CIRRELT

CIRRELT-2026-12

Two-Dimensional Guillotine Cutting Problem for Large Objects with Non-Rectangular Shapes

Carise E. Schmidt
Arinei C. L. Silva
Leandro C. Coelho
Maryam Darvish

March 2026

Bureau de Montréal

Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec

Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

Two-Dimensional Guillotine Cutting Problem for Large Objects with Non-Rectangular Shapes

Carise E. Schmidt[1*], Arinei C. L. Silva[2], Leandro C. Coelho[3,4], Maryam Darvish[3]

[1] Instituto Federal de Santa Catarina (IFSC), Avenida Nereu Ramos D, 3450, Seminário, Chapecó, 89813-000, Brazil.

[2] Universidade Federal do Paraná (UFPR), Departamento de Engenharia de Produção, Avenida Cel. Francisco H. dos Santos, Jardim das Américas, Curitiba, 81530-900, Brazil.

[3] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Laval University, Department of Operations and Decision Systems, Québec (QC).

[4] Canada Research Chair in Integrated Logistics, Laval University, Québec (QC).

Abstract. In this paper, we address the two-dimensional single large object placement problem with guillotine cutting constraints, focusing on non-rectangular shapes. Specifically, we consider large objects with circular or convex polygonal geometries, and we study a variant that includes defective regions from which no items can be extracted. Rectangular items are cut from the object using a two-stage guillotine cutting pattern, allowing orthogonal rotations of the items. To solve this problem, we adopt a recursive dynamic programming algorithm that handles the geometric complexities of non-rectangular shapes, including variable cut lengths. For convex polygonal objects, we also introduce a method that optimizes the starting angle of the cutting pattern to maximize material usage and profitability. Computational experiments on various instance sets show that our approach provides optimal solutions for circular objects with high computational efficiency. For polygonal objects, the method provides robust and effective solutions by evaluating multiple rotation angles to determine the optimal cutting orientation. Moreover, our approach can accommodate material imperfections, maintaining high material utilization levels even in the presence of defects.

Keywords: cutting and packing problems; dynamic programming; two-dimensional cutting; circular and polygonal shapes; defects.

Acknowledgements. This project was partly funded by the Canadian Natural Sciences and Engineering Research Council (NSERC) under grants 2025-03964 and 2025-04195.

---

* Corresponding author: carise.schmidt@ifsc.edu.br

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2026

1. Introduction

Cutting and packing problems have long been central to the operations research and production economics literature. Their attractiveness stems from their intricate theoretical nature and their significance across a wide range of industrial settings. Applications arise in paper (Kallrath et al., 2014), wood (Reinders and Hendriks, 1989; Kokten and Çağrı, 2022), steel (Sierra-Paradinas et al., 2021), textile (Hadj Salem et al., 2023), and glass manufacturing (Durak and Aksu, 2017; Parreño and Alvarez-Valdes, 2021), among others, where production constraints and material characteristics have led to a wide variety of problem variants and modeling extensions (Wäscher et al., 2007).

Within this broad family, a widely studied and practically relevant subclass is the "Placement Problem", which involves arranging a set of smaller items within one or more larger objects (Cui et al., 2013). This setting is commonly encountered in industrial applications, such as cutting wood shapes from a large piece or extracting fabric pieces from a roll. The primary objective is typically to maximize the total value or area of the items placed or to minimize the material waste. According to the typology proposed by Wäscher et al. (2007), placement problems can vary depending on the characteristics of the large objects (e.g., single vs. multiple), the problem's dimensionality (one-dimensional, two-dimensional (2D), or three-dimensional), and the geometric formats of both the small and large items. These problems exhibit strong similarities to linear and 2D knapsack problems (Wei and Lim, 2015), differing mainly in the heterogeneity of the item set, which is often more diverse in placement settings. Hence, many solution approaches developed for knapsack problems are also applicable to placement problems (Iori et al., 2021).

Over time, researchers have introduced numerous variants to reflect real production requirements (Gilmore and Gomory, 1965). For the small items to be extracted, the problem is classified as *constrained* when the number of items of each type is limited; otherwise, it is considered *unconstrained* (Yao et al., 2024). For the unconstrained case, dynamic programming is a natural and effective solution approach. However, it is not straightforward to extend dynamic programming techniques to the constrained version of the problem (Clautiaux et al., 2018). Nonetheless, solutions to the unconstrained case can serve as proper upper bounds on the profit in the constrained setting. Like the cutting-stock problem (Yu et al., 2009), placement problems also impose strict non-overlapping constraints. Besides, small items may be either fixed in orientation or allowed to rotate, often restricted to orthogonal (90-degree) rotations to simplify the cutting process and maintain feasibility.

Extracting small items often involves cutting large objects in a 2D setting. In this process, specific cutting constraints may be imposed, one of the most common being the *guillotine cutting*. Under this constraint, each cut must go from one edge of the large object to the opposite edge, producing sub-rectangles that can be further divided (Martin et al., 2020b). When guillotine cuts are required, an additional constraint may be introduced to limit the number of cutting *stages*. Each stage consists of a set of parallel guillotine cuts applied to the pieces generated in the previous stage. Depending on whether the number of stages is fixed, the problem is referred to as a *k-staged guillotine cutting problem*; otherwise, it is called *non-staged*. Finally, when trimming is permitted, i.e., when it is acceptable for the final layout to contain some unused space, the problem is considered to be in a *non-exact* setting.

The geometry of both the small items and the large object is commonly assumed to be rectangular, a simplification assumption that significantly eases problem modeling and solution design (Cheng et al., 1994). While rectangular items are representative of certain conventional manufacturing processes, this assumption does not hold in many practical scenarios where the raw material has a *non-rectangular* shape. An additional level of complexity arises when the large object contains *defective regions*, which must be excluded from the cutting plan. Such defects can occur in various industries: knotholes in wood panels used in furniture manufacturing, imperfections in flat glass production, cracks and fissures in marble or granite slabs, or irregularities in leather used for fashion goods (Martin et al., 2020b, 2022). Taking these unusable zones into account is essential for generating realistic cutting plans, since the optimization algorithm must avoid positioning items over defective areas and adapt the cutting pattern accordingly.

In this paper, we address the Two-Dimensional Guillotine Single Large Object Placement Problem (2D-GSLOPP) for circular and polygonal shapes, including cases with defects. Our key contributions are threefold. First, we extend existing models by explicitly handling non-rectangular geometries and material defects, which are rarely addressed together in the literature. Second, we adapt a recursive dynamic programming algorithm to efficiently generate cutting patterns in both circular and polygonal formats, managing geometric complexities such as variable cut lengths. Third, for polygonal objects, we introduce a novel angle-optimization strategy to determine the best orientation for initiating the cutting process and to improve material profitability. These contributions offer practical value to industries that work with irregular raw materials, such as stone, leather, or glass, by enabling more realistic, defect-aware, and geometry-sensitive cutting plans.

The remainder of this paper is organized as follows. Section 2 reviews the literature on exact methods and mathematical approaches related to the problem, including works that address different shape formats and the presence of defective regions. Section 3 presents the formal problem description for the specific cases considered in this study. The proposed recursive dynamic programming approach is detailed in Section 4. Section 5 reports the computational experiments conducted to evaluate the performance of the algorithm. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Literature review

We start our review in Section 2.1 discussing the classical guillotine cutting for rectangular objects classifications. Section 2.2 describes both exact and heuristic algorithms for solving these problems. In Section 2.3, we review the literature on cutting problems with defects, and in Section 2.4, we cover circular and convex shapes for the objects.

2.1. Classical guillotine cutting for rectangular objects

Most studies on 2D cutting and packing problems with guillotine constraints assume that the container (large object) is rectangular (López-Camacho et al., 2013). Only a few explore alternative geometries, and those that do typically consider such formats without guillotine constraints.

