Guillotine Cutting of Defective Boards

G. Scheithauer and J. Terno
Technische Universität Dresden, Sektion Mathematik

Summary: The paper deals with the problem to cut rectangular pieces from a defective board using Guillotine cuts. To solve this problem dynamic programming is used and it is shown how the solution process can be improved. Practical examples show the quality of the investigations. The CPU-time needed corresponds to the theoretical complexity.

AMS 1980 Subject Classification:
Primary: 90 C 39, Secondary: 90 C 11
Key words: Dynamic Programming, Integer Programming, cutting stock problems, Guillotine cuts.

1. Introduction

The optimal exploitation of the existing material is an important problem in economics. In this paper we consider the problem to cut out rectangular pieces from a defective board using Guillotine cuts.

To solve the well known cutting stock problem i.e. to get rectangular pieces from rectangular defectless sheets GILMORE and GOMORY (see e.g. [1, 2]) developed a fundamental solution strategy based on the revised simplex method with column generation technique. To generate a column respectively cutting pattern there is to handle a special KNAPSACK problem. These KNAPSACK problems can be solved in general by using dynamic programming or branch and bound. Finally, an integer solution of the cutting stock problem can be obtained from the continuous optimal solution by means of a suitable rounding procedure.

In each simplex step a new (optimal) cutting pattern is to compute by solving special KNAPSACK problems. Therefore a greatest care is necessary by choosing a suitable solution algorithm. The investigations in [4, 5, 6] show that the complete process of solving cutting stock problems can be handled in an effective way and that the corresponding software is available.

In the present paper we consider an analogous problem. Given a not necessarily rectangular board (sheet) which contains some defective areas (defects), and we look for an optimal Guillotine cutting pattern to get desired rectangular pieces. To solve this problem we use dynamic programming. Our aim is to show how the solution process can be improved.
2. Problem Formulation

In sawmills the following cutting problem is of great interest. Rectangular pieces are to be cut out from bent-lined bordered boards which additionally may contain several defects (e.g. knot-holes, chinks). Since only Guillotine cuts are admissible we assume that the upper and lower border are represented by two step functions and that the defects are rectangular (see Fig. 1).

![Diagram of sawmill cutting](image)

Considering the technological conditions we get the following cutting problem: Rectangular pieces \( T_i \) with length \( l_i \), width \( w_i \) and valuation \( d_i \), \( i = 1(1) n \), are to be cut out from a board (sheet), which is bordered by two step functions \( \gamma \) and \( \omega \) and which additionally contains several rectangular defects \( D_q \), \( q = 1(1) p \). The sum of valuations of the produced pieces is to be maximized. (Without loss of generality we assume in the following that all used values are integers.) The pieces have to be cut in the following way. The board is cut by vertical Guillotine cuts in so-called sections (first stage). The sections are cut in strips by horizontal Guillotine cuts (second stage). Finally, using vertical Guillotine cuts once more the defects are cut out of the strips and desired pieces are produced from the remaining defectless areas (third stage, see Fig. 1).

According to the classification of the cutting stock problems made by GILMORE/GOMORY [1], this cutting technique represents a 3-stage (Guillotine) cutting problem in the so-called unexact case.

Let \( L \) be the length and \( W \) the width of a rectangle which surrounds the given board and whose lower and left edges are located on the \( x \) - and \( y \)-axis of a cartesian system. The rectangular defects \( D_q \) of the board represented in the form \( R(xl_q, yl_q, xu_q, yu_q) \), \( q = 1(1) p \), where \((xl_q, yl_q)\) respectively \((xu_q, yu_q)\) denote the coordinates of the lower left resp. upper right corner of the rectangles. To uniform the following description we assume that \( D_1 = R(0, 0, 0, W) \) and \( D_p = -R(L, 0, L, W) \).
Let the upper and the lower border of the board be given by step functions \( \psi(x) \) (upper border) and \( \omega(x) \) (lower border). We assume that these step functions are continuously from the left with step points \( \mu_q, q = 0(1) p_w \), where \( \mu_0 = 0 \) and \( \mu_{p_w} = L \), respectively \( \nu_q, q = 0(1) p_w \), where \( \nu_0 = 0 \) and \( \nu_{p_w} = L \).

