The determination of the guillotine restrictions for a rectangular cutting-stock pattern

DANIELA MARINESCU
Transilvania University of Brașov
Department of Computer Science
Iuliu Maniu 50, 500091, Brașov
ROMANIA
mdaniela@unitbv.ro

ALEXANDRA BĂICOIANU
Transilvania University of Brașov
Department of Computer Science
Iuliu Maniu 50, 500091, Brașov
ROMANIA
a.baicoianu@unitbv.ro

Abstract: We consider a two-dimensional rectangular Cutting-Stock problem in case of a cutting pattern with gaps. First we present two new graph representations of the cutting pattern, weighted graph of downward adjacency and weighted graph of rightward adjacency. Using this kind of representation we propose a method to verify guillotine restrictions of the pattern which can be applied for cutting-stock pattern with gaps but also for the covering pattern without gaps and overlapping.

Key–Words: two-dimensional cutting-stock problems, cutting pattern representation, guillotine restrictions.

1 Introduction

Problems of cutting and covering (packing) of concrete or abstract objects appear under various specifications [3]: cutting-stock problems, knapsack problems, container and vehicle loading problems, pallet loading, bin-packing, assembly line balancing, etc. The problem is NP-hard and arises in various production processes with applications varying from the home-textile to the glass, steel, wood, and paper industries, where rectangular figures are cut from large rectangular sheets of materials. Furthermore, an arguably more complicated problem, called Cutting and Covering [5], can be derived by cutting a piece of material into small pieces which are then used to cover a surface without overlapping and leaving any gaps.

Dyckhoff provided in [3] a classification of the various types of cutting problems such as: one dimensional, two dimensional and three dimensional with different types of constraints. A frequent constrain, imposed by industrial applications of the two or three dimensional problem, is the so-called guillotine restriction which states that the resulting patterns need to be guillotine cuttable. Many applications of two-dimensional cutting and covering in the glass, wood, and paper industries, for example, are restricted to guillotine cutting.

In literature there have been proposed several techniques that solve the Cutting and Covering problems such as formulating the problem as a mixed integer problem, heuristics [13], genetic algorithms [7] as well as approximation algorithms [6]. All these methods result in a pattern or a set of patterns. They are not adequate for constructing cutting patterns when the approach is used to solve the guillotine cutting-stock problem.

In [1] a polynomial algorithm is presented for two special cases of guillotine cutting a rectangle into small rectangles. However, the guillotine restrictions are difficult to respect in the general pattern-generation process. So instead of generation of a cutting-stock pattern with guillotine restrictions it is possible to use an analytic method to verify if the pattern, obtained by some methods used in case of non-guillotine cutting, is with or without guillotine restrictions [11]. Nevertheless, the method from [11] is rather unpractical since the cutting pattern is represented as an array pattern, which implies a large matrix representation.

Another analytical method, presented in [12], used the graph representation [8, 9] of a covering pattern without gaps or overlapping. This method developed an algorithm for guillotine restrictions verification, based on connections between guillotine cut and the connex components of the graphs. But this algorithm is useless in case of a cutting-stock pattern with gaps. For this kind of pattern we propose another method based on two new kinds of graph representations, weighed graph representations, which is more general comparing with the method presented in [12].

2 Problem formulation

Let \( P \), a rectangular plate, characterized by length \( L \) and width \( W \). From plate \( P \) we cut \( k \) rectangular items,
$C_i, i = 1, 2, \ldots, k$. An item $C_i$ is characterized by length $l_i$ and width $w_i$.

**Definition 1** A rectangular cutting-stock pattern is an arrangement of the $k$ rectangular items $C_i$ on the supporting plate $P$, so that the borders of the items $C_i$ to be parallel with the borders of the plate $P$.

For this kind of patterns we have presented in [8, 9] two graph representations. Starting from these representations we complete the graphs by adding a value for each arc from the two graphs.

**Definition 2** A rectangular cutting pattern has guillotine restrictions if at every moment of the cutting process the remaining supporting rectangle is separated in two new rectangles by a cut from an edge to the opposite edge of the rectangle and the cutting line is parallel with the two remaining edges.

In the set of the rectangles $\{C_1, C_2, \ldots, C_k\}$ from the covering pattern we define a downwards adjacency relation and a rightwards adjacency relation.

