

RECURSIVE ALGORITHM FOR GENERATING TWO-STAGED CUTTING PATTERNS OF PUNCHED STRIPS

Yaodong Cui

Department of Computer Science, Guangxi Normal University, Guilin, 541004, P R China
ydcui@263.net

Abstract- The manufacture of parts made of metal sheet often includes two successive processes: the cutting process at which a guillotine shear cuts the sheet into strips, and the punching process at which a stamping press punches out the blanks from the strips. This paper presents an algorithm for generating optimal two-staged cutting patterns of strips for the cutting process. At the first stage the sheet is divided into segments with parallel cuts. Each segment contains strips with the same length and direction. The segments are cut into strips at the second stage. The algorithm calls a recursion function to determine the optimal strip layouts on segments of various lengths, and calls another recursion function to optimally arrange the segments in the sheet, so that the value of the pattern reaches maximum. The computational results indicate that the algorithm is efficient both in computation time and in material usage.

Keywords- two-dimensional cutting, Strip layout, Stamping, Recursion

1. INTRODUCTION

A factory in the sheet metal industry may consume thousands of tons of metal sheet in a year to make products. The manufacturing process of parts made of metal sheet often includes two successive stages: the cutting stage and the punching stage. At the cutting stage, a guillotine shear divides the sheet into strips with orthogonal cuts. Each strip contains blanks of the same type. Strips of different blank types can appear in a sheet. At the punching stage, a stamping press punches out the blanks from the strips. Typically, a strip is fed into a stamping press that punches out a blank with each stroke, feeds the strip forward, and punches out the next blank. The blank layout on the strip was determined at the tooling design stage. It cannot be changed once the tooling is built. At the cutting stage, the blank layouts on strips are taken as known. Strips of different blank types may have different widths. Nesting strips of different widths in a cutting pattern can improve material usage at this stage.

The following parameters characterize the strips of a blank type: The initial step, the succeeding step, and the strip width. Figure 1a shows a strip with initial step 100, succeeding step 65, and strip width 229. The initial step is the minimum strip length to hold the first blank. The succeeding step is the length increased to hold one additional blank. Figure 1b shows a strip equivalent to that in Figure 1a. Equivalent strips can be used in generating cutting patterns at the cutting stage. That is to say, a strip of a blank type can be taken as consisting of rectangular blanks of at most two sizes, where the length of the first blank is longer than or equal to that of the others. The strips in Figure 1 will be referred to as punched strips, to differentiate them from other types of strips

used in the literature.

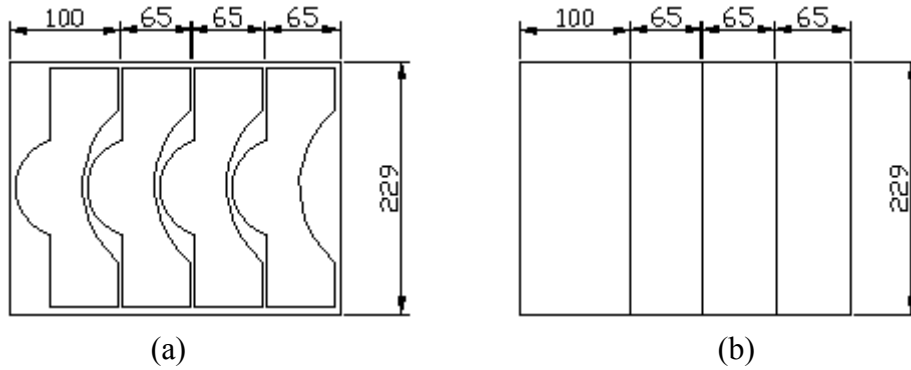


Figure 1. Strips. (a) A strip. (b) A equivalent strip.

The unconstrained two-dimensional cutting problem of strips (UTDC for short) discussed can be formally stated as: m types of strips are to be cut from stock sheet $L \times W$ with a guillotine shear. For the i th strip, $i = 1, \dots, m$, the initial step is a_i , the succeeding step is b_i , the strip width is w_i , and the value of a blank is c_i . Both the sheet size and the strip parameters are positive integers, and there is no constraint on the numbers of blanks appearing in a cutting pattern. The task is to generate the cutting pattern that has the maximum value, which is equal to the sum of the values of the blanks included. Assume that G is a cutting pattern, and x_i is the number of the i th blank type appearing in this pattern. The mathematic model for the UTDC is:

$$z = \max \sum_{i=1}^m c_i x_i; \text{ s.t. } G \text{ is a feasible cutting pattern for the cutting stage} \quad (1)$$