Since the seminal work of Gilmore and Gomory (1965), which introduced several multi-dimensional cutting problems, numerous studies have focused on 2D guillotine cutting. Most of them stem from various applications in cutting and packing operations. To generate cutting patterns using at most two stages of guillotine cuts, Gilmore and Gomory (1965, 1966) proposed dynamic programming procedures based on the knapsack function. In their approach, the first stage determines a set of parallel cutting patterns to produce longitudinal strips. In contrast, the second stage specifies how each strip should be further divided using transversal parallel cuts. This methodology has received considerable attention in the literature, as it produces optimal solutions for subproblems in which the number of items to be extracted is unbounded. Moreover, it is widely used to compute upper bounds for the constrained case, where the number of items of each type is limited.

Fig. 1 presents the rectangular two-stage guillotine setting. Panel (a) displays three alternative strip widths for the first stage; rectangles in the same shade indicate the orthogonal rotation of the same item type. Observe that there are two item types (light gray and dark gray) that can be rotated by 90°. Panel (b) shows one feasible second-stage layout within the selected strips; the white upper-right region indicates unused stock.

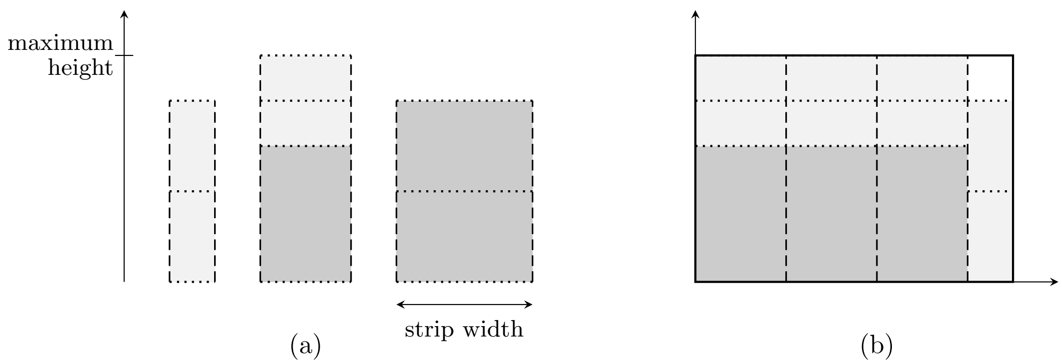

Figure 1: Rectangular two-stage guillotine with orthogonal rotations

In addition to permitting orthogonal item rotation, the rectangular two-stage guillotine setting is commonly classified as either exact (no trimming) or non-exact (trimming permitted). Fig. 2 shows these two cases under the same strip configuration. Observe that on panel (a), the guillotine cuts yield the exact objects being cut, while on panel (b), one would need an extra cut to obtain the final object.

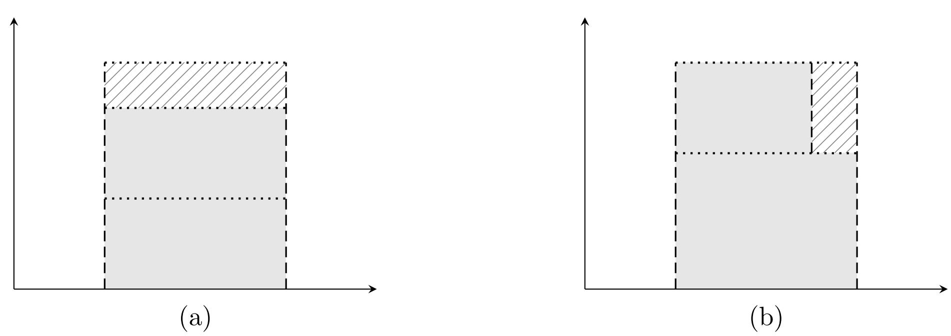

Figure 2: Exact and non-exact cases for a specific width class

Herz (1972) introduced the concept of canonical cutting patterns, defined as orthogonal guillotine cuts restricted to discretized coordinates derived from item dimensions, which reduce the computational effort to solve the Single Large Object Placement Problem (SLOPP). In the unconstrained case, Herz (1972) proposed a recursive algorithm, while Christofides and Whitlock (1977) developed an exact tree search method for the same problem. Later, Beasley (1985) proposed both exact and heuristic methods that incorporate discretized points into the dynamic programming framework initially developed by Gilmore and Gomory (1965). Hifi (2001) studied guillotine cutting problems by extending the Gilmore and Gomory approach and presented an exact algorithm capable of solving the unconstrained two-stage SLOPP.

2.2. Exact and heuristic methods for the rectangular single large object placement and the single knapsack problems

Building on the Gilmore and Gomory framework, several exact and heuristic approaches have been developed for the 2D-SLOPP and Single Knapsack Problem (SKP).

Lodi and Monaci (2003) presented integer linear programming models for the 2D-SLOPP, considering both two-stage and three-stage guillotine cuts. Their formulation is extended to handle several variants, including the unconstrained case and the possibility of orthogonal rotation of the small items. To solve the constrained two-stage SLOPP, Hifi and M'Hallah (2005) proposed an exact branch-and-bound algorithm based on a bottom-up approach. Guillotine cutting can follow a top-down approach, where the stock is divided to fit items, or a bottom-up approach, where items are combined to reconstruct the stock (Martin et al., 2021). Other exact methods were introduced by Cintra et al. (2008), addressing multiple 2D guillotine cutting problems, including the SLOPP and its variants, which require staged patterns. A primal-dual heuristic was also proposed by Morabito and Pureza (2010) to address the same class of problems.

Dolatabadi et al. (2012) proposed an exact recursive procedure combined with an implicit enumeration of all feasible patterns to generate the set of guillotine cuts for the 2D guillotine SKP. To solve large-scale instances of the unconstrained case, Russo et al. (2014) introduced an enhanced dynamic programming algorithm that reduces the search space and avoids redundant solutions. Furini et al. (2016) presented integer linear programming formulations for the constrained SKP with unrestricted-stage guillotine cuts, relying on a pseudo-polynomial number of variables and constraints. For the same problem, Clautiaux et al. (2018) proposed a hypergraph-based model, where a cutting pattern is represented as a flow in a directed acyclic hypergraph. Their model was also extended to incorporate different variants, such as the staged case. To address the constrained 2D guillotine cutting problem more broadly, Velasco and Uchoa (2019) developed exact tree search algorithms with improved bounds, using state-space relaxations and dynamic programming. More recently, Wang et al. (2025) proposed an exact algorithm for the 2D guillotine SKP that integrates advanced bounding and search strategies, achieving competitive results on benchmark instances and providing open-source resources to support future research.

2.3. Cutting problems with defects

There are relatively few studies in the literature addressing the 2D cutting problem with defects. Hahn (1968) and Scheithauer and Terno (1988) proposed dynamic programming approaches to generate cutting patterns for the non-exact case, where trimming is permitted. Afsharian et al. (2014) studied the unconstrained variant of the problem with multiple defects and unrestricted-stage guillotine cuts, also using a dynamic programming method. An instance generator for the unconstrained 2D guillotine cutting problem with defects was introduced by Neidlein et al. (2016), enabling the creation of test cases with realistic defect configurations. Finally, Durak and Aksu (2017) provided a broader discussion on various applications of cutting and packing problems with defective regions.

In a two-stage guillotine cutting context, when a defective region is present, one possible practice is to isolate the defect by a short sequence of vertical and horizontal guillotine cuts. Fig. 3 illustrates two viable sequences, highlighting that the resulting sub-rectangles may differ: (a) vertical-first, then horizontal; (b) horizontal-first, then vertical.

