ELSEVIER

European Journal of Operational Research 141 (2002) 241–252

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Invited Review

Two-dimensional packing problems: A survey

Andrea Lodi *, Silvano Martello, Michele Monaci

Dipartimento di Elettronica, Informatica e Sistemistica, University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy

Received 9 March 2001

Abstract

We consider problems requiring to allocate a set of rectangular items to larger rectangular standardized units by minimizing the waste. In *two-dimensional bin packing problems* these units are finite rectangles, and the objective is to pack all the items into the minimum number of units, while in *two-dimensional strip packing problems* there is a single standardized unit of given width, and the objective is to pack all the items within the minimum height. We discuss mathematical models, and survey lower bounds, classical approximation algorithms, recent heuristic and metaheuristic methods and exact enumerative approaches. The relevant special cases where the items have to be packed into rows forming levels are also discussed in detail. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Two-dimensional packing; Bin packing problems; Strip packing problems

1. Introduction

In several industrial applications one is required to allocate a set of rectangular items to larger rectangular standardized stock units by minimizing the waste. In wood or glass industries, rectangular components have to be cut from large sheets of material. In warehousing contexts, goods have to be placed on shelves. In newspapers paging, articles and advertisements have to be arranged in pages. In these applications, the standardized stock units are rectangles, and a common objective function is to pack all the requested items into the minimum number of units: the resulting optimization problems are known in the literature as two-dimensional bin packing problems. In other contexts, such as paper or cloth industries, we have instead a single standardized unit (a roll of material), and the objective is to obtain the items by using the minimum roll length: the problems are then referred to as two-dimensional strip packing problems. As we will see in the following, the two problems have a strict relation in almost all algorithmic approaches to their solution.

Most of the contributions in the literature are devoted to the case where the items to be packed have a fixed orientation with respect to the stock unit(s), i.e., one is not allowed to rotate them. This case, which is the object of the present article, reflects a number of practical contexts, such as the cutting of corrugated or decorated material (wood, glass, cloth industries), or the newspapers paging.

* Corresponding author. Tel.: +39-051-209-3029; fax: +39-051-209-3073.

E-mail addresses: alodi@deis.unibo.it (A. Lodi), smartello@deis.unibo.it (S. Martello), mmonaci@deis.unibo.it (M. Monaci).

PII: S0377-2217(02)00123-6

For variants allowing rotations (usually by 90°) and/or constraints on the items placement (such as the "guillotine cuts"), the reader is referred to Lodi et al. [41,42], where a three-field classification of the area is also introduced. General surveys on cutting and packing problems can be found in Dyckhoff and Finke [17], Dowsland and Dowsland [16] and Dyckhoff et al. [18]. Results on the probabilistic analysis of packing algorithms can be found in Coffman and Shor [12] and Coffman and Lueker [11].

Let us introduce the problems in a more formal way. We are given a set of n rectangular *items* $j \in J = \{1, \dots, n\}$, each defined by a *width*, w_j, and a *height*, h_j:

(i) in the *Two-Dimensional Bin Packing Problem* (2BP), we are further given an unlimited number of identical rectangular *bins* of width W and height H, and the objective is to allocate all the items to the minimum number of bins;
(ii) in the *Two-Dimensional Strip Packing Problem* (2SP), we are further given a bin of width W and infinite height (hereafter called *strip*), and the objective is to allocate all the items to the strip by minimizing the height to which the strip is used.

In both cases, the items have to be packed with their w-edges parallel to the W-edge of the bins (or strip). We will assume, with no loss of generality, that all input data are positive integers, and that $w_j \leqslant W$ and $h_j \leqslant H$ $(j = 1, \dots, n)$.

Both problems are strongly NP-hard, as is easily seen by transformation from the strongly NP-hard (one-dimensional) *Bin Packing Problem* (1BP), in which n items, each having an associated size h_j, have to be partitioned into the minimum number of subsets so that the sum of the sizes in each subset does not exceed a given capacity H.

A third relevant case of rectangle packing is the following. Each item j has an associated profit $p_j > 0$, and the problem is to select a subset of items, to be packed in a single finite bin, which maximizes the total selected profit. This problem is usually denoted as (*Two-Dimensional*) *Cutting Stock*, although it had been introduced by Gilmore and Gomory [29] as (*Two-Dimensional*) *Cutting Knapsack*.

In this survey we concentrate on two-dimensional problems in which all items have to be packed, i.e., on 2SP and 2BP. The reader is referred to Dyckhoff et al. [18, Section 5] for an annotated bibliography on two-dimensional cutting stock problems. For both 2SP and 2BP, we also consider the special case where the items have to be packed into rows forming levels.

In Section 2 we discuss mathematical models for the various problems introduced above. In Section 3 we survey classical approximation algorithms as well as more recent heuristic and metaheuristic methods. In Section 4 we introduce lower bounding techniques, while in Section 5 we describe exact enumerative approaches.

2. Models

2.1. Modeling two-dimensional problems

The first attempt to model two-dimensional packing problems was made by Gilmore and Gomory [29], through an extension of their approach to 1BP (see [27,28]). They proposed a column generation approach (see [53] for a recent survey) based on the enumeration of all subsets of items (*patterns*) that can be packed into a single bin. Let A_j be a binary column vector of n elements a_{ij} $(i = 1, \dots, n)$ taking the value 1 if item i belongs to the jth pattern, and the value 0 otherwise. The set of all feasible patterns is then represented by the matrix A, composed by all possible A_j columns $(j = 1, \dots, M)$, and the corresponding mathematical model is

$$\text{(2BP-GG)} \qquad \min \sum_{j=1}^{M} x_j \tag{1}$$

subject to

$$\sum_{j=1}^{M} a_{ij} x_j = 1 \quad (i = 1, \dots, n), \tag{2}$$

$$x_j \in \{0, 1\} \quad (j = 1, \dots, M), \tag{3}$$

where x_j takes the value 1 if pattern j belongs to the solution, and the value 0 otherwise. Observe that (1)–(3) is a valid model for 1BP as well, the only difference being that the A_j's are all columns satisfying $\sum_{i=1}^{n} a_{ij}h_i \leqslant H$.