By reason of the concrete cutting situation let \( \psi \) and \( \omega \) at least fulfil the conditions

(i) \( \omega(x) \equiv \psi(x) \), \( 0 \leq x \leq L \),

(ii) \( \psi(\mu_q) = \psi(\mu_{q+1}) \land \psi(\mu_{q+2}) = \psi(\mu_{q+3}) \)
    \( \implies \mu_{q+1} - \mu_q \geq \min_{i=1(1)n} l_i \) \( (q = 0(1) p_w - 2) \),

(iii) \( \omega(\nu_q) = \omega(\nu_{q+1}) \land \omega(\nu_{q+2}) = \omega(\nu_{q+3}) \)
    \( \implies \nu_{q+1} - \nu_q \geq \min_{i=1(1)n} l_i \) \( (q = 0(1) p_w - 2) \).

Let \( R = \{ r_j \mid j = 1(1)\alpha \} \) denote a given set of feasible lengths of the sections. We assume \( r_1 < r_2 < \ldots < r_\alpha \). Mostly,

\( R = \{ l_i \mid i = 1(1) n \} \quad \text{or} \quad R = \left\{ r \mid r \equiv \max_{i=1(1)n} l_i, r = \sum_{i=1}^n l_i y_i, y_i \geq 0, \text{integer, } i = 1(1) n \right\} \)

are used.

Based on the investigations of Gilmore/Gomory ([1], [2]) Hahn [3] was the first one to consider the problem of cutting rectangular pieces out of rectangular defective sheets. In [3] an algorithm to compute an optimal pattern was proposed for a somewhat simpler cutting technique. Investigations to improve the solution process were made only in the direction of an approximate algorithm.

### 3. Basic Recursions

The computation of an optimal cutting pattern is based on dynamic programming. In the following briefly we list the basic recursions.

Let \( H(x, r, y, w) \) denote the maximum valuation obtainable by cutting out pieces from a strip \( R (x-r, y-w, x, y) \). In accordance with the given cutting
technique the useful area in the considered strip (not shaded, see Fig. 2) consists of only rectangles with width \( w \) which contain no defect.

![Fig. 2](image)

Let \( G(x, r, y) \) be the maximum valuation obtainable by cutting of the rectangle \( R (x-r, 0, x, y) \). Let \( w_k, k = 1(1)m, m = n \), denote the different widths of the
given pieces and let be \( w_{s_1} < w_{s_2} < \ldots < w_{s_m} \). Then it holds:

\[
G(x, r, 0) = 0, \\
G(x, r, y) = \max_{i=1(1)m} g_i(x, r, y) \tag{1}
\]

with

\[
g_i(x, r, y) = \begin{cases} 
0, & \text{if } w_{s_i} > y \\
G(x, r, y - w_{s_i}) + H(x, r, y, w_{s_i}) & \text{otherwise, } y = 1(1) W.
\end{cases}
\]

Hence, an optimal cutting pattern of the section \( R(x-r, 0, x, W) \) has the valuation \( G(x, r, W) \).

Furthermore, let \( V(x) \) be the maximum valuation of the rectangle \( R(0, 0, x, W) \), \( x=0(1)L \). Then we have with respect to the allowed cuts of the first stage:

\[
V(0) = 0, \\
V(x) = \max_{j=1(1)m} v_j(x) \tag{2}
\]

with

\[
v_j(x) = \begin{cases} 
0, & \text{if } r_j > x, \\
V(x - r_j) + G(x, r_j, W) & \text{otherwise, } x=1(1) L.
\end{cases}
\]

\( V(L) \) is the value of an optimal solution of the cutting problem.

The computation of \( V(L) \) requires \( O(xL) \) optimal values \( G(x, r, W) \) of all possible sections. To calculate one \( G \)-value \( O(mW) \) computations of values \( H(x, r, y, w) \) are necessary. It turns out that the application of values \( F(r, y) \) is very useful where \( F(r, y) \) denotes the maximum valuation of a defectless rectangle \( R(0, 0, r, y) \).

