Computers & Operations Research 40 (2013) 1–8

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

A simple randomized algorithm for two-dimensional strip packing

Shuangyuan Yang [a,*], Shuihua Han [b], Weiguo Ye [b]

[a] *Software school of Xiamen University, 361005 China*
[b] *Department of Management Science, Xiamen University, 361005 China*

ARTICLE INFO

Available online 23 May 2012

Keywords:
Packing problem
Heuristic
Randomized algorithm

ABSTRACT

Two-dimensional strip packing problem is to pack given rectangular pieces on a strip of stock sheet having fixed width and infinite height. Its aim is to minimize the height of the strip such that non-guillotinable and fix orientation constraints are meet. In this paper, an improved scoring rule is developed and the least waste priority strategy is introduced, and a randomized algorithm is presented for solving this problem. This algorithm is very simple and does not need to set any parameters. Computational results on a wide range of benchmark problem instances show that the proposed algorithm obtains a better or matching performance as compared to the most of the previously published meta-heuristics.

1. Introduction

Strip packing problem is a NP-hard problem [1] and has many industrial applications such as packing a given stock material (wood, steel and plastic etc.) on a stock sheet. There exist different types of packing problems because of different constraints and objectives. Some surveys on the cutting and packing problems can be found in Refs. [2–4]. Two-dimensional strip packing problem (2SPP) in this paper is to pack given rectangular pieces on a strip of stock sheet having fixed width and infinite height. Its aim is to minimize the height of the strip. In this paper, piece rotations are not considered and guillotine cuts are not required. Based on an improved typology of the cutting and packing problems provided by Wäscher et al. in Ref. [5], this problem is referred to as a two-dimensional, open dimension problem and is looked on as OF subtype according to the categorization of Lodi et al. in Ref. [6] and Bortfeld in Ref. [7].

Some exact approaches have been presented in the literature for 2SPP. Beasley [8] proposed a tree-search for 2SPP. Martello et al. [9] presented an exact approach for 2SPP, and Lesh et al. [10] introduced a hybrid algorithm by combining a pruning method with a branch and bound. Kenmochi et al. [11] used a branch and bound algorithm based on two placement schemes, and dynamic and linear programming. However, exact algorithms become impractical as the number of rectangular pieces to pack grows. Therefore, many researchers have to investigate heuristics and meta-heuristics to obtain near optimal solutions. Heuristic algorithms [12–18] are simple and fast, however, the quality of solutions cannot be guaranteed so that metaheuristic algorithms become more and more popular. Dagli and Poshyanonda [19] proposed two different hybrid approaches based on artificial neural networks. Simulated annealing, Tabu Search and genetic algorithms [20–26] are widely used for solving the packing problems.

Recently, some excellent metaheuristic algorithms are presented. Alvarez-Valdes et al. [27] introduced GRASP approach for 2SPP, Belov et al. [28] proposed SVC and BS algorithms based on one-dimensional heuristics. Burke et al. [29] proposed a squeaky wheel optimization methodology (SWP) for 2SPP, their algorithm is a simple determined algorithm. Leung et al. (2011) [30] presented a fast determined heuristic algorithm based on level idea and a fitness valuation rule, which is very efficient for large scale problems. Leung et al. [31] developed a two-stage intelligent search algorithm (ISA) based on a simple scoring rule, local search and simulated annealing algorithm. ISA is very efficient and outperforms GRASP and SVC on average. However, ISA is based on simulated annealing algorithm, and its performance depends on the settings of parameters. Wei et al. (2011) [32] developed a complicate metaheuristic algorithm, and obtained the best results for some zero-waste problem instances. Based on the paper [31], we presented a randomized algorithm without setting any parameters. The proposed algorithm is simple and obtains a better or matching performance as compared to the most of the previously published metaheuristic algorithms.

2. A simple randomized algorithm

2.1. ISA

Based on the computational results of ISA, ISA is one of the best algorithms for solving 2SPP [31]. ISA consists of two stages:

* Corresponding author.
E-mail address: yangshuangyuan@xmu.edu.cn (S. Yang).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.05.001

Local search and simulated annealing. Local search just swaps two positions in given sequence in turn, then calling one heuristic algorithm, swapping is accepted if a better solution is obtained, otherwise, swapping does not occur. Simulated annealing algorithm includes a multi-start strategy, its performance depends on two parameters: initial temperature and cooling rate. Two stages of ISA depend on one efficient heuristic packing algorithm. This algorithm selects one rectangular piece by a scoring rule for a given lowest and most left position.