Due to the immense number of columns that can appear in A, the only way for handling the model is to dynamically generate columns when needed. While for 1BP Gilmore and Gomory [27,28] had given a dynamic programming approach to generate columns by solving, as a *slave* problem, an associated 0–1 knapsack problem, for 2BP they observed the inherent difficulty of the two-dimensional associated problem. Hence they switched to the more tractable case where the items have to be packed in rows forming levels (see Section 2.2), for which the slave problem was solved through a two-stage dynamic programming algorithm.

Beasley [4] considered a two dimensional cutting problem in which a profit is associated with each item, and the objective is to pack a maximum profit subset of items into a single bin (*cutting stock problem*). He gave an ILP formulation based on the discrete representation of the geometrical space and the use of coordinates at which items may be allocated, namely

$$x_{ipq} = \begin{cases} 1 & \text{if item } i \text{ is placed with its bottom left-hand corner at } (p,q), \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

for $i = 1,\dots,n$, $p = 0,\dots,W - w_i$ and $q = 0,\dots,H - h_i$. A similar model, in which p and q coordinates are handled through distinct decision variables, has been introduced by Hadjiconstantinou and Christofides [33]. Both models are used to provide upper bounds through Lagrangian relaxation and subgradient optimization.

A completely different modeling approach has been recently proposed by Fekete and Schepers [20], through a graph-theoretical characterization of the packing of a set of items into a single bin. Let $G_w = (V, E_w)$ (resp. $G_h = (V, E_h)$) be an interval graph having a vertex v_i associated with each item i in the packing and an edge between two vertices (v_i, v_j) if and only if the projections of items i and j on the horizontal (resp. vertical) axis overlap (see Fig. 1). It is proved in [20] that, if the packing is feasible then

(a) for each stable set S of G_w (resp. G_h), $\sum_{v_i \in S} w_i \leqslant W$ (resp. $\sum_{v_i \in S} h_i \leqslant H$);
(b) $E_w \cap E_h = \emptyset$.

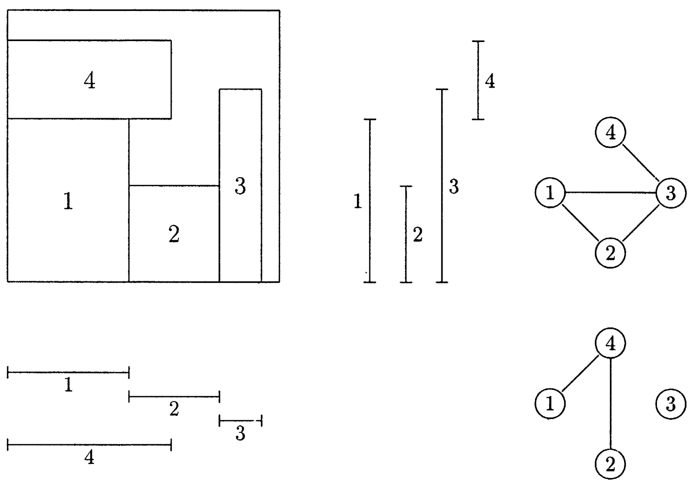

Fig. 1. The Fekete and Schepers modeling approach.

This characterization easily extends to packings in higher dimensions.

2.2. ILP models for level packing

All the above models involve a non-polynomial number of variables. (See Chen et al. [8] for polynomial but practically less effective models.) Effective ILP models involving a polynomial number of variables and constraints have been recently obtained by Lodi et al. [44] for the special case where the items have to be packed "by levels".

As will be seen in the next section, most of the approximation algorithms for 2BP and 2SP pack the items in rows forming *levels*. The first level is the bottom of the bin (or strip), and items are packed with their base on it. The next level is determined by the horizontal line drawn on the top of the tallest item packed on the level below, and so on (see Fig. 2(a)). Let us denote by 2LBP (resp. 2LSP) problem 2BP (resp. 2SP) restricted to this kind of packing.

We assume in the following, without loss of generality, that (see Fig. 2(b))

(i) in each level, the leftmost item is the tallest one;
(ii) in each bin/strip, the bottom level is the tallest one;
(iii) the items are sorted and re-numbered by non-increasing h_j values.

We will say that the leftmost item in a level (resp. the bottom level in a bin/strip) *initializes* the level (resp. the bin/strip).

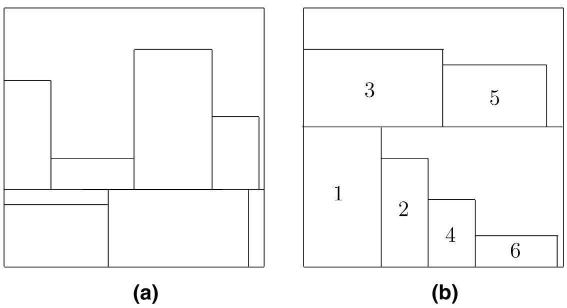

Fig. 2. (a) Level packing; (b) normalized level packing.