If the section \( R(x-r, 0, x, W) \) includes no defects then \( G(x, r, W) = F(r, W) \). Using \( F(r, w) \) the calculation of the value \( H(x, r, y, w) \) requires \( O(p) \) computational steps. Therefore the whole computational expense is proportional to

\[
xL + xL (mW + mWP) = xL + xLmW + xLmWP \tag{3}
\]

where the first term corresponds to the \( x \)-recursion, the second to the \( xL \)-recursions and the third to the \( xLmW \)-computations.

4. Improvements of the Basic Recursions

The computation of \( V(L) \) resp. \( G(x, r, W) \) corresponds to the solution of one-dimensional KNAPSACK problems with \( x \) resp. \( m \) variables. Additionally the coefficients of the object functions \( G(x, ...) \) and \( H(..., y, ...) \) depend on \( x \) resp. \( y \). Since the calculation of these coefficients determines essentially the total expense, we would like to reduce the number of \( G \)- and \( H \)-calculations. We only compute \( G(x, r, W) \) resp. \( H(x, r, y, w) \) if a changed value is to expect in comparison with \( G(x-1, r, W) \) resp. \( H(x, r, y-1, w) \). Let

\[
XU = \{x\mu_q \mid q = 1(1) p\}
\]

and

\[
XL = \{x\lambda_q \mid q = 1(1) p\}.
\]
4.1. Cutting of a Rectangular Board

A changed optimal value \( G(x, r_j, W) \) in comparison with \( G(x-1, r_j, W) \) can arise during the cut of the section \( R(x-r_j, 0, x, W) \) in the following two cases:

![Diagram](image)

**Case 1:** If \( x-r_j \in XU \) or \( x-r_j \in XU, j_0 \neq j \), then possibly further pieces with length \( r_j \) resp. \( r_j \) could be cut out of the section \( R(x-r_j, 0, x, W) \) in comparison with the section \( R(x-r_j-1, 0, x-1, W) \) (see Fig. 3).

![Diagram](image)

**Case 2:** If \( x-1 \in XL \) or \( x-r_j + r_j - 1 \in XL, j_0 \neq j \), then possibly fewer pieces with length \( r_j \) resp. \( r_j \) could be cut out of the section \( R(x-r_j, 0, x, W) \) in comparison with the section \( R(x-r_j-1, 0, x-1, W) \) (see Fig. 4).

To summarize, if there exists an index \( j_0 \) with \( j_0 \neq j \) and \( x-r_j \in XU \) or \( x-r_j + r_j - 1 \in XL \) then \( G(x, r_j, W) \) has to be computed anew. This fact can be stated in the following

**Lemma 1:** If for all indices \( j_0 \) with \( j_0 \neq j \) the conditions

\[
x - r_{j_0} \notin XU \quad \text{and} \quad x - r_j + r_j - 1 \notin XL
\]

are fulfilled then it holds

\[
G(x, r_j, W) = G(x-1, r_j, W).
\]
To take advantage of Lemma 1 we define some quantities:

\[
\delta(x) = \begin{cases} 
  z + 1, & \text{if } x - r_j \notin XU \text{ for } j = 1(1)\, x, \\
  j, & \text{if } x - r_j \in XU \text{ and } x - r_{j_0} \notin XU \text{ for } j_0 < j,
\end{cases}
\]

and

\[
\tau(x) = \begin{cases} 
  z + 1, & \text{if } x + r_j - 1 \notin XL \text{ for } j = 1(1)\, x, \\
  j, & \text{if } x + r_j - 1 \in XL \text{ and } x + r_{j_0} - 1 \notin XL \text{ for } j_0 < j,
\end{cases}
\]

\[
x = 0(1)\, L.
\]

By this we obtain two conditions which are simple to inspect.

a) If \( j < \delta(x) \) then \( x - r_j \notin XU \) for \( j_0 \equiv j \).

b) If \( j < \tau(x - r_j) \) then \( x - r_j + r_{j_0} - 1 \notin XL \) for \( j_0 \equiv j \).

