European Journal of Operational Research 54 (1991) 141–150
North-Holland

Invited Review

Cutting stock problems and solution procedures

Robert W. Haessler and Paul E. Sweeney
School of Business Administration, The University of Michigan, Ann Arbor, MI, USA

Received May 1991

Abstract: This paper discusses some of the basic formulation issues and solution procedures for solving one- and two- dimensional cutting stock problems. Linear programming, sequential heuristic and hybrid solution procedures are described. For two-dimensional cutting stock problems with rectangular shapes, we also propose an approach for solving large problems with limits on the number of times an ordered size may appear in a pattern.

Keywords: Cutting stock, trim loss, linear programming, heuristic problem solving, pattern generation, two-dimensional knapsack

Introduction

The first known formulation of a cutting stock problem was given in 1939 by the Russian economist Kantorovich (1960). The first and most significant advance in solving cutting problems was the seminal work of Gilmore and Gomory (1961, 1963) in which they described their delayed pattern generation technique for solving the one-dimensional trim loss minimization problem using linear programming. Since that time there has been an explosion of interest in this application area. Sweeney and Paternoster (1991) have identified more than 500 papers which deal with cutting stock and related problems and applications. The primary reasons for this activity are that cutting stock problems occur in a wide variety of industries, there is a large economic incentive to find more effective solution procedures, and it is easy to compare alternative solution procedures and to identify the potential benefits of using a proposed procedure.

The large variety of applications reported in the literature has led Dyckhoff (1990) to develop a classification scheme for cutting stock and packing problems (packing problems are closely related to cutting stock problems but are not considered here). He classifies problems using four characteristics as follows:

1. *Dimensionality*
   (N) Number of dimensions
2. *Kind of assignment*
   (B) All large objects and a selection of small items.
   (V) A selection of large objects and all small items.
3. *Assortment of large objects*
   (O) One large object.
   (I) Many identical large objects.
   (V) Different large objects.
4. *Assortment of small items*
   (F) Few items of different dimensions.
   (M) Many items of many different dimensions.
   (R) Many items of relatively few dimensions.
   (C) Many identical items.

Cutting stock problems are introduced with a discussion of the one-dimensional problem in which many items of relatively few sizes are to be cut from multiple pieces of a single stock size (1/V/I/R using Dyckhoff's typology). Two-dimensional cutting stock problems are more difficult to solve than one-dimensional problems because of the greater complexity of defining feasible cutting patterns. Hence the focus in two-dimensional problems is on the pattern generation process rather than on the cutting stock problem itself. The paper will conclude with a discussion of a possible new approach for generating patterns needed to solve two-dimensional cutting stock problems of type 2/V/I/R.

One-dimensional problems

An example of a one-dimensional cutting stock problem is the trim loss minimization problem which occurs in the paper industry. In this problem, known quantities of rolls of various widths and the same diameter are to be slit from stock rolls of some standard width and diameter. The objective is to identify slitting patterns and their associated usage levels which satisfy the requirements for ordered rolls at the least possible total cost for scrap and other controllable factors. The basic cutting pattern feasibility restriction in this problem is that the sum of the roll widths slit from each stock roll must not exceed the usable width of the stock roll.

Let R_i be the nominal order requirements for rolls of width W_i, $i = 1, \ldots, m$, to be cut from stock rolls of usable width UW. RL_i and RU_i are the lower and upper bounds on the order requirement, for customer order i reflecting the general industry practice of allowing overruns or underruns within specified limits. Depending on the situation, R_i may be equal to RL_i and/or RU_i. All orders are for rolls of the same diameter. This problem can be formulated as follows:

$$\text{Min} \quad \sum_j T_j X_j \tag{1}$$

$$\text{s.t.} \quad \text{RL}_i \le \sum_j A_{ij} X_j \le \text{RU}_i \quad \text{for all } i, \tag{2}$$

$$X_j \ge 0 \quad \text{and integer}, \tag{3}$$

where

A_{ij} is the number of rolls of width W_i to be slit from each stock roll that is processed using pattern j. In order for the elements A_{ij}, $i = 1, \ldots, m$, to constitute a feasible cutting pattern, the following restrictions must be satisfied:

$$\sum_i A_{ij} W_i \le \text{UW}, \tag{4}$$

$$A_{ij} \ge 0 \quad \text{and integer}, \tag{5}$$

X_j is the number of stock rolls to be slit using pattern j, and

T_j is the trim loss incurred by pattern j,

$$T_j = \text{UW} - \sum_i A_{ij} W_i. \tag{6}$$

Note that the objective in this example is simply to minimize trim loss. In most industrial applications, it is necessary to consider other factors in addition to trim loss. For example, there may be a cost associated with pattern changes and, therefore, controlling the number of patterns used to satisfy the order requirements would be an important consideration.