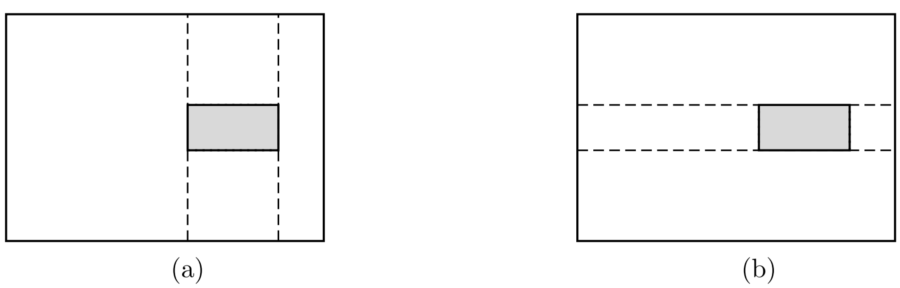

Figure 3: Two-stage guillotine cutting with isolation of a defective region

To address a cutting problem with defects arising in the glass industry, Parreño et al. (2020) proposed a beam search heuristic as a solution method. Parreño and Alvarez-Valdes (2021) introduced the first integer linear programming model capable of handling both trimming and defective regions. A variant of the constrained 2D-SLOPP with defects and non-staged guillotine patterns was studied by Martin et al. (2020b). They proposed a compact integer linear programming formulation based on the discretization of the defective object, along with a Benders decomposition algorithm and a three-phase constraint programming-based approach. Later, Martin et al. (2022) studied two additional variants of the 2D guillotine cutting problem with multiple defects. To solve these problems, the authors extended the integer linear programming and constraint programming models originally introduced in Martin et al. (2020a). Zhang et al. (2023) addressed the unconstrained 2D cutting problem with defects, considering both guillotine and non-guillotine variants. They proposed an integer linear model for the non-guillotine case and an exact dynamic programming algorithm for the guillotine case. Yao et al. (2024) propose a Benders decomposition approach for the constrained 2D non-guillotine cutting problem with defects, combining a one-dimensional bin packing master problem with a 2D packing subproblem solved via integer programming. Later in Yao et al. (2025), the same authors study the 2D non-guillotine strip packing problem with defects, placing non-rotatable rectangular items using a two-stage exact method based on integer programming, Benders decomposition, and a heuristic.

2.4. Circular and convex shapes

Studies involving circular or convex-shaped objects have also been proposed. Considering circular shapes, Reinders and Hendriks (1989) presented a case study of converting trees into lumber, modeled as a three-level guillotine cutting stock problem. In their approach, the first level determines where to segment the tree into logs, the second level cuts the logs into flitches, and the third level selects cuts to maximize the value obtained from the flitches. The authors developed a solution algorithm based on nested dynamic programming. The 2D-SKP, in which rectangular items must be orthogonally placed into a circular container to maximize the total packed area, was studied by López and Beasley (2018) without guillotine constraints. They formulated the problem as an MILP and solved both rotated and non-rotated variants. For the same problem, Bouzid and Salhi (2020) proposed a heuristic approach that iteratively packs items to form a polygon fitting within the circular container. This heuristic was embedded into two metaheuristics: a variable neighborhood search and a simulated annealing, which explore different item packing orders. More recently, Silva et al. (2022) introduced the first linear programming formulation for this problem and proposed a cutting plane algorithm. In addition, they developed a parallel enumeration algorithm that generates all non-dominated subsets of items and checks whether each subset fits within the circular container. Their methods improved upon previous results, providing better solutions for large-scale instances.

Considering the orthogonal packing of rectangular items within convex areas, Birgin et al. (2006) modeled the problem without accounting for guillotine constraints and proposed a solution approach based on nonlinear programming. For the same problem, Cassioli and Locatelli (2011) developed a heuristic method grounded in an iterated local search framework, where the key mechanism is a perturbation step to escape local optima.

For a comprehensive survey of exact and heuristic solution approaches for the constrained 2D-SLOPP with guillotine cutting, see Russo et al. (2020). Oliveira et al. (2023) also provided an overview of solution methods developed for two-dimensional rectangular cutting and packing problems, including guillotine variants of both the SKP and SLOPP. To the best of our knowledge, no approach in the literature addresses the guillotine cutting problem within convex polygonal shapes while explicitly incorporating guillotine constraints. We also refer the reader to Francescatto and Júnior (2025), which provides a systematic review of industrial aspects of cutting-related applications, focusing on two-dimensional bin packing, cutting stock, and open-dimension problems.

3. Problem description

In this section, we present the 2D-GSLOPP, considering two types of large objects: circles and convex polygons. We begin with a generalized formulation that applies to both shapes, followed by specific procedures tailored to each one. Furthermore, for the convex polygon configuration, we extend the analysis to account for the presence of a defect.

Given a single large object S with a convex shape and a set I of small rectangular items to be cut from S, each item $i \in I$ is characterized by a height h_i, a width w_i, and a profit p_i. We assume that the dimensions of the single large object and the small items (h_i and w_i) are all integers. If this is not the case, an equivalent integral instance can be obtained by an appropriate scaling. The objective is to determine a cutting pattern for S that maximizes the sum of the profits of the cut items, subject to several constraints:

(i) items must not overlap;

(ii) all cuts must follow a guillotine pattern;

(iii) at most two guillotine stages are allowed, with cuts in successive stages being orthogonal;

(iv) the cutting plan must be designed without trimming;

(v) small items may be orthogonally rotated;

(vi) there is no upper bound on the number of items of each type that can be cut.

It is important to note that, given the shape of the large object under consideration, the initial cut in each guillotine stage must follow a linear trajectory that starts and ends at the boundaries of the object. However, unlike subsequent cuts, this initial cut cannot be uniformly described as being parallel to one of the object's sides.

Without loss of generality, we assume that all cuts are infinitely thin. This assumption does not impose a significant limitation, as it is well established that a problem involving all cuts of finite (integer) width can be converted into a problem with infinitely thin cuts by adding the cutting width to the h_i and w_i of each small item $i \in I$.

Each cut of each guillotine stage can be seen as a segment parallel or perpendicular to one of the axes. For a rectangular shape and a vertical first-stage cut, each cut divides the current shape into two smaller, rectangular strips. These rectangles may have different widths W but share the same height H, meaning that the lengths of all cuts generated in this stage are the same. However, this is not the case for a circle or a convex polygon. Therefore, we need to consider the lengths of these cuts in such cases.

Let the single large object S be a circle with radius r, as shown in Fig. 4. Note that, in this case, during vertical cutting (parallel to the y-axis), the length varies depending on the location of the cut.

Considering a cut orthogonal to the x-axis, at each position x, $0 < x < 2r$, its length $h(x)$ can be calculated using the Pythagorean Theorem (see Fig. 4(b)), as $h(x) = 2\sqrt{r^2 - (r-x)^2}$.

For the second case, let the single large object be any convex polygon, defined by a set of Cartesian coordinates of its vertices $V = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, as shown in Fig. 5(a). As for the circle, the length also varies depending on where the cut is made (see Fig. 5(b)).

Considering the geometric position of the shape previously assumed, we can see the maximal variation of each coordinate (x, y) from the vertices of the polygon as an interval of length $W = x_{max}$ and $H = y_{max}$, respectively, to x and y. Taking into account a cut orthogonal to the x-axis, at each position x, $0 < x < x_{max}$, its length can be calculated as follows. Each side of the polygon is