**Remarks:**

a) If \( xu_1 \leq xu_2 \leq \ldots \leq xu_p \) is valid then the computation of \( \delta(x), x = 0(1)\, L \), according to

\[
\delta(x) := a + 1, \quad x = 0(1)\, L,
\]

\[
\delta(xu_q + r_j) := j, \quad j = 1(1)\, x, \quad q = 1(1)\, p,
\]

requires \( O(L + a\, p) \) computational steps.

b) If \( xl_1 \leq xl_2 \leq \ldots \leq xl_p \) then \( \tau(x), x = 0(1)\, L \), can be computed in the following manner

\[
\tau(x) := a + 1, \quad x = 0(1)\, L,
\]

\[
\tau(xl_q + 1 - r_j) := j, \quad j = 1(1)\, x, \quad q = p(-1)\, 1,
\]

with at most \( O(L + a\, p) \) computational steps.

In the case if \( G(x, r, W) \) has to be computed anew then several quantities \( H(x, r, y, w) \) are needed. Let

\[
YU(x, r) = \{yu_q \mid xl_q < x \wedge xu_q \geq x - r, \, q = 1(1)\, p\} \cup \{W\}
\]

and

\[
YL(x, r) = \{yl_q \mid xl_q < x \wedge xu_q \geq x - r, \, q = 1(1)\, p\} \cup \{0\}.
\]

During the computation of the \( H(x, r, y, w) \)-values we take into consideration the following two cases. If \( y - w \in YU(x, r) \) is valid then the strip \( R(x - r, y - w, x, y) \) contains at least one defect fewor which belongs to \( R(x - r, y - w - 1, x, y - 1) \). If \( y - 1 \notin YL(x, r) \) then in comparison with \( R(x - r, y - w - 1, x, y - 1) \) a new defect is to be considered (see Fig. 5).
Lemma 2: If \( y \in YU(x, r) \) and \( y - 1 \in YL(x, r) \) are valid then it holds:
\[
H(x, r, y, w) = H(x, r, y - 1, w).
\]

Remark: To apply Lemma 2 in an effective way the use of the following quantities is suitable:
\[
\vartheta(y) = \begin{cases} 1, & \text{if } y - 1 \in YL(x, r), \\ 0, & \text{otherwise}, 
\end{cases}
\]
and
\[
\eta(y) = \begin{cases} 1, & \text{if } y \in YU(x, r), \\ 0, & \text{otherwise}, 
\end{cases}
\]
\[
y = \Theta(1) W.
\]

4.2. Cutting of Bentlined Bordered Boards

During the computation of an optimal cutting pattern the upper and the lower border of the board can be handled as special defects whose "neighbouring" position can be used to reduce the number of required \( G \)-calculations.

Lemma 3: If for all indices \( j_0 \) with \( j_0 \leq j \) the conditions

1) \( x - r_j \in XU \) and \( x - r_j + r_{j_0} - 1 \in XL \),
2) \( \psi(x - r_j) = \psi(x - r_j + 1) \) and \( \psi(x - r_j + r_{j_0} - 1) = \psi(x - r_j + r_{j_0}) \),
3) \( \omega(x - r_j) = \omega(x - r_j + 1) \) and \( \omega(x - r_j + r_{j_0} - 1) = \omega(x - r_j + r_{j_0}) \)

are fulfilled then it holds
\[
G(x, r_j, W) = G(x - 1, r_j, W).
\]

We use again quantities \( \delta(x) \) and \( \tau(x) \) obtainable as follows:

**Computation of \( \delta(x) \)**

1) \( \delta(x) := x + 1, \quad x = 0(1) L \),
2) \( \delta(xu_q + r_j) := \min \{ \delta(xu_q + r_j), j \}, \quad j = 1(1) z, \quad q = 1(1) p \),
3) \( \delta(u_q + r_j) := \min \{ \delta(u_q + r_j), j \}, \quad \text{if } \psi(u_q) = \psi(u_{q+1}) \),
\[
\quad j = 1(1) z, \quad q = 1(1) p_q - 1,
\]
4) \( \delta(v_q + r_j) := \min \{ \delta(v_q + r_j), j \}, \quad \text{if } \omega(v_q) = \omega(v_{q+1}) \),
\[
\quad j = 1(1) z, \quad q = 1(1) p_q - 1.
\]

