rectangle so that its height is not longer than its width; then creates three sequences: height descending (break a tie by descending width), width descending (break a tie by descending height) and area descending (a tie is left as is). The sequences are tested for packing and the best result is chosen as the solution. During the processing of each rectangle, every possible packing position is tested to generate a series of RSs. The position that results in the largest RS is chosen as the final decision. If there is a tie, i.e. two positions producing the same largest RS, the final decision will be based on the comparison of the second largest RS; so on and so forth. This comparison strategy is the RSMP policy.

Superficially, the policy of RSMP seems to have little difference from packing a rectangle in the smallest suitable space (Gonçalves, 2007); nevertheless, the example shown in Fig. 1 illustrates that the two strategies produce quite different results. As can be seen, RSs may overlap; thereby a rectangle packed in the smallest space may occupy part of a larger space and break it into smaller pieces. RSMP examines all possible packing positions and takes the one that ensures the largest RS.

Computationally, RSMP has to keep track of all RSs and to perform multiple tests to find the packing position of each rectangle. The time and space overhead must be carefully managed in order to achieve an efficient algorithm. The proposed implementation consists of the following steps for bin packing. Few modifications are needed for strip packing as will be described later.

1	Preprocessing: create three input sequences. For each sequence, run RSMP as follows.
2	Initialize an area-descending RS list with one element: a RS of the bin size.
3	For each rectangle, find the best packing position. A rectangle larger than the bin is discarded.
3.1	Find the smallest suitable RS to pack the rectangle. If not found, a new RS equal to an empty bin is added to the front of the list.
3.2	Test all suitable RSs, from the smallest to the largest. For each RS, try packing the rectangle in eight configurations (4 corners × 2 orientations). Keep track of the best packing position based on the RSMP policy.
3.3	Update the list based on the best packing position.

3.1. Difference process (DP)

RSMP applies DP (Lai & Chan, 1997) to generate RSs from a free space partly occupied by a rectangle. Fig. 2 illustrates this fundamental process, repeatedly used in step 3. A RS is rectangular, with

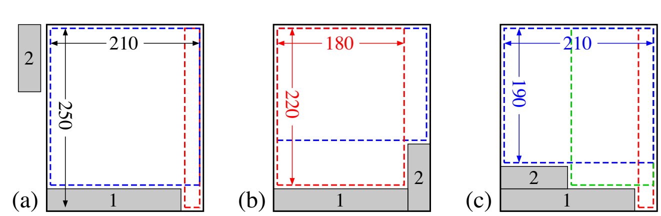

Fig. 1. (a) Two rectangles (the first 180 by 30 and the second 90 by 30) are to be packed in a bin of 210 by 250. The blue and red regions indicate two RSs after the first rectangle is packed. (b) Based on the smallest-suitable-space policy (Gonçalves, 2007), the second rectangle will be packed vertically at the bottom-right corner. The subsequent largest RS is 180 by 220. (c) RSMP will pack the second one horizontally because the largest RS in this position is 210 by 190, larger than that obtained in (b). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)