The UTDC is interesting for that it is closely related to the two-dimensional cutting stock problem of strips (TDCS for short). The TDCS is the problem of cutting all required numbers of blanks of different types from a set of rectangular sheets at minimum sheet cost. A solution of the TDCS consists of several cutting patterns, each of which can be generated by an algorithm for the UTDC. The linear programming (LP) approach have been used by many authors to solve cutting stock problems, where the blanks can be of any shape from regular to irregular [1-2]. It certainly can also be used to solve the TDCS. Starting from an initial solution, the LP approach can solve the TDCS according to the following steps:

Step 1. Determine c_i , the current value of a blank of the i th type, $i = 1, \dots, m$, from the current solution.

Step 2. Use an algorithm for the UTDC to solve Model (1) to obtain pattern G . Go to Step 4 if pattern G has a value not larger than that of the sheet used.

Step 3. Replace a pattern in the current solution with pattern G . Go to Step 1.

Step 4. Output the solution.

The LP approach solves the TDCS through iteration consisting of steps 1 to 3. A large number of UTDC must be solved before the LP approach finds a solution close to optimal.

Among the papers dealing with two-dimensional cutting problems, the number of those that use strips in generating cutting patterns is relatively small. Some authors used strips in generating cutting patterns for rectangular blanks [3-4], where both general and uniform strips are used. A general strip contains blanks of different sizes. The width of the blanks in a uniform strip is the same, and the lengths of them may be different. Both general and uniform strips differ from the punched strips. References [2] and [5] presented algorithms for generating cutting patterns of punched strips. The proposed cutting patterns are a subset of the two-staged patterns proposed in this paper. The techniques used in the above papers cannot be directly used for generating good cutting patterns of punched strips.

This paper proposes the application of two-staged cutting patterns for punched strips, and presents an algorithm for them. Both the strip layouts on the segments and the segment layout on the sheet are determined recursively. The algorithm considers all possible segment lengths implicitly, and skips non-promising strips in generating strip layout for each segment. Different versions of the algorithm are tested through experimental computation. The results indicate that the algorithm is efficient both in material usage and in computation time. Finally, solution to an example is given.

2. TWO-STAGED PATTERNS

Figure 2 shows the sketches of the proposed two-staged patterns. A pattern is an X-pattern if the cuts dividing the sheet into segments are vertical; otherwise it is a Y-pattern. Each segment contains strips in the same direction, which is denoted by the arrow in the segment. A segment containing X-strips (horizontal strips) is an X-segment, and that consisting of Y-strips (vertical strips) is a Y-segment.

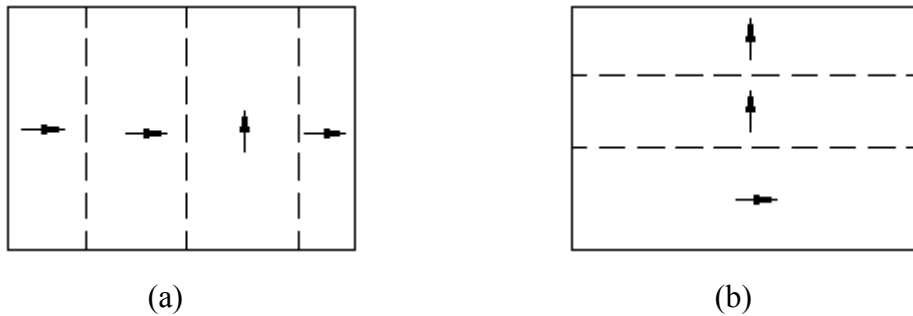


Figure 2. Two-staged patterns. (a) An X-pattern of 4 segments. (b) A Y-pattern of 3 segments

Figure 3 shows two two-staged patterns, where the number in a strip denotes the blank type. The X-pattern in Figure 3a includes three segments. The first segment is a Y-segment consisting of two Y-strips. The other two segments are X-segments. The Y-pattern in Figure 3b contains two Y-segments.