Because optimal solutions to integer cutting stock problems can be found only for values of m smaller than typically found in practices, heuristic procedures represent the only feasible approach to solving this type of problem. Two types of heuristic procedures have been widely used to solve one-dimensional cutting stock problems. One approach uses the solution to a linear programming (LP) relaxation of the integer problem above as its starting point. The LP solution is then modified in some way to provide an integer solution to the problem. The second approach is to generate cutting patterns sequentially to satisfy some portion of the remaining requirements. This sequential heuristic procedure (SHP) terminates when all order requirements are satisfied.

Linear programming solutions

Almost all LP based procedures for solving cutting stock problems can be traced back to Gilmore and Gomory (1961, 1963). They described how the next pattern to enter the LP basis could be found by solving an associated knapsack problem. This made it possible to solve the trim loss minimization problem by linear programming without first enumerating every feasible slitting pattern. This is extremely important because a large number of feasible patterns may exist when narrow widths are to be slit from a wide stock roll. Pierce (1964) showed that in such situations the number of slitting patterns can easily run into the millions. Because only a small fraction of all possible slitting patterns need to be considered in finding the minimum trim loss solution, the delayed pattern generation technique developed by Gilmore and Gomory made it possible to solve trim loss minimization problems in much less time than would be required if all the slitting patterns were input to a general purpose linear programming algorithm.

A common LP relaxation of the integer programming problem given in (1)–(3) can be stated as follows:

$$\text{Min} \quad \sum_j X_j \tag{7}$$

$$\text{s.t.} \quad \sum_j A_{ij} X_j \geq R_i \quad \text{for all } i, \tag{8}$$

$$X_j \geq 0. \tag{9}$$

Let U_i be the dual variable associated with constraint i.

The dual of this problem can be stated as

$$\text{Max} \quad \sum_i R_i U_i \tag{10}$$

$$\text{s.t.} \quad \sum_i A_{ij} U_i \leq 1, \tag{11}$$

$$U_i \geq 0. \tag{12}$$

The dual constraints in (11) provide the means for determining if the optimal LP solution has been obtained or if there exists a pattern which will improve the LP solution because the dual problem is still infeasible.

The next pattern $A = (A_1, \ldots, A_m)$ to enter the basis, if one exists, can be found by solving the following knapsack problem:

$$Z = \text{Max} \sum_i U_i A_i \tag{13}$$

$$\text{s.t.} \quad \sum_i W_i A_i \leq \text{UW}, \tag{14}$$

$$A_i \geq 0 \quad \text{and integer}. \tag{15}$$

If $Z \leq 1$, the current solution is optimal. If $Z > 1$, then A can be used to improve the LP solution.

Once found, the LP solution can be modified in a number of ways to obtain integer values for the X_j which satisfy the order requirements. One common approach is to round the LP solution down to integer values, then increase the values of X_j by unit amounts for any patterns whose usage can be increased without exceeding RU_i. Finally, new patterns can be generated for any rolls still needed using the sequential heuristic described in the next section.

In order to make this rounding problem as simple as possible, it is generally useful to place limitations on the number of times a given size can appear in a pattern. A very obvious restriction is that $A_i \leq \text{RU}_i$. If this is not satisfied, any pattern for which $A_i > \text{RU}_i$ will have to be rounded down to zero and a new pattern found by some other method. The following simple example demonstrates the advantage of placing even greater restrictions on the values of A_i. Let

$R_1 = 3, \quad W_1 = 100,$

$R_2 = 3, \quad W_2 = 90,$

$\text{UW} = 200.$

Solving this problem using the Gilmore–Gomory algorithm yields the following solution:

Pattern 1. 2–100 and 0–90, $X_1 = 1.5$.

Pattern 2. 0–100 and 2–90, $X_2 = 1.5$.

In virtually all situations, the preferred solution would be:

Pattern 3. 1–100 and 1–90, $X_3 = 3$.

Haessler (1980) demonstrated how placing restrictions on the values of A_i in the knapsack problem led to LP solutions with fewer patterns which are easier to round to integer values.