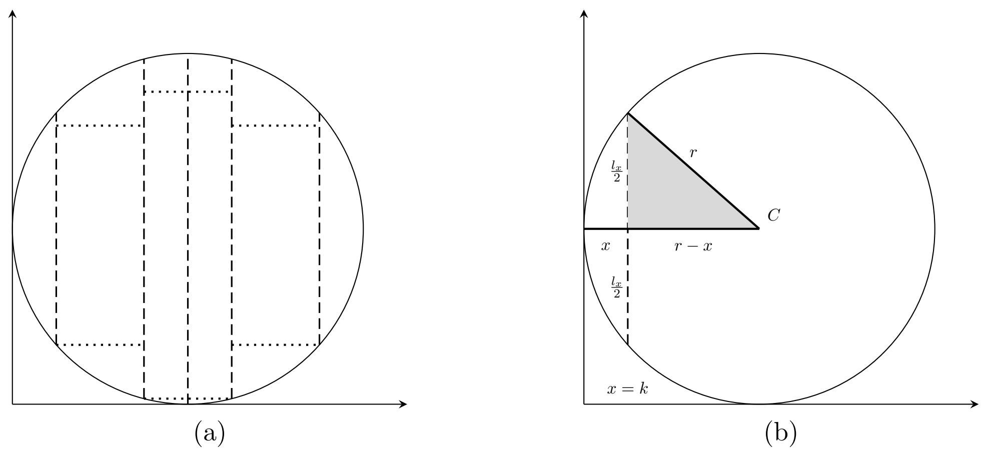

Figure 4: Guillotine cutting for a circular shape

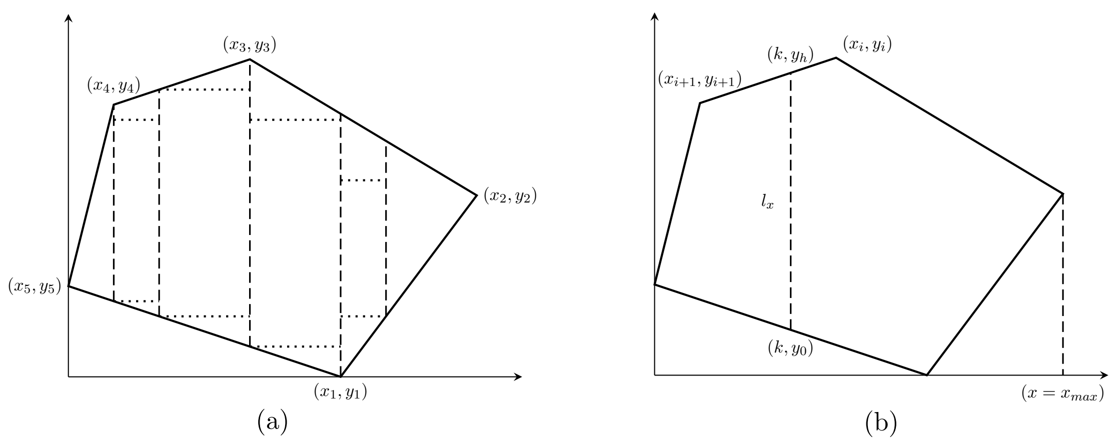

Figure 5: Guillotine cutting for a convex polygon shape

a segment formed by two consecutive vertices, (x_i, y_i) and (x_{i+1}, y_{i+1}). To calculate the length of the cut at a given position $x = k, 0 < k < x_{max}$, it is necessary to consider all pairs of consecutive vertices, (x_i, y_i) and (x_{i+1}, y_{i+1}), for which $(x_i - k)(x_{i+1} - k) \leq 0$. For each pair of vertices that satisfies this condition, there is an intersection point, which represents one extremity of the cut. The y value of the intersection point (k, y) can be obtained as $y(k) = y_i + t(y_{i+1} - y_i)$, where $t = \dfrac{(k - x_i)}{(x_{i+1} - x_i)}$, with $0 \leq t \leq 1$.

Using the absolute difference between the coordinates y from the start (k, y_a) and end (k, y_b) intersection points, we obtain the cut length $h(x)$ for a given point $x = k$. Let us now assume the polygonal shape can be rotated. Note that the rotation of the polygon does not alter its format, only its geometric position in the Cartesian plane. However, in this problem, the cutting pattern may vary depending on the position fixed during the first cut, i.e., determined by the chosen rotation angle (see Fig. 6). Thus, in this case, we consider planning the cut for different geometric rotations of the convex polygon.

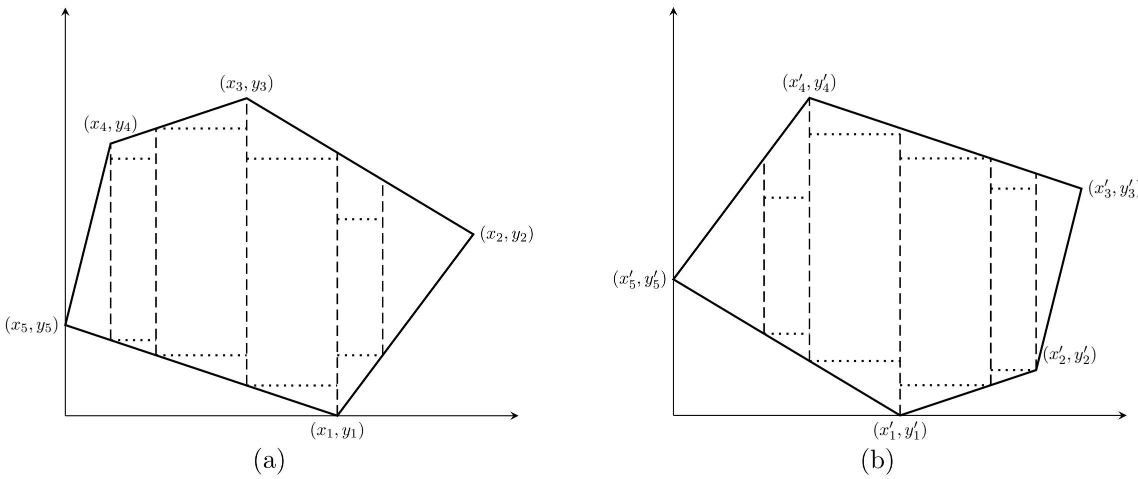

Figure 6: Guillotine cutting for different convex polygon positions

To rotate the polygon by an angle α (in radians) around the origin, we apply the rotation matrix, according to equation 1.

$$R(\alpha) = \begin{bmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{bmatrix}. \tag{1}$$

Thus, each vertex (x_i, y_i) can be transformed into the rotated vertex (x'_i, y'_i) using equation 2.

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = R(\alpha) \begin{bmatrix} x_i \\ y_i \end{bmatrix}. \tag{2}$$

Consequently, the rotated polygon for a given angle α is given by $V(\alpha) = \{(x'_1, y'_1), (x'_2, y'_2), ..., (x'_n, y'_n)\}$. Note that, for each different angle α, $0 \leq \alpha \leq 2\pi$, the respective cutting plane may have a different profit value of the extracted items.

For the convex polygon case, we also consider an extension in which the single large object has a defective region, modeled as a rectangle. Let D be the set of cartesian coordinates of this defective rectangle $D = \{(x_1, y_1), (x_2, y_1), (x_2, y_2), (x_1, y_2)\}$. In this case, we first isolate the defective region by performing guillotine cuts along its sides. Consequently, four new, smaller polygons are generated. Depending on the sequence of cuts performed, these new polygons can have different shapes, which affects the profitability of the cutting plan, both in terms of each new polygon and the overall utilization of the original profitable area. For this scenario, we consider two possibilities using the guillotine cutting. In one of them, we subdivide the polygon by cutting along the base of the defective rectangle (i.e., along its parallel side); then, we execute two orthogonal cuts on the remaining sides to completely isolate the defect. For the other scenario, we reverse the order of the guillotine cuts, starting from the two sides orthogonal to the base of the defective rectangle. Fig. 7 illustrates both situations.