Problem 2LBP can be efficiently modeled by assuming that there are n potential levels (the ith one associated with item i initializing it), and n potential bins (the kth one associated with potential level k initializing it). Hence let y_i, $i \in J$ (resp. q_k, $k \in J$) be a binary variable taking the value 1 if item i initializes level i (resp. level k initializes bin k), and the value 0 otherwise. The problem can thus be modeled as

$$(2\text{LBP}) \qquad \min \sum_{k=1}^{n} q_k \tag{5}$$

subject to

$$\sum_{i=1}^{j-1} x_{ij} + y_j = 1 \quad (j = 1, \ldots, n), \tag{6}$$

$$\sum_{j=i+1}^{n} w_j x_{ij} \leqslant (W - w_i) y_i \quad (i = 1, \ldots, n-1), \tag{7}$$

$$\sum_{k=1}^{i-1} z_{ki} + q_i = y_i \quad (i = 1, \ldots, n), \tag{8}$$

$$\sum_{i=k+1}^{n} h_i z_{ki} \leqslant (H - h_k) q_k \quad (k = 1, \ldots, n-1), \tag{9}$$

$$y_i, x_{ij}, q_k, z_{ki} \in \{0, 1\} \quad \forall i, j, k, \tag{10}$$

where x_{ij}, $i \in J \setminus \{n\}$ and $j > i$ (resp. z_{ki}, $k \in J \setminus \{n\}$ and $i > k$) takes the value 1 if item j is packed in level i (resp. level i is allocated to bin k), and the value 0 otherwise. Restrictions $j > i$ and $i > k$ easily follow from assumptions (i)–(iii) above. Eqs. (6) and (8) impose, respectively, that each item is packed exactly once, and that each used level is allocated to exactly one bin. Eqs. (7) and (9) impose, respectively the width constraint to each used level and the height constraint to each used bin.

By modifying the objective function and eliminating all constraints (and variables) related to the packing of the levels into the bins, we immediately get the model for 2LSP:

$$(2\text{LSP}) \qquad \min \sum_{i=1}^{n} h_i y_i \tag{11}$$

subject to

$$(6), (7),$$

$$y_i, x_{ij} \in \{0, 1\} \quad \forall i, j. \tag{12}$$

Computational experiments have shown that the two models are quite useful in practice. Their direct use with a commercial ILP solver produces very good solutions (and, in many cases, the optimal solution) to realistic sized instances within short CPU times. In addition, several variants of 2LBP and 2LSP can be easily handled by modifying some of the constraints, or by adding linear constraints to the models.

The two mathematical models can also be used to produce lower bounds, by relaxing the integrality requirements of the variables (see Section 4.2).

3. Approximation algorithms

In this section we concentrate on *off-line* algorithms, i.e., algorithms which have full knowledge of the input. For *on-line* algorithms, which pack the items in the order they are encountered in the scan of the input (without knowledge the next items), the reader is referred to the survey by Csirik and Woeginger [13]. In the next two sections we consider classical constructive heuristics for 2SP and 2BP, whereas metaheuristic approaches are presented together in Section 3.3.

3.1. Strip packing

Coffman et al. [10] extended two classical approximation algorithms for 1BP to the two-dimensional strip packing problem. Assume that the items are sorted by non-increasing height. The items are packed in levels, as defined in Section 2.2.

The *Next-Fit Decreasing Height* (NFDH) algorithm packs the next item, left justified, on the current level (initially, the bottom of the strip), if it fits. Otherwise, the level is "closed", a new current level is created (as a horizontal line drawn on the top of the tallest item packed on the current level), and the item is packed, left justified, on it.

The *First-Fit Decreasing Height* (FFDH) algorithm packs the next item, left justified, on the first level where it fits, if any. If no level can accommodate it, a new level is created as in NFDH. Fig. 2(b) shows an FFDH packing; for the same set of items, NFDH would close the first level (with items 1 and 2) when packing item 3.

Both algorithms can be implemented to run in $O(n \log n)$ time, through the data structures used by Johnson [36] for their one-dimensional counterparts. Coffman et al. [10] analyzed the worst-case behavior of both algorithms. Let $OPT(I)$ and $A(I)$ denote, respectively, the optimal solution value and the value produced by an approximation algorithm A for an instance I of the problem. It is proved in [10] that, for any instance I, if the heights are normalized to one, then $NFDH(I) \leqslant 2\ OPT(I) + 1$ and $FFDH(I) \leqslant \frac{17}{10} OPT(I) + 1$, and that both bounds are tight, in the sense that the multiplicative constants are the smallest possible.

Further observe that a third approach can be derived from the one-dimensional case: The *Best-Fit Decreasing Height* (BFDH) algorithm packs the next item, left justified, on that level, among those that can accommodate it, for which the residual horizontal space is a minimum. If no level can accommodate it, a new one is created as in NFDH. For the item set of Fig. 2(b), BFDH would pack item 4 in the second level and item 5 in the first one (hence opening a new level for item 6).

As we will see in the following, the algorithms above are also used as a first step in practical approximation algorithms for 2BP.

A different classical approach, which does not pack the items by levels, was defined by Baker et al. [3]. The *Bottom-Left* (BL) algorithm sorts the items by non-increasing width, and packs the current item in the lowest possible position, left justified. It is proved in [3] that $BL(I) \leqslant 3OPT(I)$. This bound too is tight. Chazelle [7] gave an efficient implementation of BL, requiring $O(n^2)$ time.

Other theoretical results on approximation algorithms for 2SP have been obtained, among others, by Sleator [52], Brown [6], Golan [31], Baker et al. [2], Høyland [34], Steinberg [53], Schiermeyer [51].

