Computers & Operations Research 36 (2009) 1608 – 1614

Contents lists available at ScienceDirect

Computers & Operations Research

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

# A least wasted first heuristic algorithm for the rectangular packing problem

Lijun Wei[a], Defu Zhang[a,b,*], Qingshan Chen[a]

[a]Department of Computer Science, Xiamen University, Xiamen 361005, China
[b]Longtop Group Post-doctoral Research Center, Xiamen 361005, China

ARTICLE INFO

Available online 14 March 2008

Keywords:
Rectangular packing
Knapsack problem
Heuristic
Local search

ABSTRACT

The rectangular packing problem is to pack a number of rectangles into a single large rectangular sheet so as to maximize the total area covered by the rectangles packed. The paper first presents a least wasted first strategy which evaluates the positions used by the rectangles. Then a random local search is introduced to improve the results and a least wasted first heuristic algorithm (LWF) is further developed to find a desirable solution. Twenty-one rectangular-packing instances are tested by the algorithm developed, the experimental results show that the presented algorithm can achieve an optimal solution within reasonable time and is fairly efficient for dealing the rectangular packing problem. LWF still performs well when it is extended to solve zero-waste and non-zero-waste strip packing instances.



## 1. Introduction

There are many different packing problems in industries with different industries incorporating different constraints and objectives. For example, in shipping encasement, objects of different sizes have to be packed as many as possible into a larger container. In newspaper editor, advertisements and articles have to be arranged in pages. The more extensive and detailed descriptions for packing problems are given in [1–3]. As the importance of these problems, various heuristic algorithms based on different strategies have been presented. These algorithms can be categorized into two categories: the traditional heuristic algorithms and the meta-heuristic algorithms. The traditional heuristic algorithms use the heuristic information to guide the search process. The bottom-left (BL) and BL fill methods [4,5] are the most famous heuristic approaches. Liu et al. presented an improved heuristic algorithm based on BL [6], a best-fit (BF) method was suggested by Burke et al. [7]. The *less flexibility first principle* was introduced by Wu et al. to determine the packing rule [8]. Zhang et al. proposed a new heuristic recursive algorithm [9], which arranges the rectangles using a recursive structure. Huang et al. presented a very effective heuristic algorithm [10], in which two important concepts, namely, the corner-occupying action and caving degree were introduced to guide the packing process. Based on a recursive structure combined with branch-and-bound techniques, Cui et al. presented a new heuristic recursive algorithm [11]. The meta-heuristic algorithms use the meta-heuristic strategies such as simulated annealing, genetic algorithm and artificial neural networks to improve the search results. Hopper et al. gave an empirical investigation of meta-heuristic and heuristic algorithms for 2D packing problem [12]. Zhang et al. presented a meta-heuristic algorithm based on the recursive strategy and simulated annealing algorithm [13]. Artificial neural networks were introduced by Dagli et al. to solve the packing problem [14]. Berthold [15] presented a genetic approach for guillotineable bin packing problem, a genetic algorithm without any encoding of the solutions was presented by Bortfeldt [16]. Heuristic and meta-heuristic approaches for a class of two-dimensional bin packing problems were presented by Lodi et al. [17]. Alvarez-Valdes et al. introduced a GRASP algorithm [18] and a tabu search algorithm [19] for the constrained two-dimensional non-guillotine cutting problems. Beasley [20] presented a population heuristic for the constrained two-dimensional non-guillotine cutting problem. Heuristic approaches for both two and three dimensional knapsack packing problems were proposed by Egeblad et al. [21].

This paper is concerned with the 2D rectangular packing problem which is also called rectangular knapsack problem with the objective of maximizing the area covered by the rectangles packed or filling rate. According to the literature [17], the 2D rectangular packing problem can be similarly categorized into four types: OG, RG, OF, RF. The paper discusses the type of RF: the items may be rotated by 90° and no guillotine constraint is stipulated. However, the items cannot be rotated for non-zero-waste strip packing problem. The paper first presents a least wasted first strategy which evaluates the positions used by the rectangles, and then introduces a random local search to improve the results, finally develops a least wasted first heuristic algorithm (LWF) for the 2D rectangular packing problem. The computational results on a class of benchmark problem instances

* Corresponding author at: Department of Computer Science, Xiamen University, Xiamen 361005, China. Tel.: +86 592 5918207; fax: +86 592 2183502.
*E-mail address:* dfzhang@xmu.edu.cn (D. Zhang).

0305-0548/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2008.03.004

show that LWF can obtain the optimal solutions in short time for these instances. LWF still performs well when it is extended to solve zero-waste and non-zero-waste strip packing instances.

## 2. Problem description

Given a rectangular sheet $B$ with width $W$ and height $H$, a set of $n$ rectangles $R$ with each rectangle $R_i$ of width $w_i$ and height $h_i$ $(1 \leqslant i \leqslant n)$, place the BL of the sheet at origin (0,0) of the two dimensional Cartesian coordinate system and let its four sides parallel to $X$ and $Y$ axis, respectively (Fig. 1). The aim of the 2D rectangular packing problem is to find a packing which maximizes the total area of the rectangles packed into the sheet or filling rate. The packing must satisfy the following constraints:

(1) each rectangle packed should be completely packed within the sheet;
(2) each rectangle can be horizontally or vertically packed into the sheet that means the rectangles are rotatable;
(3) each edge of the rectangles packed should be parallel to an edge of the sheet, which is also called orthogonal packing;
(4) any two rectangles packed should not overlap each other; and
(5) Non-guillotine packing is allowed.