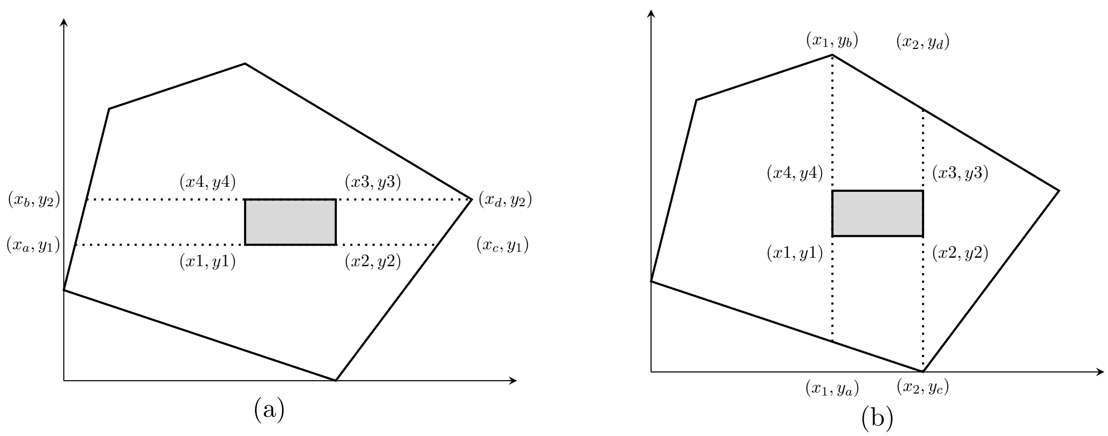

Figure 7: Guillotine cutting to isolate the defect in a convex polygon shape

Once the defect is isolated, we can repeat the process described above on each new polygon to determine its optimal cutting pattern. The way we choose to isolate the defect is the one that maximizes the overall utilization of the entire shape, considering the four smaller convex polygons generated.

4. Solution approach

To determine the most profitable cutting pattern for each shape, we use an adapted dynamic programming recursion, based on the approach presented by Gilmore and Gomory (1965) and Beasley (1985). We denote $x_{max} = W$ and $y_{max} = Y$, respectively, the maximum x and y values from the vertices of the shape.

Let P be the set of all available widths to cut a strip vertically, and Q_w all available heights for a specific width w. Moreover, let $v(w, y_1)$, $y_1 \in Q_w$, be the profit obtained by extracting a piece of dimensions w and y_1; and $F(w, y)$ be the maximum profit obtained by processing a subproblem defined by specific values of w and y. The recursion function (3) defines the profit that can be obtained:

$$F(w,y) = \begin{cases} 0, & \text{if } y < min\{y' \in Q_w\}, \\ \max\limits_{y' \in Q_w,\, y' \leq y} \Big\{ v(w,y') + F(w, y - y') \Big\}, & \text{otherwise.} \end{cases} \quad (3)$$

For vertical cuts, we introduce the recursive function $G(x)$, which represents the optimal value obtained for a subproblem that has width x. As the available height depends on the position x, we define $h(x)$ as a function that provides the available height for a given x. However, for a subproblem extending from $x-w$ to x, the available effective height is given by $min\{h(x-w), h(x)\}$. To ensure that the height value belongs to Q_w, we define the feasible height as a function of these positions, given by $y(x-w,x) = max\{y_1 \in Q_w \mid y_1 \leq min\{h(x-w), h(x)\}$.

Then, we use the previous recursion function $F(w,y)$ to process a vertical cut with width w and height y. Note that extracting a strip of width x of some point of the shape implies evaluating the available height at two positions: $x-w$ and x. Thus, the vertical recursive function (4) is given as follows.

$$G(x) = \begin{cases} 0, & \text{if } x < min\{w \in P\}, \\ \max\limits_{w \in P,\, w \leq x} \Big\{ F(w, y(x-w,x) + G(x-w) \Big\}, & \text{otherwise.} \end{cases} \quad (4)$$

The complete solution to the problem is obtained by computing $G(x_{max})$, which recursively determines the profitability of the optimal cutting strategy. For each format shape that we address, the value of $h(x)$ is calculated according to the description in Section 3.

Algorithm 1 summarizes the recursive dynamic programming approach.

5. Computational experiments

This section describes the detailed experiments we conducted to assess the performance and capabilities of our algorithms. The algorithms are coded in C++, and all computational experiments are performed on a machine equipped with an Intel Core i7 processor running at 3.4 GHz with 64 GB of RAM, with the Ubuntu Linux operating system. Section 5.1 describes the sets of instances used. The details of computational experiments and results for the circular case are provided in Section 5.2, and in Section 5.3 for the convex polygon shape.

Algorithm 1 Recursive DP algorithm for guillotine cutting for circular and polygonal shapes

Input: *Shape data (radius or vertices) and item types with width set P, admissible heights Q_w for each $w \in P$, profits $v(w, y)$, and height profile $h(x)$ on $x \in [0, x_{\max}]$.*
Output: *Optimal profit from guillotine cutting.*
Define effective available height $y(x-w, x) \leftarrow \max\{\, y \in Q_w \mid y \le \min\{h(x-w),\, h(x)\}\,\}$**.**
// Horizontal recursion:
Compute $F(w, y)$**, the optimal profit for a piece of width** w **and height** y**.**
for all $w \in P$ and $0 \le y \le y_{\max}$ **do**
  for all $y_1 \in Q_w$ such that $0 < y_1 < y$ **do**
    Compute $F(w, y)$**.**
  end for
end for
// Vertical recursion:
Compute $G(x)$**, the optimal profit for a strip of total width** x**.**
for all $0 \le x \le x_{max}$ **do**
  for all $w \in P$ such that $w \le x$ **do**
    Compute $y(x-w, x)$**.**
    Compute $G(x)$**.**
  end for
end for
Return $G(x_{\max})$**.**

5.1. Instances

For the case in which the single large object is circular, we use three sets of instances generated to solve the problem of orthogonally packing a set of rectangular items into a fixed-size circular container. The first one (set 1C) is a benchmark set from López and Beasley (2018) and contains instances with 10, 20, and 30 items, where the dimensions are randomly generated from the interval $[1, 5]$. For each configuration related to the number of items, there are two different versions: one where all items are rectangles and another where they are squares. The second one (set 2C) is from Bouzid and Salhi (2020), who follow the same procedure as the previous set to generate 18 instances containing 100, 150, and 200 items. The third set (set 3C) is generated by Silva et al. (2022) and contains five instances with 30 items with dimensions randomly generated from the interval $[1, 5]$. In total, 41 instances are used to test the problem when the single large object has a circular shape.

To test the problem using a convex polygon as a single large object, we adapt a set based on the problem instances randomly generated by Afsharian et al. (2014) for the 2D Rectangular SLOPP with defects and non-stage cutting patterns. The original instances have the following characteristics. The rectangular large object sizes ($L \times W$) are: (75×75) and (125×50) in category 1; (150×150) and (225×100) in category 2; (300×300) and (450×200) in category 3. The number of item types, m, is 5, 10, 15, 20, or 25, and the size of items, $l_i \times w_i$, is taken uniformly from the intervals $[L/\varphi, 3L/4]$ and $[W/\varphi, 3W/4]$, with $\varphi = 6, 8$, or 10. The profit p_i of each item is its area, $l_i \times w_i$. Finally, the size of the defect, $l_d \times w_d$, is taken uniformly from the intervals $[L/10, L/6]$ and $[W/10, W/6]$ and the left-lower coordinate of defects, ox_d, oy_d, are taken uniformly from the intervals $[0, L - l_d]$ and $[0, W - w_d]$. The authors divide these instances into 90 classes, each one with one to four defects, and for each combination, they generated 15 instances. We adapt only instances with one defect, resulting in a total of 1,350 instances.

By adapting this set to the polygonal shape format, we preserved the characteristics of the items and the defects from the original instances. Instead of using the original rectangle as the single large object, we create a convex polygon inscribed in a rectangle with the original dimensions of L and W multiplied by a scale factor of 1.6. The number of vertices of the polygon is randomly generated from the interval $[4, 10]$, ensuring that the defect is located within the generated polygon. We use this same set of adapted instances in both cases (with and without the defect); when the defect is not considered, the defect-related information is disregarded. The sets of polygonal instances with and without defects are named, respectively, set 1P and set 1D. All instances and detailed results are available upon request.

5.2. Results for the circular shape

We ran instances of sets 1C, 2C, and 3C using the recursive dynamic programming approach presented in Section 4. Then, we compare the result of our approach to solve the problem described in Section 3 with the profitability of extracting the largest unique square item.

Table 1 presents the average results grouped by instance set and number of items, m, as shown in the first two columns. Then, we report the profitability (i.e., the percentage of total area used), processing time (in seconds), and improvement (in %) when extracting small items instead of the largest square item. Each value in the table represents the average over all instances with the same number of items within the same set.