The primary disadvantage of using LP to solve cutting stock problems is that the number of active cutting patterns in the solution will be very close to the number of sizes ordered. This may be acceptable only if controlling trim loss is very difficult and LP is the only way to find a low trim loss solution.

Sequential heuristic procedures

With an SHP, a solution is constructed one pattern at a time until all the order requirements are satisfied. The first documented SHP capable of finding better solutions than those found manually by schedulers was described by Haessler (1971). The key to success with this type of procedure is to make intelligent choices as to the patterns which are selected early in the SHP. The patterns selected initially should have low trim loss, high usage and leave a set of requirements for future patterns which will combine well without excessive side trim.

The following procedure is capable of making effective pattern choices in a variety of situations:

1. Compute descriptors of the order requirements yet to be scheduled. Typical descriptors would be the number of stock rolls still to be slit and the average number of ordered rolls to be cut from each stock roll.
2. Set goals for the next pattern to be entered into the solution. Goals should be established for trim loss, pattern usage, and number of ordered rolls in the pattern.
3. Search exhaustively for a pattern that meets those goals.
4. If a pattern is found, add this pattern to the solution at the maximum possible level without exceeding R_i for all i. Reduce the order requirements and return to 1.
5. If no pattern is found, reduce the goal for the usage level of the next pattern and return to 3.

The pattern usage goal provides an upper bound on the number of times a size can appear in a pattern. For example, if some ordered width has an unmet requirement of 10 rolls and the pattern usage goal is 4, that width may not appear more than twice in a pattern. If after exhaustive search no pattern satisfies the goals set, then at least one goal, most commonly pattern usage, must be relaxed. This increases the number of patterns to be considered. If the pattern usage goal is changed to 3 in the above example, then the width can appear in the pattern three times. Termination can be guaranteed by selecting the pattern with the lowest trim loss at the usage level of one.

The primary advantage of this SHP is its ability to control factors other than trim loss and to eliminate rounding problems by working only with integer values. For example, if there is a cost associated with a pattern change, a sequential heuristic procedure which searches for high usage patterns may give a solution which has less than one-half the number of patterns required by an LP solution to the same problem. The major disadvantage of an SHP is that it may generate a solution which has greatly increased trim loss because of what might be called ending conditions. For example, if care is not taken as each pattern is accepted and the requirements reduced, the widths remaining at some point in the process may not have an acceptable trim loss solution. Such would be the case if only 34-inch rolls are left to be slit from 100-inch stock rolls.

Hybrid solution procedures

In addition to using an SHP to help convert an LP solution to integer values as described earlier, there are a number of ways in which these approaches can be used together to obtain the best possible answer to a given class of one-dimensional cutting stock problems. Perhaps the most obvious is to use the SHP to generate a solution which is saved and also used as the initial basis in the LP procedure. Additional LP iterations are then made to reduce the trim loss if that is possible. The better of the SHP and rounded LP solutions is selected according to the appropriate criterion for the problem being solved.

A second and more powerful hybrid solution procedure works as follows. The problem is first solved as an LP problem in order to obtain the optimal dual prices. These dual prices are used as an additional test before accepting a pattern in an SHP to ensure that the pattern does not contain a disproportionate share of sizes with relatively low dual prices. For patterns in the optimal trim loss solution

$$Z = \sum_i A_i U_i = 1.$$

If the value of Z is low (less than 0.97) for a pattern accepted in an SHP, the total trim loss of the SHP solution may be increased significantly. Although this test makes it possible to avoid making some mistakes when selecting a pattern using an SHP, it is not foolproof because the SHP may use the pattern at too high a level.

As the SHP nears completion and the pattern selection decision becomes more difficult, patterns for all residual requirements are generated using LP. If the residual LP solution does not meet some target trim value which is based on the original LP solution to the entire problem, the sequentially generated patterns are dropped one at a time in reverse order of generation and the expanded residual problem is solved using LP. This process of dropping sequentially generated patterns continues until either a satisfactory solution is obtained or all the patterns are dropped at which point the LP solution with the best possible trim loss is generated.

The advantage of this approach is that it integrates the ability of the SHP to consider factors such as slitter changes and the LP procedure to minimize trim loss into a single procedure. This procedure is capable of giving either a pure SHP or LP solution depending on which is best. Most importantly, however, is its ability to generate solutions which are part SHP and part LP and therefore likely to be better than either the pure LP or SHP solutions. Haessler (1988) has used this procedure to solve difficult trim loss problems for which it is also important to limit the number of slitter changes.

