OPTIMAL CUTTING OF LUMBER AND PARTICLEBOARDS INTO DIMENSION PARTS: SOME ALGORITHMS AND SOLUTION PROCEDURES

Celso Carnieri[1]

Professor

Departamento de Matematica
Universidade Federal do Parana
Curitiba, Brazil

Guillermo A. Mendoza

Associate Professor

Department of Forestry
University of Illinois
Urbana, IL 61801

and

Luciano Gavinho

CalculoExato

Curitiba, Brazil

(Received August 1992)

ABSTRACT

This paper describes some algorithms and procedures that can be used for determining the optimal cutting of lumber or composite boards into dimension or furniture parts. Methodologies are described for various production scenarios: 1) cutting when the direction of the grain matters (e.g., lumber), 2) cutting composite boards where grain direction does not matter, 3) rip-first cutting, 4) crosscut-first cutting, and 5) a combination of rip-first and crosscut-first. An algorithm for optimizing the cutting of all lumber types while at the same time satisfying a given order of dimension parts is also described. The models can be used interactively for comprehensive optimization of cutting a mix of lumber as shown by the two-stage decision model, or the double knapsack algorithms could be used as stand alone models for optimizing the cutting of individual lumber.

Keywords: Dynamic programming, knapsack algorithm, optimal cutting patterns, dimension cutting.

INTRODUCTION

Faced with rising raw material costs and a dwindling resource base, furniture and dimension plants have become more concerned about their efficiency in cutting lumber into furniture parts. More than ever, mill managers must now deal with the dual problem of determining the right mix of lumber input to process and how this lumber should be cut to satisfy a cutting bill (i.e., customer order for various dimension parts or blanks). Mistakes arising from these two problems could result in significant amounts of waste due to misallocation of lum-

[1] This work was done while the first author was a visiting scientist at the Department of Forestry, University of Illinois. This study was partially funded by the CNPq—Brazilian National Council for Scientific and Technological Development, Contract No. 201097/91-5(NV), and the Northeastern Forest Experiment Station, Princeton, West Virginia.

Wood and Fiber Science, 26(1), 1994, pp. 131–141

ber, and ultimately to loss of profit or increase in lumber procurement costs.

The optimal cutting of lumber into dimension parts lends itself well to formal algorithms and analytical methods. The cutting of lumber, given its dimensions, into smaller parts of various sizes or dimensions is a well-defined problem which, at least intuitively, has an almost exact solution.

While the problem looks simple, the solution is puzzling because of the significant, if not infinite, number of ways to cut the lumber. The problem is further complicated by the need to satisfy the cutting bill and the availability of lumber with different sizes and grades.

The optimal cutting of lumber is too problematic and difficult to solve without the use of some numerical or analytical tool. It is obvious, however, that inappropriate cutting could result in significant amounts of waste because leftover lumber is either too short or too narrow.

Besides lumber, composite boards are also used for much furniture. Obviously, cutting composite boards is less complicated because they are generally homogenous or of uniform grade. Lumber, on the other hand, is of different quality or grade. It has defects, and the direction of the grain could affect the amount of waste as it is cut into smaller pieces. The direction of the grain also limits the flexibility of cutting. For instance, while a 3-in. × 12-in. piece is the same as a 12-in. × 3-in. piece when cut from particleboards, they are not the same pieces when cut from lumber because of the direction of the grain.

Technological advances in flexible manufacturing systems offer an excellent environment in which optimal cutting of lumber could be implemented and potentially automated. For instance, implementing optimal lumber cutting strategies or patterns may require a number of machine setups or adjustments. If such adjustments are determined off-line (e.g., *a priori*), it is conceivable that the optimal cutting strategy could help drive these adjustments—that is, analytical solutions that were determined off-line could be implemented using computer-driven control techniques under a flexible manufacturing environment.

The objective of this paper is to describe some algorithms and heuristic procedures that can be used to determine analytically the optimal cutting of lumber or particleboard into dimension parts. The models described, unlike other previous models, offer exact solution procedures yielding cutting patterns with specific dimensions and the exact locations of the parts within the lumber or particleboard.

LUMBER CUTTING MODELS

Published literature contains a number of methods developed over the last decade addressing various forms of lumber cutting and allocation problems. One of the first methods includes the use of nomograms and yield tables indicating the relative amounts (in percent) of dimension parts that could be recovered from a given grade and species of lumber (Shumann 1971, 1972; Gilmore et al. 1984). These amounts are adjusted manually through a systematic trial and error procedure to satisfy the demand or cutting bill.

The advent of computers and faster computing capabilities in the early 80s paved the way for the use of computer-based models. These models are developed following the procedures and methods of simulation, optimization, and mathematical programming techniques (Brunner 1984; Martens and Nevel 1985; Giese and McDonald 1982; Giese and Danielson 1983).