These results show that a cutting plan for the addressed problem can be generated in a small fraction of a second, even for instances with up to 200 distinct items. Extracting only the largest possible square item from the circular boundary yields an area utilization of about two-thirds of the total. Compared with this single-item approach, our method achieves an improvement in area utilization of at least 20%, and in some cases exceeding 50%. For the tested instances, the

Table 1: Average results obtained by the proposed algorithm for a large circular object.

Set	m	Profitability (%)	Time (s)	Improvement (%)
1C	10	76.53	0.02	20.22
1C	20	85.09	0.03	33.66
1C	30	87.68	0.03	37.73
Average		83.10	0.02	30.54
2C	100	94.82	0.18	48.95
2C	150	96.00	0.25	50.80
2C	200	96.72	0.35	51.93
Average		95.85	0.26	50.56
3C	30	91.53	0.08	43.78
Global Average		89.77	0.13	41.01

improvement tends to grow with the number of items considered, and is larger when many small objects are available.

5.3. Results for the polygonal shape

For the polygonal shape case, we tested the instance set 1P using the solution approach described in Section 4. Since the solution depends on the initial orientation of the convex polygon, we vary the rotation angle α with three step sizes: 10°, 1°, and 0.1°.

The comparative average results of these experiments are reported in Table 2. The first two columns indicate the instance category and the number of items. Then, for each rotation angle step, we report the profitability (Prof.), i.e., the percentage of total area used by our solution approach, followed by its processing time (Time). For step sizes of 1° and 0.1°, we also show the improvement relative to the previous step size. Each value in Table 2 represents the average over all instances belonging to the same category and having the same number of items.

The results in Table 2 show that, for the tested instances, the average profitability is rather similar across the three categories for all rotation step sizes, with minimal gains with finer step sizes.

In Category 1, the average profitability is 81.26% with a 10° step, increasing to 82.36% with a 1° step, an average improvement of 1.40%. Refining the step to 0.1° further increases profitability to 82.57%, an additional improvement of 0.26% over the 1° step.

In Category 2, the average profitability values are 82.61%, 82.62%, and 82.81% for steps of 10° ,

Table 2: Average results of the proposed algorithm for a large polygonal object, according to the rotation angle.

Category	m	α step = 10°		α step = 1°			α step = 0.1°		
		Prof. (%)	Time (s)	Prof. (%)	Time (s)	Impr.(%)	Prof. (%)	Time (s)	Impr.(%)
1	5	81.29	0.02	82.56	0.05	1.59	82.73	0.38	0.21
1	10	85.28	0.02	86.28	0.08	1.18	86.42	0.71	0.17
1	15	78.64	0.02	79.72	0.06	1.41	80.03	0.55	0.39
1	20	78.68	0.02	79.83	0.04	1.53	80.00	0.34	0.21
1	25	82.39	0.02	83.43	0.06	1.27	83.70	0.54	0.33
Average		81.26	0.02	82.36	0.06	1.40	82.57	0.50	0.26
2	5	81.06	0.02	82.09	0.09	1.26	82.26	0.81	0.21
2	10	85.06	0.03	86.02	0.17	1.13	86.27	1.63	0.29
2	15	79.35	0.03	80.60	0.13	1.71	80.72	1.26	0.15
2	20	79.85	0.02	80.65	0.08	1.03	80.82	0.75	0.21
2	25	82.76	0.03	83.75	0.14	1.21	83.98	1.28	0.28
Average		81.61	0.03	82.62	0.12	1.27	82.81	1.15	0.23
3	5	80.95	0.03	81.94	0.17	1.24	82.12	1.64	0.21
3	10	85.55	0.05	86.52	0.37	1.14	86.74	3.57	0.25
3	15	79.92	0.04	80.94	0.29	1.34	81.05	2.81	0.13
3	20	79.78	0.03	80.70	0.18	1.17	80.85	1.65	0.18
3	25	82.87	0.04	83.96	0.28	1.33	84.17	2.72	0.25
Average		81.82	0.04	82.81	0.26	1.24	82.99	2.48	0.21
Global average		81.56	0.03	82.60	0.15	1.30	82.79	1.38	0.23

1° , and 0.1°, respectively, corresponding to improvements of 1.27% and 0.23% when reducing the step size from 10° to 1°, and from 1° to 0.1°.

In Category 3, decreasing the step from 10° to 1° increases the average profitability from 81.82% to 82.81%, a gain of 1.24%. Reducing it further to 0.1° increases profitability to 82.99%, an additional 0.21% improvement.

Overall, using a 0.1° step instead of 10° or 1° increases the global average profitability by 1.30% and 0.23%, respectively.

Processing time increases with both the object category and the rotation resolution. For a 10° step, the global average time is 0.03 seconds, increasing to 0.15 seconds for a 1° step, and 1.38 seconds for a 0.1° step. Despite this increase, from 0.03 to 1.38 seconds, the approach remains computationally efficient.

5.3.1. Results for the polygonal shape with defect

We also conducted experiments for the polygonal shape with a defect, using the instance set labeled 1D. In this case, two rotation step sizes were tested: 1° and 0.1°. Table 3 reports the average results by instance category and number of item types that can be extracted. As in the previous table, the first two columns list these parameters, followed by the profitability and processing time for each rotation step size. The last column presents the average improvement in profitability obtained when using the smaller step size. As before, each value in Table 3 represents the average over all instances belonging to the same category and with the same number of items.

Table 3: Average results of the proposed algorithm for a large polygonal object with a defect, according to the rotation angle.

Category	m	α step = 10		α step = 1			α step = 0.1		
		Profit.(%)	Time (s)	Profit.(%)	Time (s)	Impr.(%)	Profit.(%)	Time (s)	Impr.(%)
1	5	69.57	0.14	71.07	0.15	2.40	71.32	1.08	0.35
1	10	77.55	0.17	78.60	0.27	1.40	78.82	1.97	0.27
1	15	68.96	0.14	69.92	0.21	1.43	70.16	1.53	0.33
1	20	67.61	0.13	68.43	0.14	1.27	68.74	0.89	0.46
1	25	72.17	0.16	73.04	0.20	1.29	73.27	1.38	0.30
Average		71.17	0.15	72.21	0.19	1.56	72.46	1.37	0.34
2	5	70.24	0.15	71.57	0.27	1.90	71.63	2.27	0.08
2	10	76.52	0.18	78.18	0.49	2.31	78.42	4.50	0.30
2	15	69.27	0.15	70.14	0.39	1.21	70.50	3.51	0.58
2	20	67.30	0.13	68.40	0.26	1.70	68.67	1.95	0.44
2	25	73.15	0.15	74.26	0.38	1.55	74.51	3.30	0.35
Average		71.29	0.15	72.51	0.36	1.73	72.75	3.10	0.35
3	5	70.40	0.17	71.74	0.51	1.86	72.07	4.58	0.46
3	10	76.71	0.22	77.92	1.02	1.63	78.14	9.85	0.29
3	15	70.00	0.23	70.96	0.79	1.37	71.03	7.85	0.09
3	20	68.57	0.18	69.30	0.46	1.08	69.46	4.29	0.21
3	25	73.17	0.21	74.10	0.72	1.29	74.27	7.16	0.23
Average		71.77	0.20	72.81	0.70	1.44	72.99	6.75	0.26
Global average		71.41	0.17	72.51	0.42	1.58	72.73	3.74	0.32

For the tested instances, the results in Table 2 also show rather similar average profitability indices across categories.

In Category 1, the average profitability is 71.17%, 72.21%, and 72.46% for rotation step sizes of 10°, 1°, and 0.1°, respectively. This corresponds to an improvement of 1.56% when reducing the step from 10° to 1°, and 0.34% when reducing it from 1° to0.1°.

In Category 2, the average profitability values are 71.29%, 72.51%, and 72.75% for steps of 10°, 1°, and 0.1°, respectively, yielding improvements of 1.73% and 0.35% for the same reductions in step size.