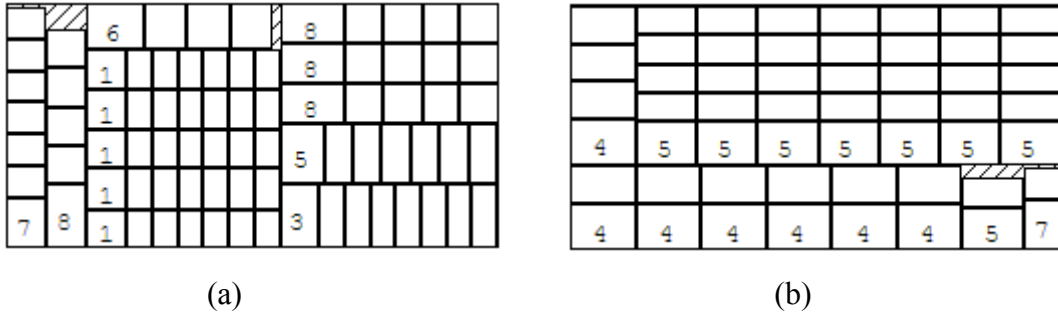


Figure 3. Two-staged patterns. (a) An X-pattern of 3 segments. (b) A Y-pattern of 2 segments.

3. THE SEGMENT LENGTHS THAT SHOULD BE CONSIDERED

To facilitate presentation, the algorithm will be described only for X-patterns, which can be easily extended to deal with Y-patterns.

3.1 Some concepts

Definition 1. Layouts and patterns.

A layout denotes the arrangement of strips on a segment, and a pattern denotes the arrangement of segments on a sheet.

Definition 2. Directed layouts

For a directed layout the strip direction in a segment is fixed. Assume that $F(x, y)$ is the value of the optimal directed layout on segment $x \times y$ or $y \times x$, where the strip direction is along length x . For segment $X \times Y$, the optimal value is $F(X, Y)$ when the segment is an X-segment; it is $F(Y, X)$ when the segment is a Y-segment.

Definition 3. Normal segment lengths for directed layouts

Normal segment lengths are measured along the strip direction. They are the exact lengths of the strips that contain integral steps. For the strips of blank type i , let $x_{i,k}$ be the k th normal length, then

$$x_{i,k} = a_i + (k-1)b_i, \quad k = 1, \dots, \text{int}[(L - a_i)/b_i], \quad i = 1, \dots, m$$

Let P_X be the set of normal lengths of all blank types. Add 0, W and L to this set. Arrange the elements in increasing order of their values, namely $P_X = \{p_1, \dots, p_M\}$, $p_1 = 0$, $p_i < p_{i+1}$ for $1 \leq i < M$, and $p_M = L$.

Normal segment lengths have the following property: Assume that $p(x)$ is the maximum normal length not larger than x , then $F(x, y) = F[p(x), y]$. The reason can be interpreted as follows. The length of the strips in segment $x \times y$ is x . If x is not a normal length, it is not a length that can be occupied by integral steps of any blank type. Therefore, shortening it to $p(x)$ will not change the number of blanks in each strip. As a result, the value of segment $p(x) \times y$ is the same as that of segment $x \times y$.

Definition 4. Normal segment widths for directed layouts

Normal segment widths are measured in the direction perpendicular to the strip direction. They are the exact widths occupied by integral strips. Assume that y is a

normal width, then:

$$y = \sum_{i=1}^m z_i w_i, \quad z_i \text{ non-negative integers, } 1 \leq i \leq m, \quad 0 \leq y \leq W.$$

Let \mathbf{Q} be the set of normal widths. Add 0 and W to this set. Arrange the elements in increasing order of their values, namely $\mathbf{Q} = \{q_1, \dots, q_N\}$, $q_1 = 0$, $q_i < q_{i+1}$ for $1 \leq i < N$, and $q_N = W$.

Normal segment widths have the following property: Assume that $q(y)$ is the maximum normal width not larger than y , then $F(x, y) = F[x, q(y)]$. The reason can be interpreted in a similarly way to that for normal lengths.

3.2 The segment lengths that should be considered

The lengths of the segments in an X-pattern are measured horizontally. Assume that G is an optimal X-pattern and contains X-segment $x \times W$. From Definition 3, replacing it with X-segment $p(x) \times W$ will not change the pattern value, i.e., there exists an optimal pattern, where the lengths of the X-segments all belong to set \mathbf{P}_X .

Assume that pattern G contains Y-segment $x \times W$. From Definition 4, replacing it with Y-segment $q(x) \times W$ will not change the value of the pattern. Furthermore, assume that the number of Y-strips in segment $q(x) \times W$ is n . This segment can be taken as consisting of n Y-segments, each of which contains only one Y-strip. That is to say, there exists an optimal pattern, where each Y-segment contains only one strip. Therefore, the set of Y-segment lengths that should be considered is $\mathbf{P}_Y = \{w_1, \dots, w_m\}$.