The models and procedures described in this paper cover a range of production environments and scenarios. Some models are formulated purposely as generic solution algorithms, which could be adopted and modified to suit a specific lumber production scenario. Other procedures, particularly heuristic methods, are also described as techniques to help improve the solution algorithms and their computational requirements to make them more practical and implementable. These procedures are significant because they could facilitate the transfer of analytical solutions into actual and practicable operations.

FORMULATION OF CUTTING PROBLEMS

The generic lumber cutting problem broadly described above can be formulated as a mathematical programming problem. From this generic formulation, alternative models will be described addressing specific production scenarios commonly faced by furniture and dimension plants. The generic lumber cutting problem can be formulated as follows:

P1

$$\text{Min TC} = \sum_k \sum_j C_k X_j k \qquad (1)$$

$$\text{s.t.} \quad \sum_k \sum_j a_{ij}{}^k X_j{}^k \geq d_i; \quad i = 1, \ldots, m \qquad (2)$$

$$\sum_j X_j{}^k \leq N_k \qquad k = 1, \ldots, K \qquad (3)$$

$$X_j{}^k \geq 0$$

where:

TC = total production cost
C_k = cost associated with lumber type k
$X_j{}^k$ = the number of lumber pieces type k cut following cutting pattern j
d_i = amount ordered for dimension part i
$a_{ij}{}^k$ = the number of dimension parts i obtained from one lumber type k following cutting pattern j
N_k = the amount of lumber type k available in stock.

The objective function in (1) minimizes the total cost of the mix of lumber input required to satisfy the demand or cutting bill. Constraint (2) is the demand or cutting bill for various dimension parts. Constraint (3) describes the amount of lumber types with different sizes and grades, available in stock (i.e., inventory).

Case I: Direction of grain matters

Most wooden furniture is made of parts cut from solid lumber. In general, lumber, particularly hardwood, has dominant grain features. Hence, the direction of the grain is significant in cutting lumber into dimension parts of various sizes. Clearly, in this situation, a 2-in. × 12-in. (i.e., width × length) board is not the same as a 12-in. × 2-in. because of the direction of the grain.

The general methodology in solving the lumber cutting problem proposed in this study follows the two-dimensional knapsack algorithm first developed by Gilmore and Gomory (1965). This technique is used to determine the optimal cutting pattern of one piece of lumber (given its size) into combinations of smaller dimension parts. However, there is a cutting bill to be met. Hence, the cutting of each piece of lumber should not be optimized independently but comprehensively, considering the availability of lumber and total economic return rather than returns from individual pieces. In other words, the problem really has two stages. The first-stage problem optimizes the allocation of lumber for cutting following the optimal cutting patterns identified by solving the second-stage problem.

The problem in Stage II is the generation of cutting patterns. Conceivably, there may be a significant, if not infinite, number of possible cutting patterns. Instead of trying to generate all possible cutting patterns, only those "potentially good" cutting patterns are generated. The solution generated from Stage I establishes the desirability or suitability of the generated cutting patterns considering all lumber types, their availability, and the demand for various dimension parts.

An algorithm for solving the two-stage problem was also developed by Gilmore and Gomory (1965). Applications of this procedure are described by Mendoza and Bare (1986) and Foronda and Carino (1991). The algorithm consists of solving two problems, independently and interactively. The first problem, called main problem, corresponds to the first-stage decision problem, while the second-stage problem corresponds to a subproblem dealing with the cutting of lumber type k.

The interface between the main problem and the subproblems is the key to solving the optimal lumber cutting problem comprehensively. This link provides the simultaneous optimization of cutting individual pieces of lumber,

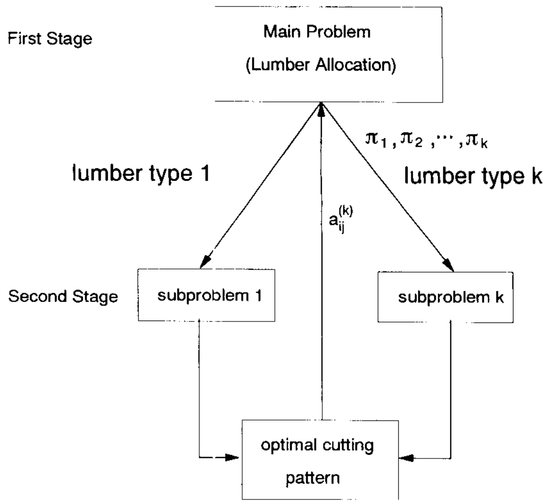

FIG. 1. The two-stage decision process.

and the allocation of lumber into optimal cutting patterns. The interface is made through Shadow Prices as described in Fig. 1. The Shadow Prices (Π_i) are the Lagrange or Simplex Multipliers associated with the constraints in Eq. (2). Hence, during the interactive solution procedure (Mendoza and Bare 1986), the Π_i's are generated after solving **P1** in the main problem, which then becomes the input to the subproblems in **P2**. The solution of each subproblem yields the optimal cutting of each lumber type, which then becomes the input to the main problem in **P1**.