**Definition 3** The rectangle $C_i$ is downward adjacent (rightward adjacent) with rectangle $C_j$ if in the cutting pattern, $C_j$ is to be found downward (respectively rightward) $C_i$ and their borders have at least two common points.

Let $C = \{C_1, C_2, \ldots, C_k\}$ and $R_d, R_r \notin C$. For any covering pattern, we defined in [8, 9], a graph of downwards adjacency and another one of rightwards adjacency. We define now two new graphs a weighted graph of downwards adjacency, $G_d$, and another one of rightwards adjacency, $G_r$. We will use in these definitions the notation $V(X, Y)$ for the value of the arc $(X, Y)$

**Definition 4** The weighted graph of downward adjacency $G_d = (C \cup \{R_d\}, \Gamma_d)$ has as vertices the rectangles $C_1, C_2, \ldots, C_k$ and a new vertex $R_d$ symbolizing the northern borderline of the supporting plate $P$. The $\Gamma_d$ is defined as follows:

\[
\begin{align*}
\Gamma_d(C_i) &\ni C_j \text{ if } C_i \text{ is downward adjacent with } C_j \\
\Gamma_d(R_d) &\ni C_i \text{ if } C_i \text{ touches the North border of the support plate } P \\
V(X, C_j) &= w_j, \forall X \in C \cup R_d \text{ and } C_j \in C
\end{align*}
\]

**Definition 5** The weighted graph of rightward adjacency $G_r = (C \cup \{R_r\}, \Gamma_r)$, where $R_r$ symbolizes the western border. The $\Gamma_r$ is defined as follows:

\[
\begin{align*}
\Gamma_r(C_i) &\ni C_j \text{ if } C_i \text{ is rightward adjacent with } C_j \\
\Gamma_r(R_r) &\ni C_i \text{ if } C_i \text{ touches the West border of the support plate } P \\
V(X, C_j) &= l_j, \forall X \in C \cup R_r \text{ and } C_j \in C
\end{align*}
\]

Let the cutting-stock pattern from Fig. 1. The weighted graphs $G_d$ and $G_r$, are represented in Fig. 2 and Fig. 3.

We remark that in the graphs $G_d$ and $G_r$ the vertex $R_d$ (respectively $R_r$) is connected by an arc to the vertex $C_i$ if and only if $C_i$ touches the northern (respectively the western) border of the support $P$.

**Remark 6** In the following we consider only the rectangular cutting-stock pattern where the rectangles are not situated under or to the right of an empty spaces. When it is not true (see Fig. 4), we can define another pattern (equivalent) by moving the rectangles (in Fig. 4 this rectangle is $C$) down or to the left till they touch the border of another rectangle. In the sense of the cutting-stock problem with a minimum rest, for ev-
ery cutting-stock pattern there is always an equivalent pattern of this form.

From the Remark 6 it results that the weighted graphs $G_d$ and $G_r$, attached to a cutting-stock pattern have the following properties [9]: the graphs are quasi strongly connected and have no circuit.

Let us have a cutting-stock pattern with guillotine restrictions. From [10] it follows that it is possible to represent a rectangular cutting-stock pattern with guillotine restrictions using an expression with two operations:

1. $\Theta$ - the s-line concatenation, an operation for horizontal cuts;
2. $\phi$ - the s-column concatenation, an operation for vertical cuts.

For example, the cutting pattern from Fig. 1 will be described by the following expression:

$$\Theta \Theta AE \Theta \Theta BDC.$$ 

### 3 Cuts determination

In [12] we presented an algorithm for cuts determination in case of a cutting pattern without gaps. But it is not possible to apply this algorithm in the case of a cutting-stock pattern with gaps.

Starting from a rectangular cutting-stock pattern with gaps we intend to find a connection between guillotine restrictions and the two weighted graphs of adjacency, $G_d$ and $G_r$.

For this purpose we will use the notation $Lpd(R_d, C_i)$, respectively $Lpr(R_r, C_i)$ for the length of the path from $R_d$ to $C_i$ in the graphs $G_d$, respectively $G_r$. We remark that $Lpd(R_d, C_i)$ represents the distance from the northern border of the plate $P$ to the southern border of piece $C_i$ and similarly $Lpr(R_r, C_i)$ represents the distance from the western border of the plate $P$ to the eastern border of piece $C_i$.