Assume that \mathbf{P} is the set of segment lengths that should be considered in searching for the optimal X-pattern, then $\mathbf{P} = \mathbf{P}_X \cup \mathbf{P}_Y$ or $\mathbf{P} = \{p_1, \dots, p_M, \dots, p_{M+m}\}$, where $p_{M+i} = w_i$, $i = 1, \dots, m$. Usually $M + m$ is much more smaller than L . Considering only the segment lengths that belong to \mathbf{P} can shorten the computation time.

4. THE RECURSION FUNCTIONS

4.1 The recursion function for the optimal strip layout on a segment

Assume that $x \times w_i$ is a strip of blank type i , $n(i, x)$ is the number of blanks in the strip, and $v(i, x)$ is the value of the strip, then

$$\begin{aligned} n(i, x) &= 0 \quad \text{for } x < a_i; \quad n(i, x) = 1 + \text{int}[(x - a_i)/b_i] \quad \text{for } x \geq a_i \\ v(i, x) &= c_i n(i, x) \end{aligned} \tag{2}$$

The optimal directed strip layout on segment $x \times y$ is recursively determined as:

$$F(x, y) = \max \{v(i, x) + F(x, y - w_i) \mid y - w_i \geq 0\}; \quad F(x, y) = 0 \quad \text{for } y = 0$$

Let $F(x, y) = \text{SegRec}(x, y)$, where function $\text{SegRec}(x, y)$ returns the value of the optimal directed strip layout on rectangle $x \times y$, with the strip direction being along the side of length x . Initially let $F(x, y) = -1$ for $x \in \mathbf{P}_X$ and $y \in \mathbf{Q}$. The steps of $\text{SegRec}(x, y)$ are:

Step 1. Let $y = q(y)$. Go to Step 7 if $F(x, y) \geq 0$. Let $F(x, y) = 0$ and go to Step 7 if

$y = 0$.

Step 2. Let $F(x, y) = 0$. Let $i = 1$.

Step 3. Let $y_1 = y - w_i$. Go to Step 6 if $y_1 < 0$.

Step 4. Let $u = v(i, x) + \text{SegRec}(x, y_1)$. Go to Step 6 if $u \leq F(x, y)$.

Step 5. Let $F(x, y) = u$.

Step 6. Let $i = i + 1$. Go to Step 3 if $i \leq m$; otherwise go to Step 7.

Step 7. Return $F(x, y)$.

The recursion is terminated in the following cases: (1) In Step 1 when $y = 0$; (2) In Step 1 when $F(x, y) \geq 0$, for that the solution to segment $x \times y$ has already been known; and (3) In Step 6 when all strips have been considered, namely when $i > m$.

4.2 The recursion function for the optimal pattern

Recall that only the segment lengths that belong to set \mathbf{P} should be considered. Segment $p_i \times W$ is an X-segment when $i \leq M$. It is a Y-segment containing only one strip when $M + 1 \leq i \leq M + m$. Assume that $U(i)$ is the value of segment $p_i \times W$, $i = 1, \dots, M + m$, $E(x)$ is the value of the optimal pattern on partial sheet $x \times W$, then

$$E(x) = \max\{U(i) + E(x - p_i) \mid x - p_i \geq 0\}; \quad E(x) = 0 \quad \text{for } x \leq 0.$$

Let $E(x) = \text{PatRec}(x)$, where function $\text{PatRec}(x)$ returns the value of the optimal pattern on partial sheet $x \times W$. Initially let $E(0) = 0$ and $E(x) = -1$ for $1 \leq x \leq L$. The steps of $\text{PatRec}(x)$ are:

Step 1. Go to Step 7 if $E(x) \geq 0$.

Step 2. Let $E(x) = 0$ and $i = 1$.

Step 3. Go to Step 6 if $x < p_i$.

Step 4. Let $u = U(i) + E(x - p_i)$. Go to Step 6 if $u \leq E(x)$.

Step 5. Let $E(x) = u$.

Step 6. Let $i = i + 1$. Go to Step 3 if $i \leq M + m$; otherwise go to Step 7.

Step 7. Return $E(x)$.

5. THE ALGORITHM

Assume that the value of the optimal X-pattern is V_X , the value of the optimal Y-pattern is V_Y , and the value of the global optimal pattern is V^* , then $V^* = \max(V_X, V_Y)$. This section presents the algorithm for determining V_X . It can be easily adapted for determining V_Y . Four versions of the algorithm will be presented.