In Category 3, the profitability is 71.77%, 72.81%, and 72.99% for steps of 10°,1°, and 0.1°, respectively. The corresponding improvements are 1.44% and 0.26%.

Overall, using the smallest step size increases the global average profitability by 1.58% when reducing the step from 10° to 1°, and by 0.32% when reducing it from 1° to 0.1°.

As in the previous tests, processing time increases with instance size (category) and with rotation resolution. The average computational times are 0.17 seconds for a 10° step, 0.42 seconds for a 1° step, and 3.74 seconds for a 0.1°step. Even when testing two alternative ways of isolating the defect, the total average time remains below 10 seconds in all cases, with the maximum observed in Category 3 for $m = 10$ using the smallest step size.

6. Conclusions

This paper presents an exact approach to solve the Two-Dimensional Guillotine Single Large Object Placement Problem to a given precision, considering both circular and convex polygonal shapes. We have also proposed an extension of our method to handle items with an arbitrary rectangular defect. The proposed methodology relies on a dynamic programming recursion that discretizes the problem space efficiently, enabling the rapid generation of optimal cutting patterns. Computational experiments on multiple instance sets show that the proposed approach can produce highly profitable cutting plans within seconds, even for instances involving a large number of distinct items.

The results demonstrate significant improvements in material utilization compared with traditional methods that extract only the largest square item. For circular objects, the proposed approach yields profitability gains ranging from approximately 20% to over 50%, while maintaining very short solution times. For convex polygonal shapes, the analysis shows that reducing the rotation angle step can yield small additional improvements in profitability, albeit with increased processing times. Nonetheless, even with finer rotation steps, the method remains computationally efficient and well-suited for practical scenarios requiring rapid decision-making.

Our algorithm provides a robust, flexible framework for addressing complex cutting problems involving diverse object geometries and defects. Its ability to generate high-quality cutting patterns quickly makes it highly relevant for industrial applications, where maximizing material utilization and ensuring computational efficiency are critical to optimizing production processes and reducing waste.

References

R. P. Abeysooriya, J. A. Bennell, and A. Martinez-Sykora. Jostle heuristics for the 2D-irregular shapes bin packing problems with free rotation. *International Journal of Production Economics*, 195:12–26, 2018. doi: 10.1016/j.ijpe.2017.09.014.

M. Afsharian, A. Niknejad, and G. Wäscher. A heuristic, dynamic programming-based approach for a two-dimensional cutting problem with defects. *OR Spectrum*, 36(1):971–999, 2014. doi: 10.1007/s00291-014-0363-x.

J.E. Beasley. Algorithms for unconstrained two-dimensional guillotine cutting. *The Journal of the Operational Research Society*, 36(4):297–306, 1985. doi: 10.1057/jors.1985.51.

E.G. Birgin, J.M. Martínez, F.H. Nishihara, and D.P. Ronconi. Orthogonal packing of rectangular items within arbitrary convex regions by nonlinear optimization. *Computers & Operations Research*, 33(12):3535–3548, 2006. doi: 10.1016/j.cor.2005.03.031.

M.C. Bouzid and S. Salhi. Packing rectangles into a fixed size circular container: Constructive and metaheuristic search approaches. *European Journal of Operational Research*, 285(3):865–883, 2020. doi: 10.1016/j.ejor.2020.02.048.

C. Carnieri, G.A. Mendoza, and L.G Gavinho. Solution procedures for cutting lumber into furniture parts. *European Journal of Operational Research*, 73(3):495–501, 1994. doi: 10.1016/0377-2217(94)90244-5.

A. Cassioli and M. Locatelli. A heuristic approach for packing identical rectangles in convex regions. *Computers & Operations Research*, 38:1342–1350, 2011. doi: 10.1016/j.cor.2010.12.001.

C. H. Cheng, B. R. Feiring, and T. C .E. Cheng. The cutting stock problem – a survey. *International Journal of Production Economics*, 36(3):291–305, 1994. doi: 10.1016/0925-5273(94)00045-X.

N. Christofides and C. Whitlock. An algorithm for two-dimensional cutting problems. *Operations Research*, 25(1):30–44, 1977. doi: 10.1287/opre.25.1.30.

G.F. Cintra, F.K. Miyazawa, Y. Wakabayashi, and E.C. Xavier. Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. *European Journal of Operational Research*, 191(1):61–85, 2008. doi: 10.1016/j.ejor.2007.08.007.

F. Clautiaux, R. Sadykov, F. Vanderbeck, and Q. Viaud. Combining dynamic programming with filtering to solve a four-stage two-dimensional guillotine-cut bounded knapsack problem. *Discrete Optimization*, 29:18–44, 2018. doi: 10.1016/j.disopt.2018.02.003.

Y. Cui, L. Yang, Z. Zhao, T. Tang, and M. Yin. Sequential grouping heuristic for the two-dimensional cutting stock problem with pattern reduction. *International Journal of Production Economics*, 144(2):432–439, 2013. doi: 10.1016/j.ijpe.2013.03.011.

M. Dolatabadi, A. Lodi, and M. Monaci. Exact algorithms for the two-dimensional guillotine knapsack. *Computers & Operations Research*, 39(1):48–53, 2012. doi: 10.1016/j.cor.2010.12.018.

B. Durak and D.T. Aksu. Dynamic programming and mixed integer programming based algorithms for the online glass cutting problem with defects and production targets. *International Journal of Production Research*, 55(24):7398–7411, 2017. doi: 10.1080/00207543.2017.1349951.

M. Francescatto and A. N. Júnior. Two-dimensional bin packing, cutting stock, and open dimension problems: A survey of practical requirements. *Computers & Operations Research*, 183:107209, 2025. doi: 10.1016/j.cor.2025.107209.

F. Furini, E. Malaguti, and D. Thomopulos. Modeling two-dimensional guillotine cutting problems via integer programming. *INFORMS Journal on Computing*, 28(4):736–751, 2016. doi: 10.1287/ijoc.2016.0710.

P.C. Gilmore and R.E. Gomory. Multistage cutting stock problems of two and more dimensions. *Operations Research*, 13(1):94–120, 1965. doi: 10.1287/opre.13.1.94.

P.C. Gilmore and R.E Gomory. The theory and computation of knapsack functions. *Operations Research*, 14(6):1045–1074, 1966. doi: 10.1287/opre.14.6.1045.

K. Hadj Salem, E. Silva, J.F. Oliveira, and M.A Carravilla. Mathematical models for the two-dimensional variable-sized cutting stock problem in the home textile industry. *European Journal of Operational Research*, 306(2):549–566, 2023. doi: 10.1016/j.ejor.2022.08.018.

S.G. Hahn. On the optimal cutting of defective sheets. *Operations Research*, 16(6):1100–1114, 1968. doi: 10.1287/opre.16.6.1100.

J.C. Herz. Recursive computational procedure for two-dimensional stock cutting. *IBM Journal of Research and Development*, 16(5):462–469, 1972. doi: 10.1147/rd.165.0462.

M. Hifi. Exact algorithms for large-scale unconstrained two and three staged cutting problems. *Computational Optimization and Applications*, 18(1):63–88, 2001. doi: 10.1023/A:1008743711658.

M. Hifi and R. M'Hallah. An exact algorithm for constrained two-dimensional two-staged cutting problems. *Operations Research*, 53(1):140–150, 2005. doi: 10.1287/opre.1040.0154.

M. Iori, V.L. de Lima, S. Martello, F. K. Miyazawa, and M. Monaci. Exact solution techniques for two-dimensional cutting and packing. *European Journal of Operational Research*, 289(2):399–415, 2021. doi: 10.1016/j.ejor.2020.06.050.

J. Kallrath, S. Rebennack, J. Kallrath, and R. Kusche. Solving real-world cutting stock-problems in the paper industry: Mathematical approaches, experience and challenges. *European Journal of Operational Research*, 238(1):374–389, 2014. doi: 10.1016/j.ejor.2014.03.027.