Sweeney and Haessler (1990) used information about optimal dual prices to develop a procedure for solving a one-dimensional cutting stock problem with quality variations across the width of the stock rolls. The lower quality material in the stock roll is not scrapped because there are orders which can be filled from the lower quality material. Higher quality material can also be used to satisfy any orders for lower grades. A two phase procedure is used. In the first phase, each nonperfect stock roll is assigned a value based on the nature of its quality variations. Patterns are generated and given a value based on the shadow prices obtained from solving an LP problem. If the value of the selected pattern exceeds the value of the stock roll, the pattern is accepted for that one stock roll. In the second phase, any order requirements not met from stock rolls with quality variations are slit from first quality stock rolls based on an LP solution to the residual problem.

One-dimensional problems with multiple stock sizes

A very interesting problem with multiple stock sizes occurs when the stock sizes are available at different locations and therefore freight cost also influences the choice of which stock size will be used.

This problem can be formulated as follows:

$$\text{Min} \quad \sum_k \left(C_{1k} \sum_j T_{jk} X_{jk} + \sum_i \sum_j C_{2ki} A_{ijk} X_{jk} \right) \tag{16}$$

$$\text{s.t.} \quad \mathrm{RL}_i \le \sum_k \sum_j A_{ijk} X_{jk} \le \mathrm{RU}_i \quad \text{for all } i, \tag{17}$$

$$\sum_j X_{jk} \le M_k \quad \text{for all } k, \tag{18}$$

$$X_{jk} \ge 0 \quad \text{and integer} \tag{19}$$

where

A_{ijk} is the number of rolls for order i to be cut from stock width k using pattern j,

X_{jk} is the number of stock of width k to be processed according to pattern j,

T_{jk} is the trim loss incurred by using pattern j with stock width k,

C_{1k} is the dollar value of trim loss per unit for stock width k,

C_{2ki} is the cost of shipping one roll for order i which is produced from stock width k. It is assumed that the stock width defines the production location. If all the production options are at the same location, this value can be set to 0, and

M_k is the maximum number of rolls of stock width k which can be used.

The LP relaxation of this problem developed by Beged Dov (1970) is

$$\text{Min} \quad \sum_j \sum_k C_{jk} X_{jk} \tag{20}$$

$$\text{s.t.} \quad \sum_j \sum_k A_{ijk} X_{jk} \ge R_i \quad \text{for all } i, \tag{21}$$

$$X_{jk} \ge 0. \tag{22}$$

For the most general case with varying costs for trim loss and freight, C_{jk} can be represented as follows:

$$C_{jk} = C_k + C_{pk}\left(\sum_i A_{ijk} W_i \right) + \sum_i C_{2ki} A_{ijk} \tag{23}$$

where

C_k is the cost excluding material of making one stock roll of width k,

C_{pk} is the material cost per inch of that portion of production roll k actually used, and

C_{2ki} is the cost of shipping one roll for order i from the location where stock width k is made.

In this situation, the shadow prices must be adjusted before finding the next cutting pattern that should enter the LP basis, if one exists. The following knapsack problem must be solved for each stock width.

$$Z_k = \max \sum_i \left(U_i - C_{pk} W_i - C_{2ki} \right) A_{ijk} - C_k, \tag{24}$$

$$\text{s.t.} \quad \sum_i W_i A_{ijk} \leq \text{UW}_k, \tag{25}$$

$$A_{ijk} \geq 0 \quad \text{and integer.} \tag{26}$$

The pattern for which Z_k has the largest value greater than zero is the one which enters the solution.

1.5-dimensional problems

There are a number of cutting stock problems which are more complex than the one-dimensional problems discussed previously, but are not true two-dimensional problems. These are generally referred to as 1.5-dimensional. Haessler and Talbot (1983) discussed an example of this type of problem which occurs in the production of corrugated shipping containers. In this situation a customer order will be for R_i blanks of width W_i and length L_i. The corrugated material is produced continuously out of a variety of available rollstock widths. Slitters and cutoff knives can be set to produce appropriate size blanks. The number of cutoff knives, which is commonly two, determines the number of different orders which can be combined across the width of the corrugator.