Each subproblem is solved as a two-dimensional or double knapsack problem. Intuitively, this corresponds to cutting the lumber in two steps: first lengthwise (along the length), and then crosswise (across the length). The first knapsack problem optimizes cutting based on the length of the dimension parts. The second knapsack problem optimizes the cutting based on the width of the dimension parts.

The first knapsack problem can be described as follows: For all widths w_i, calculate Π_i^*, the optimum value obtainable by fitting rectangles $w_j \times l_j$ (where $w_j \leq w_i$), end to end, into a strip of width w_i and length L_k. This problem is formally formulated as:

P2

$$\text{Max}\Pi_i^* = \Pi_1 a_1 + \Pi_2 a_2 + \ldots + \Pi_i a_i \quad (4)$$

$$\text{s.t.} \quad l_1 a_1 + l_2 a_2 + \ldots + l_i a_i \leq L_k \quad (5)$$

$$a_j \geq 0, \text{ integer}$$

where:

Π_j = shadow price associated to the *jth* constraint in Eq. (2); j = 1, 2, . . . , i

a_j = number of rectangles type j that appear in strip i.

The above formulation requires that the blanks are indexed and arranged such that $w_i \leq w_{i+1}$; i = 1, . . . , m − 1.

The second knapsack problem, which optimizes the cutting along the length, can be described as follows;

P3

$$\text{Max } M_k = \Pi_1^* b_1 + \ldots + \Pi_m^* b_m \quad (6)$$

$$\text{s.t.} \quad w_1 b_1 + w_2 b_2 + \ldots + w_m b_m \leq W_k \quad (7)$$

$$b_i \leq 0; \text{ integer}$$

where:

M_k = the "best" value for lumber type k

Π_i^* = value obtained from the first knapsack problem corresponding to various widths

b_i = number of strips of width w_i

W_k = width of lumber type k.

M_k is the information needed for comparison among all lumber types to determine which cutting pattern will be included in **P1**. The comparison is made by choosing the lumber type with the minimum ($C_k - M_k$).

Solving for M_k is necessary only when cutting a mix of lumber of different sizes. This is necessary to determine the optimal cutting pattern for each lumber type given the respective dimensions or sizes. If cutting of individual lumber pieces is to be optimized independently, then **P2** and **P3** are solved only once yielding one cutting pattern. Moreover, the Π_i's are simply the price or value of each dimension part, not the Shadow Price or Simplex Multiplier associated to the constraint in Eq. (2).

Hence, the Π_i's and M_k are relevant only when the cutting scenario to be optimized involves a mix of lumber with different sizes.

For each subproblem corresponding to lumber type k, the two knapsack problems are solved using Dynamic Programming and the recursive calculation described below;

$$\begin{aligned} &\text{Max } F_s(x) = \\ &= \text{Max}\{\Pi_s + F_s(x - l_s), F_{s-1}(x)\} \\ &\qquad \text{for } s > 1 \end{aligned} \tag{8}$$

Following conventional Dynamic Programming, $F_s(x)$ is the value of the best combination that can be fitted into a board (i.e., lumber or particleboard) of length (or width) × using only the first s variables. The variable s is the index variable traditionally used in multistage Dynamic Programming.

Gilmore and Gomory (1965) have shown that the recursive Dynamic Programming problem described above has significant computational advantages compared to other knapsack solution algorithms. These computational advantages stem from the *a priori* ordering of the sizes, and the fact that all the Π_i's need to be calculated only once. The n knapsack problems are solved using only one "lookup table" generated by solving Eq. (8).

First cut: Rip or crosscut

The double knapsack model described in **P2** and **P3** corresponds to a production scenario in which the first cut is made along the length (i.e., rip-first) followed by a crosscut. However, it is conceivable that efficiency could be improved if the first cut is a crosscut followed by a cut along the length. It may also be possible to have some lumber cut following the rip-first then crosscut, while others are crosscut first.

The solution procedure for the crosscut-first problem is the same as the rip-first procedure described previously. Conceptually, the procedure starts by rotating the lumber and dimension parts (i.e., the widths and lengths of all the lumber and dimension parts are swapped). Then, a two-dimensional or double knapsack algorithm like **P2** and **P3** is solved.

If the production setup allows for either method to be used for some lumber, then the solution procedure that determines which is the best first cut will simply involve comparison between the results of the two solutions (i.e., rip-first or crosscut-first). The numerical example presented in the succeeding section demonstrates a situation in which the crosscut-first yielded higher value recovery than the rip-first strategy.