Recently, Kenyon and Rémila [38] proposed an asymptotic fully polynomial approximation scheme for 2SP: for any given ε, it finds a feasible solution whose value is within a factor of $1 + \varepsilon$ of the optimum (up to an additive term), and runs in time polynomial both in n and $1/\varepsilon$. The scheme, which is based on a new linear programming relaxation, extends a previous work by Fernandez de la Vega and Zissimopoulos [25], and combines techniques developed for 1BP by Fernandez de la Vega and Lueker [24] and by Karmarkar and Karp [37].

3.2. Bin packing

Chung et al. [9] studied the following two-phase approach to 2BP. The first phase of algorithm *Hybrid First-Fit* (HFF) consists of executing algorithm FFDH of the previous section to obtain a strip packing. Consider now the 1BP instance obtained by defining one item per level, with size equal to the level height, and bin capacity H. It is clear that any solution to this instance provides a solution to 2BP. Hence, the second phase of HFF obtains a solution to 2BP by solving the induced 1BP instance through the well-known *First-Fit Decreasing* one-dimensional algorithm (see [36]). The time complexity of the resulting algorithm remains $O(n \log n)$. It is proved in [9] that, if the heights are normalized to one, then $\mathrm{HFF}(I) \leqslant \frac{17}{8}\mathrm{OPT}(I) + 5$. This bound is not proved to be tight: the worst example gives $\mathrm{HFF}(I) = \frac{91}{45}(\mathrm{OPT}(I) - 1)$.

The same idea can also be used in conjunction with NFDH and BFDH, by using, in the second phase, one-dimensional algorithms *Next-Fit Decreasing* (NFD) and *Best-Fit Decreasing* (BFD) (see [36]), respectively. The former algorithm (*Hybrid Next-Fit*, HNF) is equivalent to a single-phase algorithm that packs the next item on the current level of the current bin, if it fits, and otherwise on a new level, created either in the current bin (if possible), or in a new one. Frenk and Galambos [26] analyzed the asymptotic worst-case performance of HNF: if the heights and widths are normalized to one, then $\mathrm{HNF}(I) \leqslant 3.382\mathrm{OPT}(I) + 9$, and the bound is tight (in the sense used above). (The same algorithm was independently considered and experimentally evaluated by Berkey and Wang [5].) The latter approach (BFDH followed by BFD) was implemented by Berkey and Wang [5]. They called it *Finite Best-Strip* (FBS), although, for the sake of uniformity, *Hybrid Best-Fit* (HBF) would be a more appropriate name. Algorithms HNF, HFF and HBF can be implemented so as to require $O(n \log n)$ time.

More recently, Lodi et al. [39,41] proposed different approaches. Their *Floor–Ceiling* (FC) algorithm, in addition to packing the items, from left to right, with their bottom edge on the level floor, also packs items, from right to left, with their top edge touching the level *ceiling*, i.e., the horizontal line drawn on the top of the tallest item packed on the floor. The *Knapsack Packing* (KP) algorithm [41] packs one level at a time by initializing it with the tallest unpacked item, and completing the level packing through the solution of an associated knapsack problem, which maximizes the total area packed in the level. In the second phase of FC and KP, a finite bin solution is obtained by packing the levels into bins, either through algorithm BFD, or by using an exact algorithm for 1BP. The resulting time complexity can thus be non-polynomial, although, in practice, the exact algorithms are halted after a prefixed number of iterations.

Berkey and Wang [5] considered two single-phase approaches. Algorithm *Finite First-Fit* (FFF), a variation of HFF, packs the next item on the lowest level of the first bin where it fits, if any; otherwise, a new level is created either in the first suitable bin, or by initializing a new bin. Algorithm *Finite Bottom-Left* (FBL), a variation of BL, considers the items according to non-increasing width. For each item, the initialized bins are scanned to determine the one which can pack it in the lowest and leftmost position; a new bin is initialized when no feasible position is encountered. Algorithm FFF can be implemented so as to require $O(n \log n)$ time, while algorithm FBL was implemented by using the $O(n^2)$ time approach proposed by Chazelle [7] for BL.

Another algorithm which does not pack the items into levels was recently proposed by Lodi et al. [41]: algorithm *Alternate Directions* (AD) packs the items into non-horizontal *bands*, alternatively from left to right and from right to left. The algorithm has $O(n^3)$ time complexity.

3.3. Metaheuristics

Metaheuristic techniques are nowadays a frequently used tool for the approximate solution of hard combinatorial optimization problems. We refer the reader to Aarts and Lenstra [1] and

Glover and Laguna [30] for general introductions to this area. The impact of these techniques on the practical solution of two-dimensional packing problems has been quite impressive.

Dowsland [15] presented one of the first metaheuristic approaches to 2SP. His simulated annealing algorithm explores both feasible solutions and solutions in which some of the items overlap. During the search, the objective function is thus the total pairwise overlapping area, and the neighborhood contains all the solutions corresponding to vertical or horizontal items shifting. As soon as a new feasible solution improving the current one is found, an upper bound is fixed to its height. Few computational experiments on small-size instances are reported.

Jakobs [35] proposed a genetic algorithm for 2SP. His approach is based on a representation of a packing pattern by means of a permutation giving the order in which the items are packed, while the packing positions are determined through the Bottom-Left strategy (see Section 3.1). This representation turns out to be particularly useful in genetic algorithms for an effective use of crossover and mutation operators. Few numerical examples are reported.