Although rectangular blanks are being cut, it is not a two-dimensional problem because trim loss along the length of the corrugator is not a dimensional issue. The problem is more complex than the one-dimensional problem because of the desire to match up orders both across the width and along the length dimensions. The total set of factors to be considered is:

- roll stock changes
- slitter changes
- minimum runs of a slitter setup
- order contiguity
- corrugator width utilization
- roll stock availability
- side trim

If the problem is formulated with pattern usage variables as shown earlier, the resulting formulation is extremely complex. A more effective way to deal with this problem is to solve it using a two-stage process. In the first stage, all possible corrugator setups which will completely produce one, two or three orders from a single roll stock size are generated. All the restrictions on setup feasibility must be met. The cost of any setup for producing a subset of orders, can be completely determined except for the cost of the roll stock changeover which depends on which other setups are selected.

In the second stage the least cost means of producing each order can be determined by solving the following set partitioning problem:

$$\text{Min} \quad \sum_j C_j X_j + \sum_k S_k Y_k \tag{27}$$

$$\text{s.t.} \quad \sum_j A_{ij} x_j = 1 \quad \text{for all } i, \tag{28}$$

$$\sum_j F_{jk} X_j \leq M_k Y_k \quad \text{for all } k, \tag{29}$$

$$X_j, \ Y_k = 0 \text{ or } 1 \quad \text{for all } j, k \tag{30}$$

where

X_j is 1 if element j is used and 0 otherwise,

Y_k is 1 if stock size k is used and 0 otherwise,

A_{ij} is 1 if order i appears in element j and 0 otherwise,

F_{jk} is the lineal quantity of stock size k required by setup j,

M_k is the lineal quantity of stock size k available in inventory,

C_j is the total cost of using element j exclusive of the cost of changing to the required roll stock size, and

S_k is the cost of changing to stock size k.

The value C_j includes the cost of corrugator time and paper used plus the cost of pattern changes. If any order is not produced at the maximum quantity, this value is scaled to reflect the cost of producing the order at its maximum quantity, which typically is 110% of the quantity ordered. This scaling is needed to avoid selecting elements simply because the order quantities produced are at the lower end of the acceptable range. It is assumed that if two or more elements use the same stock size, they will be run sequentially so there will be only one setup for each stock size.

Rectangular two-dimensional problems

The formulation of a higher dimensional cutting stock problem is exactly the same as that of the one-dimensional problem given in (1)–(3). The only added complexity comes in trying to define and generate feasible cutting patterns. The sim-

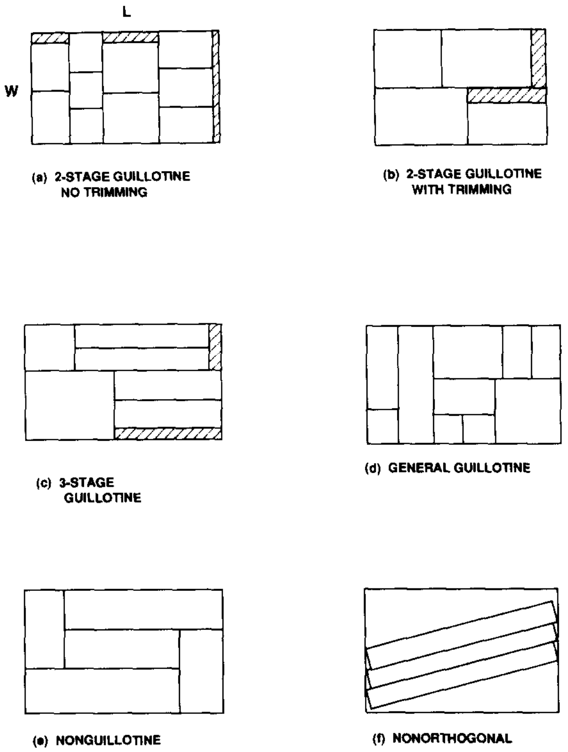

Figure 1. Sample cutting patterns

plest two-dimensional case is one in which both the stock and ordered sizes are rectangular. Most of the important issues regarding cutting patterns for rectangular two-dimensional problems can be seen in the examples shown in Figure 1.

One important issue not covered in Figure 1 is a limit on the number of times an ordered size can appear in a pattern. This generally is a function of the maximum quantity of pieces, RU_i, required for order i. If R_i is small, it is just as important for the two-dimensional case as the one dimensional case that the number of times size i appears in a pattern should be limited. This becomes less important as R_i becomes larger and as the difference between RU_i and RL_i becomes larger.