**Remark 7** If a cutting-stock pattern has an horizontal guillotine cut situated to a distance $M$ from the North border of the supporting plate $P$ then the set of the items, $C_i$, can be separated in two subsets $S_1$, the set of the items situated above this cut, and $S_2$ the set of the items situated below this cut. Of course in the weighted graph $G_d$ we have:

1. $Lpd(R_d, C_i) \leq M$ for every $C_i \in S_1$;
2. $Lpd(R_d, C_i) > M$ for every $C_i \in S_2$.

We obtain a similar result if the cutting-stock pattern has a vertical cut.

The two conditions from the above remarks are necessary but not sufficient because it is possible to cut to intersect some items from the set $S_2$. We present in the following necessary and sufficient conditions for a guillotine cut.

**Theorem 8** Let a rectangular cutting-stock pattern with possible gaps and the weighted graph $G_d$ attached to the pattern. The cutting-stock pattern has an horizontal guillotine cut on the distance $M$ from the northern border of the supporting plate if and only if it is possible to separate the sets of the items, $C_i$ in two subsets, $S_1$ and $S_2$ so that:

1. $C = S_1 \cup S_2, S_1 \cap S_2 = \emptyset$;
2. For every $C_j \in C$ so that $(R_d, C_j) \in \Gamma_d$ it follows that $C_j \in S_1$;
3. $Lpd(R_d, C_i) \leq M$ for every $C_i \in S_1$;
4. If there is $C_j \in S_1$ so that $Lpd(R_d, C_j) < M$ then all direct descender of $C_j$ will be in $S_1$.

**Proof:**

i. Suppose that the cutting-stock pattern has a vertical guillotine cut. That means the sets of items $C$ can be separated in two subsets, $S_1$, the set of the vertices situated above the cut, and $S_2$, the set of the vertices situated below the cut. From the Remark 7 it follows that the conditions 1, 2 and 3 are fulfilled.

Suppose that the condition 4 is not fulfilled. That means there are two items $C_j \in S_1$ and $C_i \in S_2$ so that $Lpd(R_d, C_j) < M$ and the item $C_i$ is a successor of $C_j$. Because $C_i \in S_2$ it follows that $Lpd(R_d, C_i) > M$ and an horizontal cut situated on the distance $M$ from the northern border of the supporting plate will intersect the item $C_i$. It means that without the condition 4 it is impossible to separate the set of the items by an horizontal cut. So our supposition that the condition 4 is not fulfilled is false.

ii. Suppose all the conditions 1-4 are fulfilled but it is not possible to make an horizontal cut on the distance $M$ in the cutting-stock pattern. It follows that there is at least item $C_i \in S_2$ which is intersected by such a cut. It means that the distance from the
northern border of the supporting plate to the northern border of the item $C_i$ is less than $M$ and the distance from the northern border of the supporting plate to the southern border of the item $C_i$ is greater than $M$.

But from the Remark 7 it follows that the northern border of the item $C_i$ is identical with the southern border of some item $C_j$, situated above $C_i$. That means $(C_j, C_i) \in \Gamma_d$ and $L_{pd}(R_d, C_j) < M$ and so $C_j \in S_1$. From condition 4, because $C_i$ is a successor of $C_j$, it follows that $C_i$ must be in $S_1$ in contradiction with our hypothesis. That means that if the conditions 1-4 are fulfilled then there is an horizontal guillotine cut in the cutting-stock pattern. \hfill \Box

We obtain a similar result if we consider the weighted graph of rightwards adjacency.

4 The algorithm for verification of the guillotine restrictions

The results from the previous theorem suggest an algorithm for verification of the guillotine restrictions, in case of a cutting-stock pattern with gaps.

**Input data:** The weighted graphs $G_d$ or $G_r$ attached to a rectangular cutting pattern.

**Output data:** The s-pictoral representation of the cutting pattern like a formula in a Polish prefixed form.

**Method:** The algorithm constructs the syntax tree for the s-pictoral representation of the cutting pattern, starting from the root to the leaves (procedure PRORD). For every vertex of the tree it verifies if it is possible to make a vertical (procedure VCUT) or horizontal cut (HCU'T), using an algorithm for decomposition of a graph in two components, $S_1$ and $S_2$.

We will use the following notations:
- $G'_r, G'_d$ are the subgraphs of $G_r|_X$, respectively $G_d|_X$, where we can add, if it is necessary, the root $R_r (R_d)$ and the arcs starting from $R_r (R_d)$, like in Definition 4.
- $L_1 (L_2)$ is the weight of the first (second) cutting support which contains all the items from $S_1 (S_2)$.
- $\text{succ}(C_i|G)$ is the set of successors of the item $C_i$ in the graph $G$.