Lodi et al. [39–41] developed tabu search algorithms for 2BP. The main characteristic of the unified framework in [41] is the adoption of a search scheme and a neighborhood which are independent of the specific packing problem to be solved. The framework can thus be used for virtually any variant of 2BP (see Section 1), and has been easily extended to the three-dimensional extension of 2BP (see [43]). Starting from a feasible solution, a move recombines, through a constructive heuristic, a subset of items currently packed into k different bins, plus one item currently packed in a *target bin* (a bin which is more likely to be emptied). The algorithm automatically updates the value of k during the search so as to escape from local optima. A high value of k implies a powerful recombination, but also the obvious drawback of a high computing time for the neighborhood exploration. The only part which depends on the specific problem variant is the constructive heuristic, used to obtain the first solution and to recombine the items at each move. Computational experiments on 500 randomly generated instances with up to 100 items are reported for the classical 2BP and for three variants. (Instances and random generator are available at http://www.or.deis.unibo.it/ORinstances/.)

An extension of the approach by Dowsland [15] to 2BP (as well as to its three-dimensional generalization) was recently proposed by Færø et al. [19]. They use similar neighborhood and search strategy within a *guided local search* approach (see [54] for details). Given a lower bound and an upper bound on the optimal solution value, if these do not coincide, the algorithm randomly assigns the items packed in the highest numbered bin to the other bins. The new solution is generally not feasible, so the new objective function is the total pairwise overlap, plus a term that penalizes, during the search, "unlikely" infeasible patterns. The neighborhood is explored through item shiftings. Minimizing the new objective function corresponds to finding a feasible solution that involves one less bin. The process is iterated until either the upper bound becomes equal to the lower bound, or a prefixed time limit is reached. Specialized techniques to reduce the time complexity for the exploration of this very large neighborhood are implemented. Computational experiments on the set of instances considered in [41] are reported.

4. Lower bounds

4.1. Lower bounds for bin and strip packing

Obvious lower bounds for our problems are obtained by allowing each item to be split into unit squares. We get, respectively for 2BP and 2SP, the *geometric bound* (computable in linear time)

$$L_g^b = \left\lceil \frac{\sum_{j=1}^n w_j h_j}{WH} \right\rceil, \quad L_g^s = \left\lceil \frac{\sum_{j=1}^n w_j h_j}{W} \right\rceil. \tag{13}$$

Let $L(I)$ denote the value produced by a lower bound L for an instance I of the problem. Martello and Vigo [49] proved that, for any instance I, $L_g^b(I) \geqslant \frac{1}{4}\mathrm{OPT}(I)$ while Martello et al. [46] showed that $\max(L_g^s(I), \max_{j\in J}\{h_j\}) \geqslant \frac{1}{2}\mathrm{OPT}(I)$. Both worst-case bounds are tight.

Specialized bounds for 2BP were proposed by Martello and Vigo [49]. Given any integer value q, $1 \leqslant q \leqslant \frac{1}{2}W$, let

$$K_1 = \{j \in J : w_j > W - q\}, \tag{14}$$

$$K_2 = \{j \in J : W - q \geqslant w_j > \tfrac{1}{2}W\}, \tag{15}$$

$$K_3 = \{j \in J : \tfrac{1}{2}W \geqslant w_j \geqslant q\}. \tag{16}$$

As no two items of $K_1 \cup K_2$ may be packed side by side into a bin, a lower bound B^W for the sub-instance given by the items in $K_1 \cup K_2$ is given by any lower bound for the 1BP instance defined by element sizes h_j ($j \in K_1 \cup K_2$) and capacity H (in [49], B^W is computed through the one-dimensional lower bounds by Martello and Toth [48] and Dell'Amico and Martello [14]). By also considering the items in K_3, and observing that none of them may be packed besides an item of K_1, we get the bound

$$B^W(q) = B^W + \max\left\{0, \left\lceil \left\{ \sum_{j \in K_2 \cup K_3} w_j h_j - (HB^W - \sum_{j \in K_1} h_j)W \right\} \Big/ WH \right\rceil \right\}. \tag{17}$$

By interchanging widths and heights, we get a symmetric bound $B^H(q)$, hence an overall lower bound for 2BP:

$$L_2^b = \max\left(\max_{1 \leqslant q \leqslant \frac{1}{2}W} \{B^W(q)\}, \max_{1 \leqslant q \leqslant \frac{1}{2}H} \{B^H(q)\} \right). \tag{18}$$

It is shown in [49] that L_2^b dominates L_g^b (see (13)), and can be computed in $O(n^2)$ time. A computationally more expensive $O(n^3)$ lower bound, L_3^b, which in some cases improves on L_2^b (especially when the instance includes many items with similar sizes), was also given in [49].

Martello et al. [46] proposed specialized lower bounds for 2SP. Assuming that the items are sorted by non-increasing height, define $k = \max\{i : \sum_{j=1}^{i} w_j \leqslant W\}$, and let $i(\ell)$ be the minimum index value such that $w_\ell + \sum_{j=1}^{i(\ell)} w_j > W$ (for each $\ell > k$ satisfying $w_\ell + \sum_{j=1}^{k} w_j > W$). Then a valid lower bound for 2SP is

$$L_1^s = \max\left\{ h_\ell + h_{i(\ell)} : \ell > k \text{ and } w_\ell + \sum_{j=1}^{k} w_j > W \right\}. \tag{19}$$

This bound can be computed in $O(n \log n)$ time, and no dominance relation exists between it and L_g^s. Another lower bound for 2SP was derived from L_2^b (see (18)), by defining, for any integer q in [1], the bound

$$B(q) = \sum_{j \in K_1 \cup K_2} h_j + \max\left(0, \left\lceil \left\{ \sum_{j \in K_3} w_j h_j - \left(\sum_{j \in K_2} (W - w_j) h_j \right) \right\} \Big/ W \right\rceil \right), \tag{20}$$

where K_1, K_2 and K_3 are the sets defined by (14)–(16). The overall lower bound is thus

$$L_2^s = \max_{1 \leqslant q \leqslant W/2} \{B(q)\}. \tag{21}$$

Lower bound L_2^s, which can be computed in $O(n \log n)$ time, dominates L_g^s, while no dominance exists between bounds L_1^s and L_2^s. Finally, a non-polynomial bound for 2SP was obtained in [46], by considering the relaxation obtained by "cutting" each item j ($j = 1, \ldots, n$) into h_j unit-height *slices* of width w_j, and by solving the corresponding 1BP instance (of capacity W) with the additional constraint that, for each item j, the h_j unit-height slices derived from it are packed into h_j *contiguous* one-dimensional bins. The optimal solution of the this problem produces a lower bound L_3^s dominating all bounds above.