Currently, rip-first seems to be the most convenient considering sizes of lumber and the layout of most dimension plants. It is, however, conceivable that as scarcity of raw materials become even more acute, making conversion efficiency even more significant, and as flexible manufacturing systems find their way to dimension and furniture manufacturing, the problems of rip-first or crosscut-first (or a combination of both) will become more relevant.

Case II. Direction of grain does not matter

Some wooden furniture or cabinets are made from various types of composite or particleboards. Unlike lumber, particleboards are generally homogeneous and do not have any dominant grain. Hence, a 3-in. × 12-in. board could be cut lengthwise or crosswise. In other words, a 3-in. × 12-in. board is the same as a 12-in. × 3-in. board.

In this case, a blank or dimension part can be turned and cut from the board if this is a better alternative. Hence, for the blank *i* in this position, the width is l_i and the length is w_i. The $a_{ij}^{(k)}$ in the model as described in **P1** is the number of *all* blanks that are cut using cutting pattern *j*, both in the original position and after it is turned. In other words, the formulation expands into 2m variables in the knapsack problems. The main problem **P1** remains to have *m* constraints corresponding to the order of dimension parts.

Model modifications.—Each ordered dimension part will now have two variables in the knapsack problem corresponding to *2 widths* and *2 lengths.* The 2*m* widths are again ordered and indexed such that $w_i \leq w_{i+1}$. **P1**

TABLE 1. *Ordered blanks and their prices.*

Blanks	Π
2.5 × 3	5
3 × 4	7
3 × 2.5	5
4 × 5	12
4 × 3	7
5 × 4	12

TABLE 2. *Optimal values of $F_s(x)$ for P2.*

	S					
x[a]	1	2	3	4	5	6
2.5	0	0	5^3	5^3	5^3	5^3
3	5	5^1	5^3	5^3	7^5	7^5
4	5	7^2	7^2	7^2	7^5	12^6
4.5	5	7^2	7^2	7^2	7^5	12^6
5	5	7^2	10^3	12^4	12^4	12^6
5.5	5	7^2	10^3	12^4	12^5	12^6
6	10	10^1	10^3	12^4	14^5	14^5
7	10	12^2	12^3	12^4	14^5	19^6
8	10	14^2	15^3	17^4	19^5	24^6
8.5	10	14^2	15^3	17^4	19^5	24^6
11	15	19^2	20^3	24^4	26^5	31^6
	$\frac{15}{\Pi_1^*}$	$\frac{19}{\Pi_2^*}$	$\frac{20}{\Pi_3^*}$	$\frac{24}{\Pi_4^*}$	$\frac{26}{\Pi_5^*}$	$\frac{31}{\Pi_6^*}$

[a] These are the only "useful states" in the system that need to be examined. Not all of the states between 2.5 and 11 need to be generated and evaluated. Carnieri et al. (1991) describes a procedure for determining the set of useful numbers or states for the type of dynamic programming problem addressed in this paper.

stays the same, except the a_{ij}^k for a cutting pattern *j* and a blank *i* now represents the sum of the number of times the blank ordered *i* appears in the position $w_i \times l_i$ and the number of times it appears in the position $l_i \times w_i$. The knapsack formulations need three slight modifications: 1) the Shadow Price Π_i from **P1** is now associated with both $w_i \times l_i$ and $l_i \times w_i$; 2) the formulation in **P2** stays the same, but now there are 2m problems to solve; and 3) the formulation in **P3** also has *2m* variables, where a variable b_i represents the number of times a strip of width w_i appears in the cutting pattern (i.e., w_i is chosen from the ordered $w_i \leq w_{i+1}$, including the position based on width and the original position based on length).

NUMERICAL EXAMPLE

A numerical example is presented in this section to demonstrate the algorithms described in the previous sections. The more general case in which grain does not matter is illustrated with both rip-first and crosscut-first scenarios. To simplify the sample problem, only the cutting of one lumber type (i.e., Stage II problem) is described. An application is also described later to demonstrate results from an actual implementation of the algorithms.

The problem involves cutting an individual piece of lumber whose width and length are 9.7 and 11 (units are disregarded without any loss of generality), respectively. Three types of blanks must be cut with the following dimensions and values.

Blank	3 × 4	2.5 × 3	4 × 5
Π	7	5	12

Since only one lumber type is to be cut, the Π_i's correspond to the actual prices or values of the blanks. Again, if this lumber is one in a mix of lumber types, the Π_i's could be the associated Shadow Price obtained in Stage I.