**Computation of \( \tau(x) \)**

1) \( \tau(x) := x + 1, \quad x = 0(1) L \),
2) \( \tau(xl_q - r_j + 1) := \min \{ \tau(xl_q - r_j + 1), j \}, \quad j = 1(1) z, \quad q = 1(1) p \),
3) \( \tau(u_q - r_j + 1) := \min \{ \tau(u_q - r_j + 1), j \}, \quad \text{if } \psi(u_q) = \psi(u_{q+1}) \),
\[
\quad j = 1(1) z, \quad q = 1(1) p_q - 1,
\]
4) \( \tau(v_q - r_j + 1) := \min \{ \tau(v_q - r_j + 1), j \}, \quad \text{if } \omega(v_q) = \omega(v_{q+1}) \),
\[
\quad j = 1(1) z, \quad q = 1(1) p_q - 1.
\]
With the help of these quantities Lemma 3 can be formulated as follows:

**Lemma 3:** (second version): If \( j < \delta(x) \) and \( j < \gamma(x - r_j) \) are fulfilled then it holds:

\[
G(x, r_j, W) = G(x - 1, r_j, W).
\]

To compute \( H(x, r, y, w) \) only when it is necessary we define the sets

\[
YU(x, r) = \{yu_q \mid x_u < x \wedge xu_q > x - r, q = 1(1)p\} \cup \{(t) \mid x - r < t < x\},
\]
\[
YL(x, r) = \{yl_q \mid x_l < x \wedge xu_q > x - r, q = 1(1)p\} \cup \{(t) \mid x - r < t < x\}
\]

and the quantities

\[
\phi(y) = \begin{cases} 1, & \text{if } y - 1 \in YL(x, r), \\ 0, & \text{otherwise} \end{cases}
\]
\[
\eta(y) = \begin{cases} 1, & \text{if } y \in YU(x, r), \\ 0, & \text{otherwise} \end{cases}
\]

Then we have

**Lemma 4:** If \( \phi(y) = 0 \) and \( \eta(y - w) = 0 \) then it holds:

\[
H(x, r, y, w) = H(x, r, y - 1, w).
\]

5. Remarks on Computational Experience

In order to compute optimal cutting pattern of a board including several defects a procedure based on the recursions of Chapter 3 and the improvements of Chapter 4 was implemented on a BESM6 computer of the Technical University of Dresden.

In general the set of feasible lengths of the sections was choosen as \( R = \{l_i \mid i = 1(1)n\} \).

To get not only "small" pieces analogous to [3] we use a progressive valuation of the pieces, in general \( d_i = \beta l_i w_i + \gamma (l_i w_i)^2 \) with \( \beta = 0.1 \) and \( \gamma = 0.0001 \) and a corresponding rounding procedure.

First we refer some results of examples of \( n = 6 \) pieces. The data of length, width and valuation are shown in Table 1.

<table>
<thead>
<tr>
<th>( i )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l_i )</td>
<td>29</td>
<td>33</td>
<td>45</td>
<td>46</td>
<td>50</td>
<td>83</td>
</tr>
<tr>
<td>( w_i )</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>( d_i )</td>
<td>6</td>
<td>11</td>
<td>15</td>
<td>16</td>
<td>39</td>
<td>75</td>
</tr>
</tbody>
</table>

The computation of an optimal cutting pattern (see Fig. 6) of a rectangular board with \( L = 456, W = 35 \) and \( p = 5 \) defects required 8.0 sec. A second nonrectangular
board was handled as a rectangle with \( L = 418, W = 41 \) and \( p = 30 \) defects. The computation time needed to get an optimal cutting pattern (see Fig. 7) was 22.1 sec.

Table 2 shows the results of several practical examples using the data from Table 1. In the last column the quotient of the CPU-time and a theoretical quantity COMPL is given. Analogous to Formula (3) the computational expense of the improved algorithm into Section 4 is proportional to

\[
xL + x^2\bar{p} (mW + mp^2),
\]

where \( \bar{p} \) is the average number of defects per section. COMPL is defined by the essential part of (4):

\[
\text{COMPL} = \bar{p} (W + \bar{p}^2).
\]

The average computation time of the 15 examples amounts to 78.2 sec.