Fekete and Schepers [21,23] proposed a general bounding technique for bin and strip packing problems in one or more dimensions, based on *dual feasible functions*. A function $u : [0,1] \to [0,1]$ is called dual feasible (see [45]) if for any finite set S of nonnegative real numbers, we have the relation

$$\sum_{x \in S} x \leqslant 1 \Rightarrow \sum_{x \in S} u(x) \leqslant 1. \tag{22}$$

Consider any 1BP instance, and normalize it by setting $h_j = h_j/H$ ($j = 1, \ldots, n$) and $H = 1$. For any dual feasible function u, any lower bound for the transformed instance having item sizes $u(h_1), \dots, u(h_n)$ is then a valid lower bound for the original instance. In [23] Fekete and Schepers introduced a class of dual feasible functions for 1BP, while in [21] they extended the approach to the packing in two or more dimensions. For a d-dimensional bin packing problem, a set of d dual feasible functions $\{u_1, \dots, u_d\}$ is called a *conservative scale*. Thus, given any conservative scale $\mathscr{C} = \{u_1, u_2\}$, a valid lower bound for 2BP is given by

$$L(\mathscr{C}) = \sum_{j=1}^{n} u_1(w_j)u_2(h_j), \tag{23}$$

where the h_j and w_j values are assumed to be normalized as shown above. Given a set $\mathscr{V}$ of conservative scales, a valid lower bound is

$$L^b = \max_{\mathscr{C} \in \mathscr{V}} L(\mathscr{C}). \tag{24}$$

For 2SP, as no restriction is given on the height, we can assume $u_2(h_j) = h_j$ for any j, and compute a lower bound L^s from (23) and (24). Fekete and Schepers [21] gave dual feasible functions for which L^b dominates L_3^b by Martello and Vigo [49], and L^s dominates L_2^s (see (21)).

4.2. Lower bounds for level packing

The mathematical models of Section 2.2 produce *continuous bounds* for 2LBP and 2LSP by relaxing the integrality requirements of the variables. Let L_c^b and L_c^s denote the lower bounds obtained, respectively for 2LBP and 2LSP, by rounding up to the closest integer the solution values of the resulting linear programs. It is proved in [44] that these bounds dominate the geometric bounds, i.e., that $L_c^b \geqslant L_g^b$ (resp. $L_c^s \geqslant L_g^s$) for any instance of 2LBP (resp. 2LSP).

Lodi et al. [44] also proposed combinatorial bounds that dominate the corresponding continuous bounds. Both bounds are based on the relaxation obtained by allowing item splitting. For 2LSP, any item is allowed to be split into two *slices* of integer width through a vertical cut. For 2LBP, any level is in addition allowed to be split into two *sectors* of integer height through a horizontal cut. It is shown in [44] that both relaxations can be solved exactly in $O(n \log n)$ time through the following simple algorithms.

Let us first consider the 2LSP relaxation, and assume that the items are sorted by non-increasing h_j values. Initialize the first level at height h_1, consecutively pack into it items $1, 2, \dots,$ until the first item i is found which does not fit. Split item i into two slices: one having the width needed to exactly fill the current level, and one for initializing, at height h_i, the next level. Proceed in the same way until all items are packed.

Let us now consider the 2LBP relaxation. The algorithm consists of two steps. First, the 2LSP relaxation is solved through the above algorithm. Then, the resulting levels are consecutively packed in the first bin, starting from the bottom, until the first level is found which does not fit. This level is horizontally split into two sectors: one having the height needed to exactly fill the current bin, and one for initializing the next bin. The algorithm proceeds in the same way until all levels are packed.

Let L_{cut}^b (resp. L_{cut}^s) denote the bound obtained for 2LBP (resp. 2LSP). It is proved in [44] that, for any instance I of 2LBP (resp. 2LSP), $L_{\text{cut}}^b(I) \geqslant \frac{1}{4}\text{OPT}(I)$ (resp. $L_{\text{cut}}^s(I) \geqslant \frac{1}{2}\text{OPT}(I)$), and that both worst-case bounds are tight.

5. Exact algorithms

An enumerative approach for finding the optimal solution of 2SP was proposed by Martello et al. [46]. Initially, the items are sorted by non-increasing height, and a reduction procedure determines the optimal packing of a subset of items in a portion of the strip, thus possibly reducing the instance size.

The branching scheme is an adaptation of the branch-and-bound algorithms proposed by Scheithauer [50] and Martello et al. [47] for 2BP. At each decision node the current partial solution packs the items of a subset $I \subset J$. The packing of I defines the so-called *envelope* through a set of $k \leqslant |I| + 1$ candidate positions (*corner points*) for the bottom-left corners of the unpacked items (see Fig. 3). At most $k \cdot |J \setminus I|$ descendant nodes are then generated by placing each item of $J \setminus I$ in all

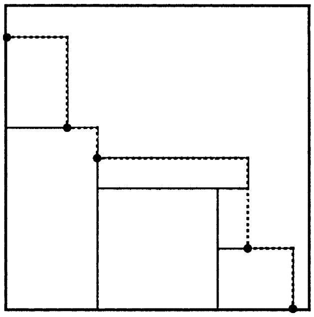

Fig. 3. The envelope associated with the five currently packed items is marked by a dashed line, while black points indicate the corner points.

feasible corner points. The number of nodes is reduced through techniques that avoid the multiple generation of decision nodes producing the same pattern.