The method ADD() is used for addition of the next member in the Polish prefixed form.

We remark that we can apply this algorithm also in case of a cutting-stock pattern without gaps and, of course, in the case of covering pattern with or without gaps.

4.1 Example

Let us have the cutting-stock pattern from Fig. 1 with the weighted graphs, $G_d$ and $G_r$.

```plaintext
PROCEDURE PRORD(G, C, L, W, ADD())
begin
  VCUT(G_r, C, L, W, err, S_1, S_2, L_1, L_2);
  if err = 0 then
    if |C| = 1 then ADD(C)
    else ADD(\emptyset);
    PRORD(G_d, S_1, L_1, W, ADD());
    PRORD(G_d, S_2, L_2, W, ADD());
  end
  else No guillotine restrictions
end
PROCEDURE VCUT(G_r, X, L, W, err, S_1, S_2, L_1, L_2)
begin
  err = 0;
  CONSTRUCT-SUBGRAPH(G_r, G'_r, X, R_r);
  V := \bigcup \{ C_i | C_i \in X, (R_r, C_i) \in \Gamma_r \}, where all the elements are unmarked
  maxM := \max \{ l_i | C_i \in V \}
  P_1 := \{ l_i | C_i \in V \}
  while \exists C_i \in V unmarked element do
    mark C_i;
    if P_i < maxM then
      for C_j \in succ(C_i|G'_r) do
        V := V \bigcup \{ C_j \} if C_j is an unmarked element
        P_j := P_i + l_j
        if P_j > maxM then
          MaxM := P_j
      end
    end
  end
  maxM := \max \{ L_{pd}(R_r, C_i) | C_i \in V \}
  if maxM = L then
    err = 1;
  end
  L_1 := maxM;
  L_2 := L - maxM;
  S_1 := V;
  S_2 := X - V;
end
```
For this example \( L = 8 \), \( W = 5, 5 \), and first we are trying a vertical cut, with \( \max M = 2, 5 \). We obtain two sets, one composed from nodes \( \{A, E\} \) and \( \{B, D, C\} \), see Fig. 5, Fig. 6.

In the syntactic tree from Fig. 7 we have 2 components connected using the operation column concatenation \( \varnothing \) for the vertical cut. The prefix Polish notation for this syntactic tree from Fig. 7 is: \( \varnothing \).

We continue to make horizontal or vertical cut for the left and right components from the syntactic tree until every component contains only one item from the covering pattern.

Resuming, we will have one horizontal cut on the first component in items \( A \) and \( E \). On the second component obtained at the previous step, we will have a vertical cut, which means the decomposition in one new component, 3, and the item \( C \). In the final step it will be the horizontal cut from Fig. 8, Fig. 9. In here, we get the items \( B \) and \( D \), which are connected with...
line concatenation. The syntactic tree for the whole example is presented in Fig. 10.

The Polish notation for the tree from Fig. 10 is:
\[ \text{⊘⊖ } AE \text{⊘⊖ } BDC. \]

### 4.2 Correctness and Complexity

The correctness of the algorithm follows from the theorems 1, that make the connection between a guillotine cut and the decomposition of a graph in two sub-graphs.

The procedure `PREORD()` represents a preorder traversal of a graph, so the complexity is \( O(k) \) [2], where \( k \) is the number of the cutting items. Also, in the procedure `VCUT`, respectively `HCUT` we traverse a subgraph of the initial graph. So, the complexity of the algorithm is \( O(k^2) \).

### 5 Conclusions

No matter if it is a guillotine covering [12] or cutting-stock, the problem is a constraint on a complete partition of two-dimensional space. Guillotine partitions were introduced in 1980ies and have numerous applications [4] in computational geometry, computer graphics, pattern recognition etc.

Various aspects of the problem are found in industries that produce two dimensional sheets of glass, textiles, paper or other material. A similar problem arises in the design of layouts for integrated circuits or in the design of an optimal placement of a set of solar panels. Like the complete partition, the guillotine problem remains NP hard. For this reason it is better to use an algorithm for generating an unconstrained covering or cutting-stock pattern and, after that or in each step, to use our algorithms for verifying the guillotine restrictions of the generated pattern.

**Acknowledgements:** The research was supported by the Grant IDEI no. 134/2007.

**References:**