<table>
<thead>
<tr>
<th>board</th>
<th>length</th>
<th>width</th>
<th>number of defects</th>
<th>CPU-time (sec.)</th>
<th>CPU-time COMPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>430</td>
<td>42</td>
<td>38</td>
<td>38.4</td>
<td>0.15</td>
</tr>
<tr>
<td>2</td>
<td>443</td>
<td>37</td>
<td>73</td>
<td>66.6</td>
<td>0.09</td>
</tr>
<tr>
<td>3</td>
<td>443</td>
<td>42</td>
<td>51</td>
<td>45.5</td>
<td>0.12</td>
</tr>
<tr>
<td>4</td>
<td>446</td>
<td>32</td>
<td>80</td>
<td>62.5</td>
<td>0.07</td>
</tr>
<tr>
<td>5</td>
<td>435</td>
<td>41</td>
<td>53</td>
<td>57.4</td>
<td>0.13</td>
</tr>
<tr>
<td>6</td>
<td>444</td>
<td>42</td>
<td>81</td>
<td>92.5</td>
<td>0.09</td>
</tr>
<tr>
<td>7</td>
<td>418</td>
<td>35</td>
<td>90</td>
<td>85.7</td>
<td>0.06</td>
</tr>
<tr>
<td>8</td>
<td>416</td>
<td>36</td>
<td>136</td>
<td>208.1</td>
<td>0.05</td>
</tr>
<tr>
<td>9</td>
<td>415</td>
<td>40</td>
<td>96</td>
<td>132.1</td>
<td>0.07</td>
</tr>
<tr>
<td>10</td>
<td>417</td>
<td>38</td>
<td>69</td>
<td>72.4</td>
<td>0.09</td>
</tr>
<tr>
<td>11</td>
<td>413</td>
<td>33</td>
<td>114</td>
<td>168.7</td>
<td>0.06</td>
</tr>
<tr>
<td>12</td>
<td>238</td>
<td>32</td>
<td>16</td>
<td>9.4</td>
<td>0.07</td>
</tr>
<tr>
<td>13</td>
<td>427</td>
<td>25</td>
<td>87</td>
<td>79.8</td>
<td>0.07</td>
</tr>
<tr>
<td>14</td>
<td>426</td>
<td>20</td>
<td>80</td>
<td>54.2</td>
<td>0.06</td>
</tr>
<tr>
<td>15</td>
<td>427</td>
<td>26</td>
<td>75</td>
<td>64.0</td>
<td>0.08</td>
</tr>
</tbody>
</table>

AVERAGE: 73.2 78.2
In [3] an example with $L=239$, $W=120$, $n=14$ (corresponding to 7 different pieces which can be rotated) and 13 defects (including the border) is stated and a corresponding cutting pattern is shown. Using the same valuations of the pieces we get the cutting pattern illustrated in Figure 8 which satisfies the technological conditions considered in [3] and which has an optimal value $12.1\%$ greater than in [3].

Fig. 8

In comparison with the used cutting technique a similar (illustrated in Figure 9) is imaginable.

Fig. 9

In that case another, more expensive $H$-computation is necessary. By this the definition of the quantities $\delta(y)$ and $\eta(y)$ has to be carried out in an analogous manner to $\delta(x)$ and $\tau(x)$.

In Chapter 4 several possibilities were investigated to reduce the number of needed $G$- and $H$-computations. Additional reductions of the computational expense can possibly obtained by a further improvement of the algorithms to solve the Knapsack problems.

Acknowledgements: The authors are indebted to the referees for their valuable remarks.
References

1114.
[4] SCHEITHAUER, G.; J. TERNO: Das Pilotprogramm CUT 84 zur Lösung von Zuschnitt-

Received June 1986, revised April 1987

G. Scheithauer
Technische Universität Dresden
Sektion Mathematik
Mommsenstr. 13
Dresden
DDR - 8027

J. Terno
Technische Universität Dresden
Sektion Mathematik
Mommsenstr. 13
Dresden
DDR - 8027