Martello and Vigo [49] proposed an enumerative approach to the exact solution of 2BP. The items are initially sorted in non-increasing order of their area, and a reduction procedure tries to determine the optimal packing of some bins. A first incumbent solution, of value z^*, is then heuristically obtained. The algorithm is based on a two-level branching scheme: in the *outer branch-decision tree* the items are assigned to the bins without specifying their position; feasible packings for the items currently assigned to a bin are determined either heuristically or through an *inner branch-decision tree* that enumerates all possible patterns.

The outer branch-decision tree is searched in a depth-first way. At level k ($k = 1, \dots, n$), item k is assigned, in turn, to all the initialized bins and, possibly, to a new one (if the number of bins required by the current partial solution is less than $z^* - 1$).

The feasibility of the assignment of an item k to a bin already packing a set I of items is first heuristically checked in two ways: (i) if any lower bound for the sub-instance defined by $I \cup \{k\}$ has value greater than one, then the assignment is infeasible; (ii) if any upper bound on the same sub-instance has value one, then the assignment is feasible. If both attempts fail, the inner branching scheme enumerates all the possible ways to pack the items of $I \cup \{k\}$ into a single bin through the *left-most downward* principle: there exists an optimal solution in which each item is shifted left and down as much as possible (see [32]). If a feasible packing is found (or if attempt (ii) above is successful) the outer enumeration is resumed; otherwise (or if attempt (i) above is successful), an outer backtracking is performed.

Fekete and Schepers [22] recently developed an enumerative approach, based on their model (see Section 2.1), to the exact solution of the problem of packing a set of items into a single bin. This could be used for alternative exact approaches to 2BP and 2SP. For 2BP, by using it in place of the inner decision-tree in the two-level approach above. For 2SP, by determining, through binary search, the minimum height $\bar{H}$ such that all the items can be packed into a single bin of base W and height $\bar{H}$.

Acknowledgements

We thank the Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) and the Consiglio Nazionale delle Ricerche (CNR), Italy, for the support to the project.

References

[1] E. Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, Wiley, Chichester, 1997.
[2] B.S. Baker, D.J. Brown, H.P. Katseff, A 5/4 algorithm for two-dimensional packing, Journal of Algorithms 2 (1981) 348–368.
[3] B.S. Baker, E.G. Coffman Jr., R.L. Rivest, Orthogonal packing in two dimensions, SIAM Journal on Computing 9 (1980) 846–855.
[4] J.E. Beasley, An exact two-dimensional non-guillotine cutting tree search procedure, Operational Research 33 (1985) 49–64.
[5] J.O. Berkey, P.Y. Wang, Two dimensional finite bin packing algorithms, Journal of the Operational Research Society 38 (1987) 423–429.
[6] D.J. Brown, An improved BL lower bound, Information Processing Letters 11 (1980) 37–39.
[7] B. Chazelle, The bottom-left bin packing heuristic: An efficient implementation, IEEE Transactions on Computers 32 (1983) 697–707.