The cutting pattern shown in Figure 1 (a) is an example of two-stage guillotine cuts. The first cut can be in either the horizontal or vertical direction. A section cut perpendicular to the first, yields a finished piece. Figure 1 (b) is similar except a third cut can be made to trim the pieces down to the correct dimension. Figure 1 (c) shows the situation in which the third cut can create 2 ordered pieces.

For simple staged cutting such as shown in Figure 1 (a–c), Gilmore and Gomory (1965) showed how cutting patterns can be generated by solving two one-dimensional knapsack problems. To simplify the discussion, assume that the orientation of each ordered piece is fixed relative to stock piece and the first guillotine cut on the stock pieces must be along the length (larger dimension) of the stock piece. For each ordered width W_k find the contents of a strip of width W_k and length L which gives the maximum contribution to dual infeasibility.

$$Z_k = \text{Max} \sum_{i \in I_k} U_i A_{ik} \tag{31}$$

$$\text{s.t.} \quad \sum_{i \in I_k} L_i A_{ik} \leq L, \tag{32}$$

$$A_{ik} \geq 0 \quad \text{and integer}, \tag{33}$$

$$I_k = \{ i \mid W_i \leq W_k \}. \tag{34}$$

Next find the combination of strips which solve the following problem:

$$Z = \text{Max} \sum_k Z_k A_k \tag{35}$$

$$\text{s.t.} \quad \sum_k W_k A_k \leq W, \tag{36}$$

$$A_k \geq 0 \quad \text{and integer}. \tag{37}$$

Any pattern for which Z is greater than one will yield an improvement in the LP solution.

The major difficulty with this approach is the inability to limit the number of times and ordered size appears in a pattern. It is easy to restrict the number of times a size appears in a strip and to restrict the number of strips in a pattern. The problem is that small ordered sizes with small quantities may end up as filler in a large number of different strips. This makes the two-stage approach to developing patterns ineffective when the number of times a size appears in a pattern must be limited.

Gilmore and Gomory (1966) also developed dynamic programming recursions for generating patterns both for staged and general guillotine cuts. Their recursion for generating patterns with general guillotine cuts such as shown in Figure 1 (d) is:

$$G(X, Y) = \underset{X_0, Y_0}{\text{Max}} \{ H(X, Y), G(X_0, Y) + G(X - X_0, Y), G(X, Y_0) + G(X, Y - Y_0) \} \tag{38}$$

where

$x_0 \leq \frac{1}{2}X$ and $Y_0 \leq \frac{1}{2}Y$,

$G(X, Y)$ is the maximum value that can be obtained from an X by Y rectangle using W_i by L_i rectangles at price shadow U_i and any succession of guillotine cuts, and

$H(X, Y) = \text{Max}_i\{U_i \mid W_i \leq X \text{ and } L_i \leq Y\}$.

Beasley (1985a) presented some computational improvements to the Gilmore–Gomory recursion given above and demonstrated that this procedure was capable of solving problems with as many as 50 sizes in under 2 seconds on a CDC 7600. The major problem with this approach is also its inability to limit the number of times a size appears in a pattern.

Christofides and Whitlock (1977) used a depth-first branch and bound algorithm to find optimal patterns for the general guillotine cut case as shown in Figure 1 (d) with limits on the number of times a size can appear in the pattern. Even though they took great pains to make the procedure as efficient as possible by eliminating duplicate cuts, the average time to generate a pattern with 20 ordered sizes was over 2 minutes on a CDC 7600 computer. This suggests that this procedure could be used to solve only small to medium sized cutting stock problems because the number of iterations required to find an optimal LP solution could easily exceed 2 or 3 times the number of ordered sizes.

Wang (1983) developed an alternative approach to generating general guillotine cutting patterns with limits on the number of times a size appears in a pattern. She combined rectangles in a horizontal and vertical build process as shown in Figure 2 where 0_i is an ordered rectangle of width W_i and length L_i.

She used an acceptable value for trim loss, B, rather than the shadow price of the ordered sizes to drive her procedure which is as follows:

Step 1. a. Choose a value for B the maximum acceptable trim waste.
b. Define $L^{(0)} = F^{(0)} = \{0_1, 0_2, \ldots, 0_n\}$, and set $K = 1$.