5.1 The basic version (Version A1)

Version A1 consists of the following steps:

Step 1. Let $\mathbf{P}_X = \{0, 1, \dots, L\}$, $\mathbf{Q} = \{0, 1, \dots, W\}$, and $\mathbf{P} = \{0, 1, \dots, L, w_1, \dots, w_m\}$. The number of elements in \mathbf{P}_X is $M = L + 1$, and that in \mathbf{Q} is $N = W + 1$.

Step 2. From Equation (2), obtain the strip values $v(i, x)$ for $x \in \mathbf{P}_X$, $i = 1, \dots, m$.

Step 3. Let $F(x, y) = -1$ for $x \in \mathbf{P}_X$ and $y \in \mathbf{Q}$. Let $E(0) = 0$, and $E(x) = -1$

for $1 \leq x \leq L$.

Step 3. Let $U(i) = \text{SegRec}(p_i, W)$ for $1 \leq i \leq M$.

Step 4. Let $U(i) = v(i, W)$ for $i = M + 1, \dots, M + m$.

Step 5. Let $V_x = \text{PatRec}(L)$.

5.2 The version based on normal lengths and widths (Version A2)

The steps of Version A2 are the same as those of A1 except Step 1, which is as follows:

Step 1. Determine P_x and Q from definitions 3 and 4. Determine P according to Section 3.2.

5.3 The version based on the solutions for shorter segments (Version A3)

Version A3 is an improved version of A1. It uses the solution to a shorter segment as the initial solution to the current segment, and skips some non-promising strips.

In Step 3 of Version A1 and A2, function $\text{SegRec}()$ is called M times to find the value of the optimal layouts on X-segments $p_i \times W$, $i = 1, \dots, M$. Recall that $p_i < p_{i+1}$ for $1 \leq i < M - 1$. Assume that $1 < j < i \leq M$ and the solution to segment $p_j \times y$ is known, namely $F(p_j, y) \geq 0$. This solution is also feasible to segment $p_i \times y$, for that $p_j < p_i$. Let this solution be the initial solution to segment $p_i \times y$. This should be done in Step 2 of function $\text{SegRec}()$. In steps 3 to 6 of $\text{SegRec}()$, m types of strips are considered one by one. A strip of width w_i can be skipped, if $v(i, p_i) = v(i, p_j)$. The reason is that introducing a strip of width w_i and value $v(i, p_j) = v(i, p_i)$ into the solution to segment $p_j \times y$ cannot improve the solution, for that the solution is already optimal. As a result, introducing the same strip into the initial solution to segment $p_i \times y$ cannot improve the solution either, for that this initial solution is actually the optimal solution to segment $p_j \times y$.

5.4 The final algorithm (Version A4)

Version A4 differs from A3 only in Step 1. Step 1 of A3 is the same as that of A1, whereas Step 1 of A4 is the same as that of A2.

6. THE COMPUTATIONAL RESULTS

6.1 Test on random problems

The computations were performed with a computer of clock rate 2.8GHz and inner memory 512MB. Fifty random test problems were used to test all versions of the algorithm. These problems were generated according to the uniform distributions in the following variable ranges: Sheet length in [2000, 2500], sheet width in [1000, 1250], initial step and strip width both in [100, 450]. Each problem includes 30 blank types. The succeeding step $b_i = \text{int}(r_i a_i)$, where r_i is uniformly distributed in [0.6, 0.9]; the value is $c_i = b_i w_i$, $i = 1, \dots, 30$. Once the paper is published, these problems will be made available on the Internet:

<http://www.gxnu.edu.cn/Personal/ydcui/Index.asp>

The material usage is used to measure the quality of a pattern, which is determined as:

$$u = [(\sum_{i=1}^m c_i x_i) / (L \times W)] \times 100 (\%)$$

Here u is the material usage, and x_i is the number of the i th blank type appearing in the pattern. The average material usage is 97.7747%. The average computation time of one problem is 2.741s for A1, 0.361s for A2, 0.298s for A3, and 0.124s for A4. Version A2 considers only normal segment lengths and widths. It is much more efficient than A1; Version A3 uses the solution to shorter segments as the initial solution to the current segment, and skips non-promising strips. It is also much more efficient than A1. Version A4 uses both techniques. It is the most efficient one among all versions.