[8] C.S. Chen, S.M. Lee, Q.S. Shen, A analytical model for the container loading problem, European Journal of Operational Research 80 (1995) 68–76.
[9] F.K.R. Chung, M.R. Garey, D.S. Johnson, On packing two-dimensional bins, SIAM Journal of Algebraic and Discrete Methods 3 (1982) 66–76.
[10] E.G. Coffman Jr., M.R. Garey, D.S. Johnson, R.E. Tarjan, Performance bounds for level-oriented two-dimensional packing algorithms, SIAM Journal on Computing 9 (1980) 801–826.
[11] E.G. Coffman Jr., G.S. Lueker, Probabilistic Analysis of Packing and Partitioning Algorithms, Wiley, Chichester, 1992.
[12] E.G. Coffman Jr., P.W. Shor, Average-case analysis of cutting and packing in two dimensions, European Journal of Operational Research 44 (1990) 134–144.
[13] J. Csirik, G. Woeginger, On-line packing and covering problems, in: Online algorithms, Springer Lecture Notes in Computer Science, vol. 1442, 1996, pp. 147–177.
[14] M. Dell'Amico, S. Martello, Optimal scheduling of tasks on identical parallel processors, ORSA Journal on Computing 7 (1995) 191–200.
[15] K. Dowsland, Some experiments with simulated annealing techniques for packing problems, European Journal of Operational Research 68 (1993) 389–399.
[16] K.A. Dowsland, W.B. Dowsland, Packing problems, European Journal of Operational Research 56 (1992) 2–14.
[17] H. Dyckhoff, U. Finke, Cutting and Packing in Production and Distribution, Physica Verlag, Heidelberg, 1992.
[18] H. Dyckhoff, G. Scheithauer, J. Terno, Cutting and packing (C&P), in: M. Dell'Amico, F. Maffioli, S. Martello (Eds.), Annotated Bibliographies in Combinatorial Optimization, Wiley, Chichester, 1997, pp. 393–413.
[19] O. Færø, D. Pisinger, M. Zachariasen, Guided local search for the three-dimensional bin packing problem, Technical paper, DIKU, University of Copenhagen, 1999.
[20] S.P. Fekete, J. Schepers, On more-dimensional packing I: Modeling, Technical paper ZPR97-288, Mathematisches Institut, Universität zu Köln, 1997.
[21] S.P. Fekete, J. Schepers, On more-dimensional packing II: Bounds, Technical paper ZPR97-289, Mathematisches Institut, Universität zu Köln, 1997.
[22] S.P. Fekete, J. Schepers, On more-dimensional packing III: Exact algorithms, Technical paper ZPR97-290, Mathematisches Institut, Universität zu Köln, 1997.
[23] S.P. Fekete, J. Schepers, New classes of lower bounds for bin packing problems, in: Integer Programming and Combinatorial Optimization (IPCO 98), Springer Lecture Notes in Computer Science, vol. 1412, 1998, pp. 257–270.
[24] W. Fernandez de la Vega, G.S. Lueker, Bin packing can be solved within $1 + \epsilon$ in linear time, Combinatorica 1 (1981) 349–355.
[25] W. Fernandez de la Vega, V. Zissimopoulos, An approximation scheme for strip-packing of rectangles with bounded dimensions, Technical paper, LRI, Université de Paris Sud, Orsay,1991.
[26] J.B. Frenk, G.G. Galambos, Hybrid next-fit algorithm for the two-dimensional rectangle bin-packing problem, Computing 39 (1987) 201–217.
[27] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting stock problem, Operations Research 9 (1961) 849–859.
[28] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting stock problem – part II, Operations Research 11 (1963) 863–888.
[29] P.C. Gilmore, R.E. Gomory, Multistage cutting problems of two and more dimensions, Operations Research 13 (1965) 94–119.
[30] F. Glover, M. Laguna, Tabu Search, Kluwer Academic Publishers, Boston, 1997.
[31] I. Golan, Performance bounds for orthogonal oriented two-dimensional packing algorithms, SIAM Journal on Computing 10 (1981) 571–582.
[32] E. Hadjiconstantinou, N. Christofides, An exact algorithm for general, orthogonal, two-dimensional knapsack problems, European Journal of Operational Research 83 (1995) 39–56.
[33] E. Hadjiconstantinou, N. Christofides, An exact algorithm for the orthogonal, 2-D cutting problems using guillotine cuts, European Journal of Operational Research 83 (1995) 21–38.
[34] S. Høyland, Bin-packing in 1.5 dimension, in: Proceedings of the Scandinavian Workshop on Algorithm Theory, Springer Lecture Notes in Computer Science, vol. 318, 1988, pp. 129–137.
[35] S. Jacobs, On genetic algorithms for the packing of polygons, European Journal of Operational Research 88 (1996) 165–181.
[36] D.S. Johnson, Near-optimal bin packing algorithms, Ph.D. Thesis, MIT, Cambridge, MA, 1973.
[37] N. Karmarkar, R.M. Karp, An efficient approximation scheme for the one-dimensional bin-packing problem, in: Proceedings of the 23rd Annual IEEE Symposium on Found. Comput. Sci., 1982, pp. 312–320.
[38] C. Kenyon, E. Rémila, A near-optimal solution to a two-dimensional cutting stock problem, Mathematics of Operations Research 25 (2000) 645–656.
[39] A. Lodi, S. Martello, D. Vigo, Neighborhood search algorithm for the guillotine non-oriented two-dimensional bin packing problem, in: S. Voss, S. Martello, I.H. Osman, C. Roucairol (Eds.), Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Kluwer Academic Publishers, Boston, 1998, pp. 125–139.
[40] A. Lodi, S. Martello, D. Vigo, Approximation algorithms for the oriented two-dimensional bin packing problem, European Journal of Operational Research 112 (1999) 158–166.
[41] A. Lodi, S. Martello, D. Vigo, Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems, INFORMS Journal on Computing 11 (1999) 345–357.

[42] A. Lodi, S. Martello, D. Vigo, Recent advances on two-dimensional bin packing problems, Discrete Applied Mathematics 123/124 (2002) 373–380.
[43] A. Lodi, S. Martello, D. Vigo, Heuristic algorithms for the three-dimensional bin packing problem, European Journal of Operational Research, this issue.
[44] A. Lodi, S. Martello, D. Vigo, Models and bounds for two-dimensional level packing problems, Journal of Combinatorial Optimization, to appear.
[45] G.S. Lueker, Bin packing with items uniformly distributed over intervals [a,b], in: Proceedings of the 24th Annual Symposium on Found. Comp. Sci., 1983, pp. 289–297.
[46] S. Martello, M. Monaci, D. Vigo, An exact approach to the strip packing problem, Technical paper OR/00/18, Dipartimento di Elettronica, Informatica e Sistemistica, Università di Bologna, 2000.
[47] S. Martello, D. Pisinger, D. Vigo, The three-dimensional bin packing problem, Operations Research 48 (2000) 256–267.
[48] S. Martello, P. Toth, Knapsack Problems: Algorithms and Computer Implementations, Wiley, Chichester, 1990.
[49] S. Martello, D. Vigo, Exact solution of the two-dimensional finite bin packing problem, Management Science 44 (1998) 388–399.
[50] G. Scheithauer, Equivalence and dominance for problems of optimal packing of rectangles, Ricerca Operativa 83 (1997) 3–34.
[51] I. Schiermeyer, Reverse fit: A 2-optimal algorithm for packing rectangles, in: Proceedings of the 2nd Eur. Symposium Algorithms (ESA), 1994, pp. 290–299.
[52] D. Sleator, A 2.5 times optimal algorithm for packing in two dimensions, Information Processing Letters 10 (1980) 37–40.
[53] A. Steinberg, A strip-packing algorithm with absolute performance bound 2, SIAM Journal on Computing 26 (1997) 401–409.
[54] C. Voudouris, E. Tsang, Guided local search and its application to the traveling salesman problem, European Journal of Operational Research 113 (1999) 469–499.