Following the model modifications when grain does not matter, the blanks are turned, generating six instead of three blanks, as follows:

Blanks	Π	
3 × 4	7	
2.5 × 3	5	original position
4 × 5	12	
4 × 3	7	
3 × 2.5	5	rotated position
5 × 4	12	

Before solving **P2**, the blanks are rearranged in increasing order as described in Table 1.

Solving **P2** using Dynamic Programming and the recursive relationship described in Eq. (8), Table 2 is generated.

Table 2 completes the solution of **P2**. The second knapsack problem in **P3** can now be solved. Again, the recursive relationship in Eq. (8) is used given the optimal values of Π_i*. The result is summarized in Table 3.

TABLE 3. *Optimal values of $F_s(x)$* for P3.*

x	s = 1	2	3	4
2.5	15	15^1	15^1	15^1
2.7	15	15^1	15^1	15^1
3	15	20^2	20^2	20^2
3.2	15	20^2	20^2	20^2
3.7	15	20^2	20^2	20^2
4	15	20^2	26^3	26^3
4.2	15	20^2	26^3	26^3
4.7	15	20^2	26^3	26^2
5	30	30^1	30^1	31^4
5.7	30	35^2	35^2	35^2
6.7	30	40^2	41^3	41^3
7.2	30	40^2	46^3	46^3
9.7	45	60^2	61^3	61^3

* The superscripts are included and are used for backtracking purposes.

The optimal solution can now be obtained from Table 3. The maximum total value of all blanks cut is 61. Following the rip-first cutting scenario, the first strip cut is a strip whose width is the 3rd^2 width (i.e., 61^3) in Table 1 (width is 4). After the first strip, the remaining width is 5.7 (i.e., 9.7 − 4). From Table 3, the optimal solution is 35^2; hence the next strip has a width equal to the 2nd width in Table 1 (i.e., width is 3). This brings the remaining width to 2.7 (i.e., 5.7 − 3). From Table 3, the optimal solution along x = 2.7 is 15^1. Hence, the last strip has a width of 2.5, which is the first width in Table 1.

The optimal length of the blanks can now be determined. Along the first strip cut with a width of 4, we can refer to Table 2. Since the width is 4 inches (which corresponds to the 5th position in Table 1), we can look at Table 2 for s = 5. The optimal solution is 26^5. Hence the length corresponds to the 5th blank with length equal to 3. This brings the remaining length to 8. Examining Table 2, x = 8, s = 5, the solution is again to have the blank whose length is 3. This backtracking can be continued to get the optimal solution, which is shown in Fig. 2.

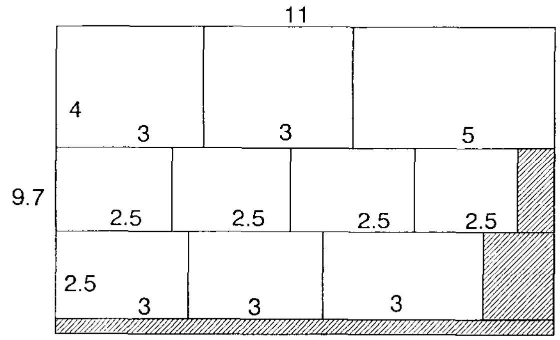

FIG. 2. Optimal cutting pattern (rip-first).

Consider now the case when the first cut is a crosscut. In this situation, the parent stock is turned so that the length becomes the width and vice versa. Using the same data, and following the same procedure as in the rip-first scenario, the results obtained for **P2** and **P3** are summarized in Tables 4 and 5.

TABLE 4. *Optimal values of $F_s(x)$ for P2 (crosscut-first).*

x	s = 1	2	3	4	5	6
2.5	0	0	5^3	5^3	5^3	5^3
2.7	0	0	5^3	5^3	5^3	5^3
3	5	5^1	5^3	5^3	7^5	7^5
3.2	5	5^1	5^3	5^3	7^5	7^5
3.7	5	5^1	5^3	5^3	7^5	7^5
4	5	7^2	7^2	7^2	7^5	12^6
4.2	5	7^2	7^2	7^2	7^5	12^6
4.7	5	7^2	7^2	7^2	7^5	12^6
5	5	7^2	10^3	12^4	12^4	12^6
5.7	5	7^2	10^3	12^4	12^5	12^6
6.7	10	10^1	10^3	12^4	14^5	17^6
7.2	10	12^2	12^3	12^4	14^5	19^6
9.7	15	15^1	17^3	19^4	21^5	24^6

TABLE 5. *Optimal values of $F_s(x)$ for P3 (crosscut-first).*

x	s = 1	2	3	4
2.5	15^1	15^1	15^1	15^1
3	15^1	17^1	17^1	17^1
3.5	15^1	17^1	17^1	17^1
4	15^1	17^1	21^3	21^3
4.5	15^1	17^1	21^3	21^3
5	30^1	30^1	30^1	30^1
5.5	30^1	32^2	32^1	32^1
6	30^1	34^2	34^2	34^1
7	30^1	34^2	38^3	38^3
8	45^1	47^2	47^2	51^4
8.5	45^1	49^2	49^2	51^4
11	60^1	64^2	64^2	64^2