Let $f_i$ $(1 \leqslant i \leqslant n)$ denote whether the sheet $R_i$ has been packed into the sheet or not, if the rectangle $R_i$ has been packed into the sheet then $f_i = 1$, otherwise $f_i = 0$. For every rectangle $R_i$ packed into the sheet, let $(x_{li}, y_{li})$ denote the coordinates of its BL corner, and $(x_{ri}, y_{ri})$ denote the coordinates of its top-right corner. The mathematical formulation of the problem can be described as follows:

$$\max \quad \sum_{i=1}^{n} f_i w_i h_i$$

Subject to

$$(1)\ f_i = 0 \vee (0 \leqslant x_{li} < x_{ri} \leqslant W \wedge 0 \leqslant y_{li} < y_{ri} \leqslant H),\quad i = 1, 2, \ldots, n,$$

$$(2)\ f_i = 0 \vee (x_{ri} - x_{li} = w_i \wedge y_{ri} - y_{li} = h_i) \vee (x_{ri} - x_{li} = h_i \wedge y_{ri} - y_{li} = w_i),\quad i = 1, 2, \ldots, n,$$

$$(3)\ f_i = 0 \vee f_j = 0 \vee (x_{li} \geqslant x_{rj} \vee x_{lj} \geqslant x_{ri} \vee y_{li} \geqslant y_{rj} \vee y_{lj} \geqslant y_{ri}),\quad i, j = 1, 2, \ldots, n, i \neq j,$$

$$(4)\ f_i \in \{0, 1\}, \quad i = 1, 2, \ldots, n,$$

(1) Implies all the rectangles packed are completely in the sheet; (2) implies the rectangles packed are rotatable; (3) implies any two rectangles packed cannot overlap each other; and (4) implies the rectangles can be chosen to pack or not.

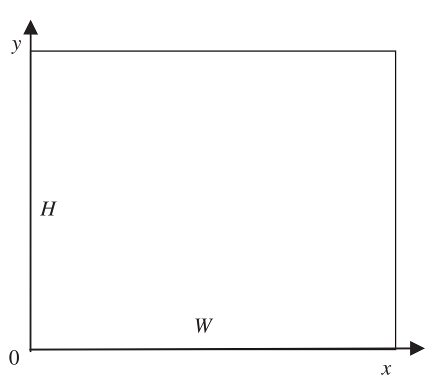


**Fig. 1.**

## 3. Least wasted first heuristic algorithm

### 3.1. Least wasted first strategy

The way of finding positions to pack rectangles is used, which is presented by Martello et al. [22]. The first rectangle is packed with its BL corner at the origin. Let $I$ denote the set of the rectangles packed into the sheet, the remaining rectangles may be packed only at the positions $p$ such that any rectangle in $I$ is below $p$ or at the left of $p$, and cannot overstep the border of the sheet, more formally, the rectangles to be packed can be packed only at the positions contained in the set:

$$S(I) = \{(x, y) : (\forall R_i \in I, x \geqslant x_{ri} \vee y \geqslant y_{ri}) \wedge (x \leqslant W \wedge y \leqslant H)\}.$$

Fig. 2 shows the feasible regions (the area enclosed by the broken line) the remaining rectangles can be packed at the region which is called an *envelope* (the concept of *envelope* is different from that in [22]). It must be noted that we only consider of points where the slope of the *envelope* changes from vertical to horizontal (black points in Fig. 2) [22], these positions are called feasible positions. To check whether a rectangle can be packed at a feasible position, we just need to check if the rectangle oversteps the border of the sheet or not. There is an optimization which can be used to reduce the number of the feasible positions. If the gap between a feasible position and the sheet's border is less than the smallest edge of the unpacked rectangles, this position is called the bad position (see Fig. 3) because none of the unpacked rectangles can be packed at this position. We can discard the bad positions and make some changes to the *envelope*. For example, in Fig. 3, if both $l_1$ and $l_2$ are smaller than the smallest edge of the unpacked rectangles, we discard the bad positions 1 and 4, the changed *envelope* is shown in Fig. 4 (the black region is discarded).

In addition, we must consider which rectangle and which position should be selected when the number of such rectangle and position is greater than one. Inspired by the experience in daily life, we should make the *envelope* as smooth as possible. The less the changes of the slope of *envelop* from vertical to horizontal are, the smoother the *envelope* is. So we should develop a way to measure the smooth of a placement. We can see that each feasible position is formed by a horizontal line and a vertical line. When a rectangle is packed at a position, if its width is equal to the horizontal line or its height is equal to the vertical line, we call this pack a *good pack* (Fig. 5). We use a variable *goodness number* (*GN*) to evaluate such pack. If both the width is equal to horizontal line and the height is equal to vertical line, *GN* of this pack will be 2; if only one of them is equal, then *GN* will be 1; otherwise *GN* will be 0. We will select the placement with larger value of *GN* when the wasted area is same. Therefore we can describe the way of selecting rectangle and position as follows:

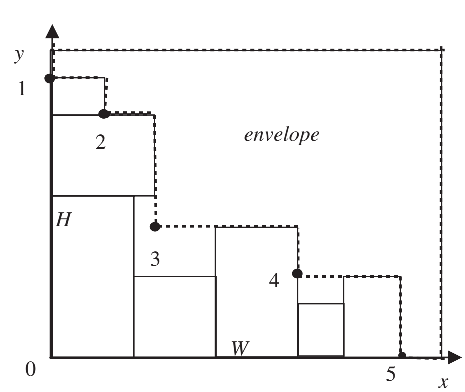


**Fig. 2.** Feasible positions.