E.S. Kokten and S. Çağrı. A cutting stock problem in the wood products industry: a two-stage solution approach. *International Transactions in Operational Research*, 29(2):879–907, 2022. doi: 10.1111/itor.12802.

A. Lodi and M. Monaci. Integer linear programming models for 2-staged two-dimensional knapsack problems. *Mathematical Programmimg*, 94(1):257–278, 2003. doi: 10.1007/s10107-002-0319-9.

E. López-Camacho, H. Terashima-Marín, G. Ochoa, and S. E. Conant-Pablos. Understanding the structure of bin packing problems through principal component analysis. *International Journal of Production Economics*, 145(2):488–499, 2013. doi: 10.1016/j.ijpe.2013.04.041.

C.O. López and J.E. Beasley. Packing unequal rectangles and squares in a fixed size circular container using formulation space search. *Computers & Operations Research*, 94:106–117, 2018. doi: 10.1016/j.cor.2018.02.012.

E Malaguti, R. Medina Durán, and P. Toth. Approaches to real world two-dimensional cutting problems. *Omega*, 47:99–115, 2014. doi: 10.1016/j.omega.2013.08.007.

M. Martin, E.G. Birgin, R.D. Lobato, R. Morabito, and P. Munari. Models for the two-dimensional rectangular single large placement problem with guillotine cuts and constrained pattern. *International Transactions in Operational Research*, 27(2):767–793, 2020a. doi: 10.1111/itor.12703.

M. Martin, P.H.D.B. Hokama, R. Morabito, and P. Munari. The constrained two-dimensional guillotine cutting problem with defects: an ILP formulation, a Benders decomposition and a CP-based algorithm. *International Journal of Production Research*, 58(9):2712–2729, 2020b. doi: 10.1080/00207543.2019.1630773.

M. Martin, R. Morabito, and P. Munari. A bottom-up packing approach for modeling the constrained two-dimensional guillotine placement problem. *Computers & Operations Research*, 115: 104851, 2020c. doi: 10.1016/j.cor.2019.104851.

M. Martin, R. Morabito, and P. Munari. A top-down cutting approach for modeling the constrained two- and three-dimensional guillotine cutting problems. *Journal of the Operational Research Society*, 72(12):2755–2769, 2021. doi: 10.1080/01605682.2020.1813640.

M. Martin, R. Morabito, and P. Munari. Two-stage and one-group two-dimensional guillotine cutting problems with defects: a CP-based algorithm and ILP formulations. *International Journal of Production Research*, 60(6):1854–1873, 2022. doi: 10.1080/00207543.2021.1876270.

R. Morabito and M. Arenales. Optimizing the cutting of stock plates in a furniture company. *International Journal of Production Research*, 38(12):2725–2742, 2000. doi: 10.1080/002075400411457.

R. Morabito and V. Pureza. A heuristic approach based on dynamic programming and and/or-graph search for the constrained two-dimensional guillotine cutting problem. *Annals of Operations Research*, 179(1):297–315, 2010. doi: 10.1007/s10479-008-0457-4.

V. Neidlein, A. Scholz, and G. Wäscher. SLOPPGEN: a problem generator for the two-dimensional rectangular single large object placement problem with defects. *International Transactions in Operational Research*, 23(1-2):173–186, 2016. doi: 10.1111/itor.12119.

O. Oliveira, D. Gamboa, and E. Silva. An introduction to the two-dimensional rectangular cutting and packing problem. *International Transactions in Operational Research*, 30(6):3238–3266, 2023. doi: 10.1111/itor.13236.

F. Parreño and R. Alvarez-Valdes. Mathematical models for a cutting problem in the glass manufacturing industry. *Omega*, 103:102432, 2021. doi: 10.1016/j.omega.2021.102432.

F. Parreño, M.T. Alonso, and R Alvarez-Valdes. Solving a large cutting problem in the glass manufacturing industry. *European Journal of Operational Research*, 287(1):378–388, 2020. doi: 10.1016/j.ejor.2020.05.016.

J. Puchinger and G.R. Raidl. Models and algorithms for three-stage two-dimensional bin packing. *European Journal of Operational Research*, 183(3):1304–1327, 2007. doi: 10.1016/j.ejor.2005.11.064.

M.P. Reinders and T.H.B Hendriks. Lumber production optimization. *European Journal of Operational Research*, 42(3):243–253, 1989. doi: 10.1016/0377-2217(89)90436-0.

M. Russo, A. Sforza, and C. Sterle. An exact dynamic programming algorithm for large-scale unconstrained two-dimensional guillotine cutting problems. *Computers & Operations Research*, 50:97–114, 2014. doi: 10.1016/j.cor.2014.04.001.

M. Russo, M. Boccia, A. Sforza, and C. Sterle. Constrained two-dimensional guillotine cutting problem: upper-bound review and categorization. *International Transactions in Operational Research*, 27(2):794–834, 2020. doi: 10.1111/itor.12687.

G. Scheithauer and J. Terno. Guillotine cutting of defective boards. *Optimization*, 19(1):111–121, 1988. doi: 10.1080/02331938808843323.

M. Sierra-Paradinas, O. Soto-Sánchez, A. Alonso-Ayuso, F.J. Martín-Campo, and M. Gallego. An exact model for a slitting problem in the steel industry. *European Journal of Operational Research*, 295(1):336–347, 2021. doi: 10.1016/j.ejor.2021.02.048.

A. Silva, L.C. Coelho, M. Darvish, and J. Renaud. A cutting plane method and a parallel algorithm for packing rectangles in a circular container. *European Journal of Operational Research*, 303(1):114–128, 2022. doi: 10.1016/j.ejor.2022.02.023.

E. Silva, J.F. Oliveira, T. Silveira, L. Mundim, and M.A. Carravilla. The floating-cuts model: a general and flexible mixed-integer programming model for non-guillotine and guillotine rectangular cutting problems. *Omega*, 114:102738, 2023. doi: 10.1016/j.omega.2022.102738.

A.S. Velasco and E. Uchoa. Improved state space relaxation for constrained two-dimensional guillotine cutting problems. *European Journal of Operational Research*, 272(1):106–120, 2019. doi: 10.1016/j.ejor.2018.06.016.

Sunkanghong Wang, Roberto Baldacci, Qiang Liu, and Lijun Wei. EATKG: An open-source efficient exact algorithm for the two-dimensional knapsack problem with guillotine constraints. *European Journal of Operational Research*, 2025. doi: 10.1016/j.ejor.2025.05.033.

L. Wei and A. Lim. A bidirectional building approach for the 2D constrained guillotine knapsack packing problem. *European Journal of Operational Research*, 242(1):63–71, 2015. doi: 10.1016/j.ejor.2014.10.004.

G. Wäscher, H. Haußner, and H. Schumann. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3):1109–1130, 2007. doi: 10.1016/j.ejor.2005.12.047.

S. Yao, H. Zhang, Q. Liu, J. Leng, and L. Wei. Combinatorial Benders' decomposition for the constrained two-dimensional non-guillotine cutting problem with defects. *International Journal of Production Research*, 62(23):8299–8325, 2024. doi: 10.1080/00207543.2024.2338194.

S. Yao, H. Zhang, L. Wei, and Q. Liu. An exact approach for the two-dimensional strip packing problem with defects. *Computers & Industrial Engineering*, 200:110866, 2025. doi: 10.1016/j.cie.2025.110866.

M.T. Yu, T.Y. Lin, and C. Hung. Active-set sequential quadratic programming method with compact neighbourhood algorithm for the multi-polygon mass production cutting-stock problem with rotatable polygons. *International Journal of Production Economics*, 121(1):148–161, 2009. doi: 10.1016/j.ijpe.2009.01.014.

H. Zhang, S. Yao, Q. Liu, J. Leng, and L. Wei. Exact approaches for the unconstrained two-dimensional cutting problem with defects. *Computers & Operations Research*, 160:106407, 2023. doi: 10.1016/j.cor.2023.106407.