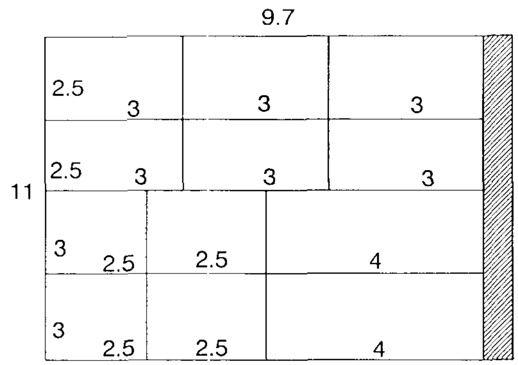

FIG. 3. Optimal cutting pattern (crosscut-first).

The optimal cutting pattern is described in Fig. 3. The optimal cutting pattern for the crosscut-first cutting scenario yielded a maximum total value of 64, which is higher than the value recovery of the rip-first scenario, which is equal to 61.

AN APPLICATION

A software system has been developed for the algorithms described in this paper and has also been applied in actual cutting operations in Brazil. One particular application involves the cutting of parent stock with dimensions 2,000 × 2,000 (millimeters). In this application, the grain does not matter (e.g., particleboards were cut). Five sizes of blanks were ordered. The sizes of the blanks and the quantity ordered are described below:

Blank	Size (mm)	Demand (order)
1	460 × 363	150
2	363 × 135	150
3	1,170 × 400	50
4	425 × 345	150
5	460 × 345	55

For this problem, four optimal cutting patterns were generated as shown in Figs. 4–7. (Figures are not drawn according to scale). The

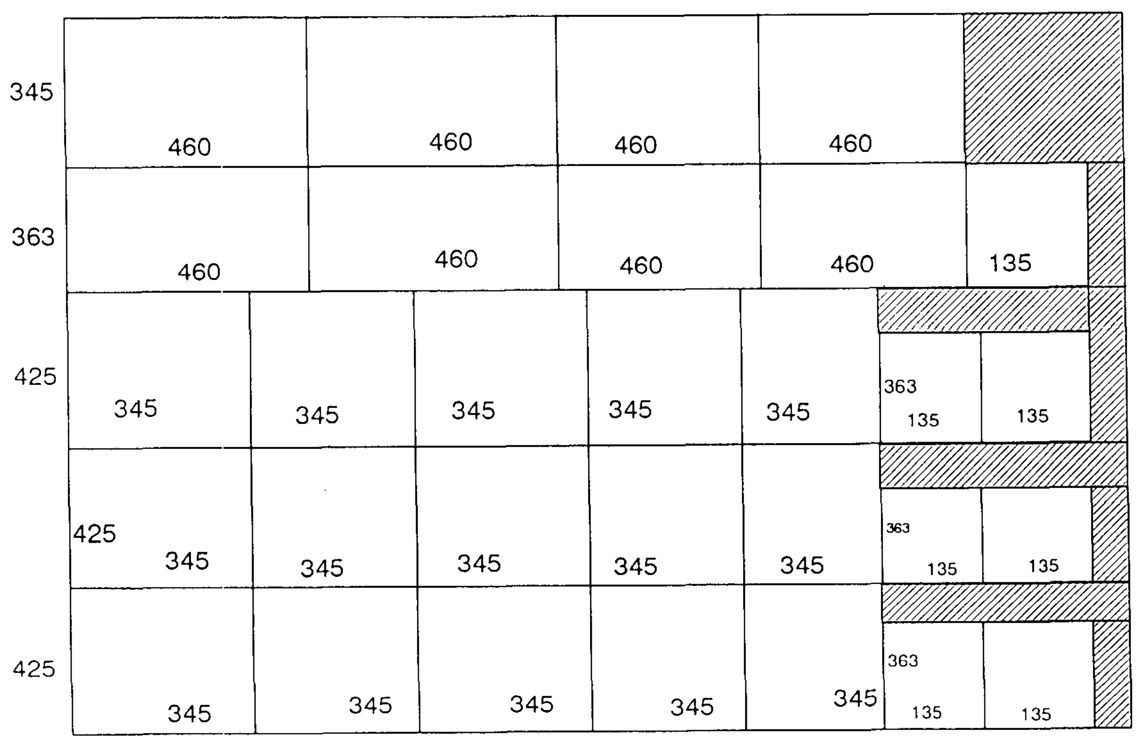

FIG. 4. Cutting pattern 1 (96% recovery).