Step 2. a. Compute $F^{(K)}$ which is the set of all rectangles T satisfying

(i) T is formed by a horizontal or vertical build of two rectangles from $L^{(K-1)}$,

(ii) the amount of trim waste in T does not exceed B, and

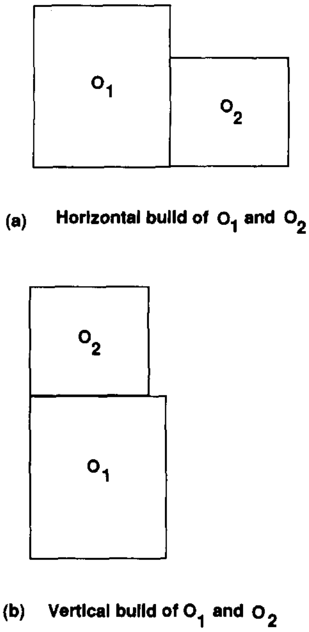

Figure 2.

(iii) those rectangles 0_i, appearing in T do not violate the constraints on the number of times a size can appear in a pattern.
b. Set $L^{(K)} = L^{(K-1)} \cup F^{(K)}$. Remove any equivalent (same component) rectangle patterns from $L^{(K)}$.

Step 3. If $F^{(K)}$ is non-empty, set $K = K + 1$ and go to Step 2. Otherwise go to Step 4.

Step 4. a. Set $M = K - 1$.
b. Choose the rectangle in $L^{(M)}$ which has the smallest total trim waste when placed in the stock rectangle.

Viswanathan and Bagchi (1988) and Vasko (1989) have both improved this algorithm by using better bounds to select the rectangles to be combined. Viswanathan and Bagchi used the unconstrained dynamic programming solution to the general guillotine problem in a best-first search algorithm which generates optimal solutions with significantly fewer nodes than required by Wang (1983) or Christofides and Whitlock (1977).

Vasko used the solution to the two-stage cutting pattern problem to find initial upper bounds on the general guillotine problem with restrictions on the number of times a size can appear in a pattern. This procedure is reported to be 25 times faster than Wang's for generating optimal solutions to the pattern generation problem.

Beasley (1985b) used Lagrangean relaxation of a zero–one integer formulation of the non-guillotine cut case shown in Figure 1 (e) as a bound in a tree search procedure to find optimal cutting patterns for the non-guillotine cut problem. Subgradient optimization and reduction tests make it possible to solve small problems with 10 sizes on a CDC 7600 in anywhere from 10 to 229 seconds.

Hadjiconstantinou and Christofides (1991) have developed a exact-tree search procedure for solving the non-guillotine cut problem with a constraint on the number of times a size can appear in the pattern. This procedure also uses subgradient optimization and reduction tests to make the algorithm more efficient. They report computational times for 2 problems with 15 ordered sizes of 3.2 and 65.2 seconds on a Cyber-855 computer. Optimal solutions to two other problems with 15 ordered sizes were not found in 800 seconds.

All the two-dimensional patterns considered to this point involve only orthogonal cuts. Rinnooy Kan, de Wit and Wijmenga (1987) demonstrated that under certain circumstances yield can be significantly improved if non-orthogonal cuts can be used. Figure 1 (f) shows an example where non-orthongonal cuts must be used because one-dimension of the ordered size is greater than both the width and length of the stock size. They derive relationships which indicate under which circumstances this will be beneficial based on the effective height and width of the tilted pieces.

Future directions

It is clear that moving from one to two dimensions causes significant difficulty in the pattern generating process. This is all the more alarming in light of the fact that only rectangular shapes were considered. It must also be noted that most of the two-dimensional papers referenced did not solve cutting stock problems as defined in (1)–(3). Except for Wang (1983), the primary focus was simply on pattern generation. Wang used her pattern generating procedure to solve a cutting stock problem. However, she did not use the Gilmore–Gomory delayed pattern generating technique but rather generated a large number of low trim patterns all at once and used a standard LP procedure to solve the cutting stock problem. This is the same approach which was made obsolete for one-dimensional problems in 1961 when Gilmore and Gomory published their first paper on cutting stock problems.

Taken together this suggests that there is much more research needed on procedures for solving two-dimensional cutting stock problems. An alternative worth considering, especially in those cases where there are many different ordered sizes with small order quantities, might be to first select a subset of orders to consider by solving a one-dimensional knapsack problem as in (13)–(15) based on area and then see if the resulting solution can be put together into a feasible two-dimensional pattern. Wang's algorithm seems to be ideal for this purpose inasmuch as the trim loss in the pattern would be known.