6.2 Solution to a cutting stock problem

Six types of blanks are to be cut from sheet 2000×1200. Assume that d_i is the demand for the i th blank type. The data of the blank types are as follows ($a_i \times b_i \times w_i \times d_i$):

144×127×224×9720, 277×255×256×5647, 234×206×185×8725, 348×286×208×8985

A LP approach similar to that in [2] is combined with the algorithm of this paper to solve the example. Figure 4 shows the solution that contains four patterns. The total number of sheets used to meet the blank demand is 670 pieces, and the material usage is 94.04%. Pattern 4 is a Y-pattern, and the others are X-patterns. Each pattern contains two segments except Pattern 3, which consists of three segments.

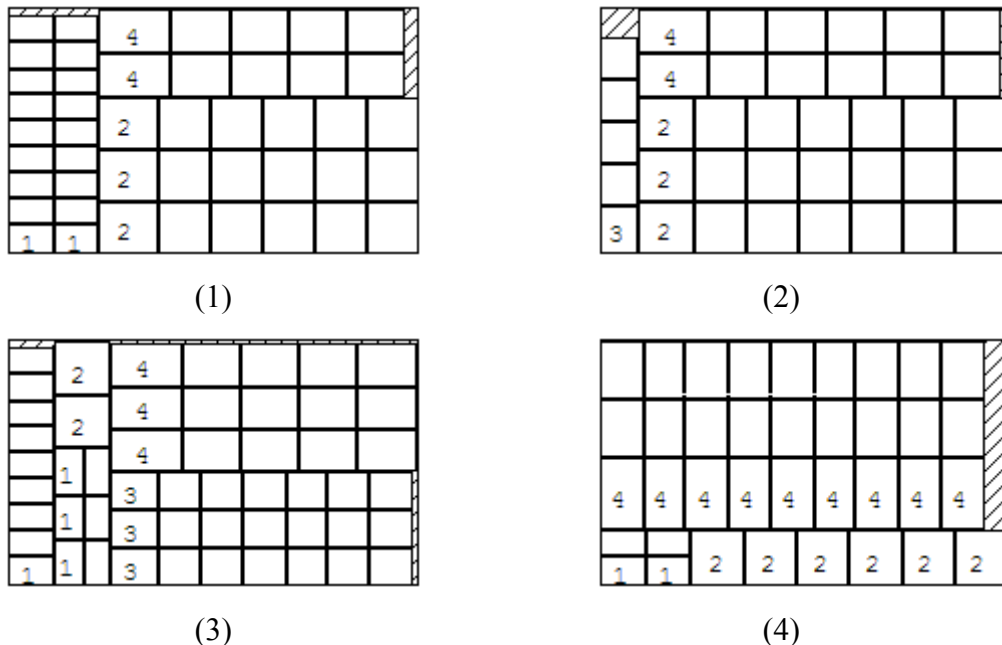


Figure 4. Cutting patterns. (1) 202 pieces. (2) 55 pieces. (3) 403 pieces (4) 10 pieces

7. CONCLUSIONS

For factories in the sheet metal industry, the number of blank types processed by the

cutting and punching processes may range from 5 to 50. Generating good cutting patterns is helpful for improving material usage at the cutting stage. The algorithm presented in this paper can generate optimal two-staged cutting patterns feasible for the cutting stage. The computation time is reasonable for practical use.

Combined with the LP approach, the algorithm may be used to solve the cutting stock problem, where the variables are the numbers of sheets cut according to each pattern. The values of the variables must be rounded to integers. LP approach is not very adequate for the case where the demands for the blank types are smaller, for that the rounding procedure will cause the solution to departure from the optimal solution. To deal with the above situation, it is better to use an algorithm for the constrained two-dimensional cutting problem of strips (CTDC for short). The CTDC differs from the UTDC in that there exists constraint on the number of each blank type appearing in the cutting pattern. The two-staged pattern may be used in developing the algorithms for the CTDC.

REFERENCES

1. J. M. Valerio de Carvalho, LP models for bin packing and cutting stock problems, *European Journal of Operational Research* 14, 253-273, 2002.
2. Y. Cui, Generating optimal T-shape cutting patterns for circular blanks, *Computers & Operations Research*, 32, 143-152, 2005.
3. R. Morabito, M. N. Arenales, Staged and constrained two-dimensional guillotine cutting problems: An AND/OR-graph approach, *European Journal of Operational Research* 94, 548-560, 1996.
4. J. Riehme, G. Scheithauer, J. Terno, The solution of two-stage guillotine cutting stock problems having extremely varying order demands, *European Journal of Operational Research* 91, 543-552, 1996
5. Y. Cui, A cutting stock problem and its solution in the manufacturing industry of large electric generators, *Computers & Operations Research* 32, 1709-1721, 2005