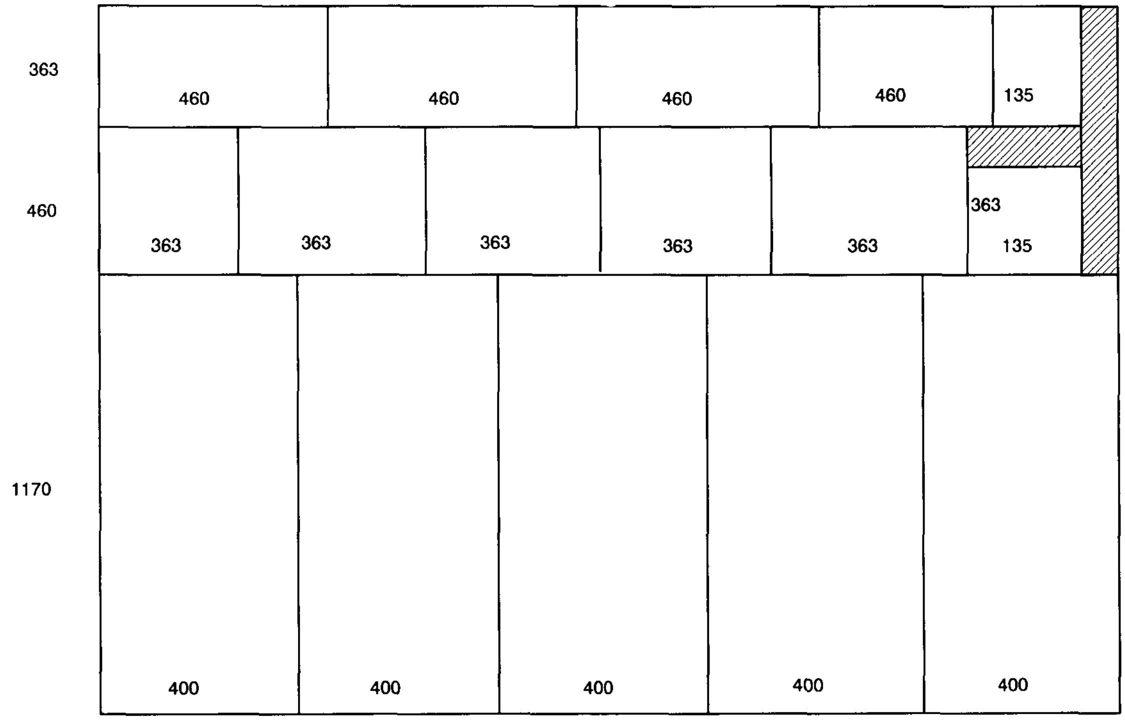

FIG. 5. Cutting pattern 2 (98.5% recovery).

solution generated also indicates that a total of 22 particleboards must be cut to satisfy the order of blanks. The number of boards to be cut following cutting patterns 1, 2, 3, and 4, are 3, 10, 6, and 3, respectively. The actual production schedule showing the number of boards cut for each of the four cutting patterns and the total number of parts obtained are shown in Table 6. The fact that grain does not matter is highlighted in cutting pattern 2 where the blanks 363 × 460 were cut both lengthwise and crosswise.

SUMMARY AND CONCLUSIONS

Faced with the dwindling source of quality raw materials, dimension plants will now have to address more carefully the issue of optimal cutting, not only to maximize recovery, but more importantly, to get the most economic return from their lumber while consistently satisfying orders from their customers. This problem has received little attention in the past; but as competition for raw materials intensifies, and as advances in flexible manufacturing

TABLE 6. *Production schedule in terms of the number of blanks or parts obtained from using the four cutting patterns.*

		Cutting patterns								
		1(3)[a]		2(10)		3(6)		4(3)		
Blanks	Size	Per board	Total	Per board	Total	Per board	Total	Per board	Total	Total
1	460 × 363	4	12	9	90	2	12	12	36	150
2	363 × 135	7	21	2	20	15	90	7	21	152
3	1,170 × 400	—	—	5	50	—	—	—	—	50
4	425 × 345	15	45	—	—	15	90	5	15	150
5	460 × 345	4	12	—	—	6	36	5	15	63

[a] The numbers within the parentheses are the number of boards cut.