2.2. Improved heuristic algorithm

Heuristic algorithm in ISA is based on a scoring rule. The scoring rule considers five cases. In fact, for $h_1 \geq h_2$, case Fig. 1(4) in Leung et al. [31] includes one special case (2) in Fig. 1 which should be given higher priority than the general case of Fig. 1(4) in Ref. [31]. Where w denotes the width of the current available space s, h_1 denotes the height of the left wall of s, and h_2 denotes the height of the right wall of s. In addition, case (6) in Fig. 1 has the same effect as case (5) in Fig. 1, so it should be considered separately, however, the latter has higher priority than the former because of $h_1 \geq h_2$. In this paper, eight cases are considered, where case (8) in Fig. 1 means the white space is wasted because its width w is less than the width of any unplaced rectangular piece. For $h_1 < h_2$, there exists similar eight cases. Therefore, the improved heuristic algorithm has different scoring rules from ISA in Leung et al. [31]. Similar to ISA, the rectangular piece i with the maximum score is selected to place, if there are several rectangular pieces of the same maximum score, then the first hit is selected for cases (1)–(7) in Fig. 1. Where, for case (7) in Fig. 1, $h_i < \min\{h_1,h_2\}$ is possible.

However, for the selected piece corresponding to one of cases (5), (6) and (7) in Fig. 1, if it wastes some space, then the least waste priority strategy is used to decrease the waste: Let the width of the space is w, for (5) and (6) in Fig. 1, reselect one piece from unplaced pieces, which its height is the same as the height of the selected piece and its width is maximal and is less than w. For example, as shown in Fig. 1(6), if placing the black piece leads to the waste of the white space, reselecting one piece with larger width can decrease the waste if it can be found from unplaced pieces according to the selecting condition. For case (7) in Fig. 1, reselect one piece from unplaced pieces, where its width is maximal and is less than w, and its height is larger than the height of the selected piece. The selected piece is placed if we cannot reselect one. The least waste priority strategy is called after one piece is selected by the scoring rule. For example, assume that the piece i selected by the scoring rule corresponds to case (7), if $w-w_i$ is less than the minimum width of all the unplaced pieces, then the least waste priority strategy is used, and reselects one piece.

2.3. A simple randomized algorithm

Simulated annealing is a powerful randomized algorithm, but its performance significantly depends on the settings of parameters. In particular, how to set the parameter value is a complicated and difficult task. In this paper, we use a simple randomized algorithm without setting any parameters. This simple randomized algorithm (SRA) is as follows:

```
Randomized Algorithm()
  LS()
  **while** running time is less than 60 seconds and the lower
  bound is not found **do**
    **for** $i \leftarrow 1$ **to** $n$ **do**
      randomly select two pieces $j$ and $k$ in $X$;
      obtain a new ordering $X'$ by swapping the order of pieces
  $j$ and $k$;
      $currenth \leftarrow$ HeuristicPacking($X'$);
      **if** currenth < besth **then**
        besth ← currenth;
        $X \leftarrow X'$;
      else
        $p \leftarrow$ currenth/(currenth + besth);
        **if** $p <$ rand(0,1) **then**
          $X \leftarrow X'$;
    randomly select one sorting way by perimeter or area or
  width;
  **return** $besth$;
```

where *besth*, *current* and rand(0,1) are the same as Leung et al. [31]. The sheet is looked on as a strip having infinite height. n is the number of rectangular pieces and X is a given sequence of pieces. p is a real number, the larger p is, the smaller the probability that X' is accepted is. The first step of LS() in this paper is different from that of LS() in Ref. [31], namely, their sorting ways are different. In detail, LS() in this paper is as follows:

```
LS()
  sort all unplaced pieces by non-increasing ordering of height
  and obtain ordering $X$;
  $besth \leftarrow$ HeuristicPacking($X$);
  **for** $i \leftarrow 1$ **to** $n$-1 **do**
    **for** $j \leftarrow i+1$ **to** $n$ **do**
```

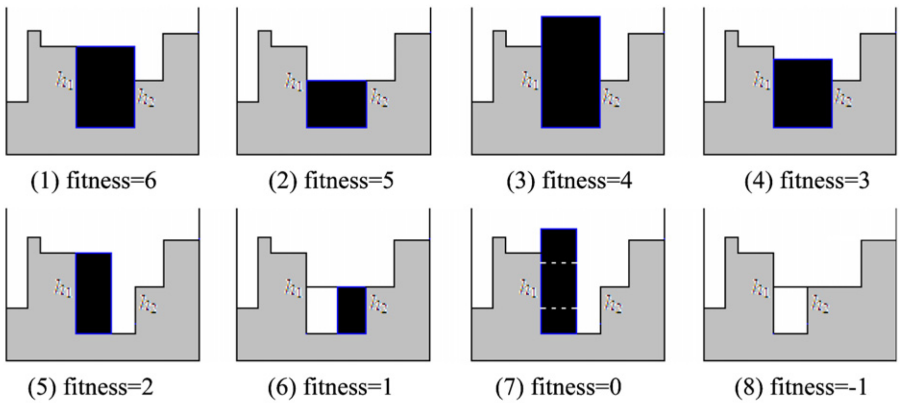

Fig. 1. The scoring rule for $h_1 \geq h_2$.