A candidate set of items to be included in the next pattern could be found by solving the following problem:

$$Z = \text{Max} \sum_i U_i A_i \tag{39}$$

$$\text{s.t.} \quad \sum_i \text{AR}_i A_i \le \text{UAR} \quad \text{for all } i, \tag{40}$$

$$A_i \le b_i, \tag{41}$$

$$A_i \ge 0 \quad \text{and integer} \tag{42}$$

where

AR_i is the area of ordered rectangle i, and

b_i is the upper limit on the number of times order i can be included in the pattern.

This candidate pattern $(A_1, \ldots, A_m)$ could then be tested for feasibility using Wang's procedure. If the AR_i are small, the chances are that there will be little trim loss in the candidate patterns generated. This may require that UAR be reduced to force some trim loss to make it more likely that feasible patterns are found.

References

Beasley, J.E. (1985a), "Algorithms for unconstrained two-dimensional guillotine cutting", *Journal of the Operational Research Society* 36, 297–306.

Beasley, J.E. (1985b), "An exact two-dimensional non-guillotine cutting tree search procedure", *Operations Research* 33, 49–64.

Beged Dov, A.G. (1970), "Some computational aspects of the *M* paper mills and *P* printers paper trim problem", *Journal of Business Administration* 1, 15–34.

Christofides, N., and Whitlock, C. (1977), "An algoritham for two-dimensional cutting problems", *Operational Research* 2530–44.

Dyckhoff, H. (1990), "A typology of cutting an packing problems", *European Journal of Operational Research* 44, 145–159.

Gilmore, P.C., and Gomory, R.E. (1961), "A linear programming approach to the cutting stock problem", *Operations Research* 9, 848–859.

Gilmore, P.C., and Gomory, R.E. (1963), "A linear programming approach to the cutting stock problem, Part II", *Operations Research* 11, 863–888.

Gilmore, P.C., and Gomory R.E. (1965) "Multistage cutting stock problems of two and more dimensions", *Operations Research* 13, 94–120.

Gilmore, P.C., and Gomory, R.E. (1966), "The theory and computation of knapsack functions", *Operations Research* 14, 1045–1074.

Hadjiconstantinou, E., and Christofides, N. (1991), "An optimal algorithm for general, orthogonal, 2-D cutting problems", Technical Report MS-91/2, Imperial College, London.

Haessler, R.W. (1971) "A heuristic programming solution to a nonlinear cutting stock problem", *Management Science* 17, 793–802.

Haessler, R.W. (1980), "A note on computational modifications to the Gilmore–Gomory cutting stock algorithm", *Operations Research* 28, 1001–1005.

Haessler, R.W. (1988), "A new generation of paper machine trim programs", *TAPPI Journal* 71 (August), 127–130.

Haessler, R.W., and Talbot, F.B. (1983), "A 0–1 model for solving the corrugator trim problem", *Management Science* 29 (December), 200–209.

Kantorovich, L.V. (1960), "Mathematical methods of organizing and planning production", reprinted in *Management Science* 6, 366–422.

Pierce, J.F. (1964), *Some Large Scale Production Problems in the Paper Industry*, Prentice-Hall, Englewood Cliffs, NY.

Rinnooy Kan, A.H.G., De Wit, J.R., and Wijmenga, R.T. (1987), "Nonorthogonal two-dimensional cutting patterns", *Management Science* 33, 670–684.

Sweeney, P.E., and Paternoster, E.R. (1991), "Cutting and packing problems: An updated literature review", Working Paper No. 654, University of Michigan, School of Business.

Sweeney, P.E., and Haessler, R.W. (1990), "One-dimensional cutting stock decisions for rolls with multiple quality grades", *European Journal of Operational Research* 44, 224–231.

Vasko, F.J. (1989), "A computational improvement to Wang's two-dimensional cutting stock algorithm", *Computers and Industrial Engineering* 16, 109–115.

Viswanathan, K.V. and A. Bagchi (1988), "An exact best-first search procedure for the constrained rectangular guillotine knapsack problem", *Proceedings of the American Association for Articicial Intelligence*, 145–149.

Wang, P.Y. (1983), "Two algorithms for constrained two-dimensional cutting stock problems", *Operations Research* 31, 573–586.