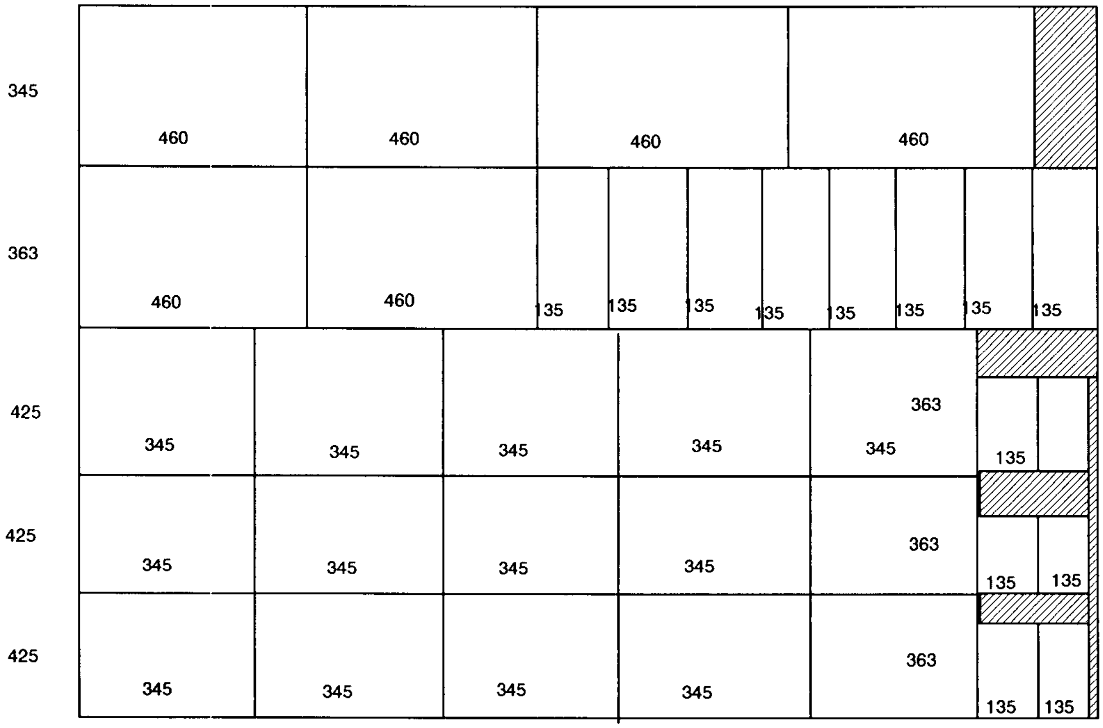

Fig. 6. Cutting pattern 3 (96.3% recovery).

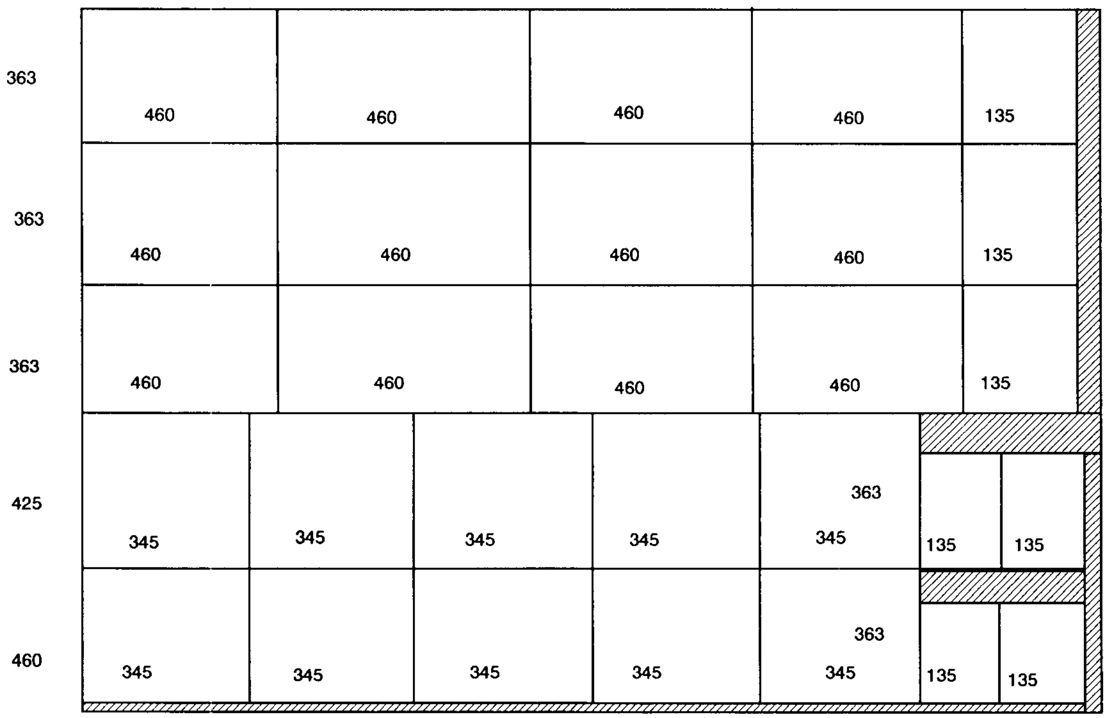

Fig. 7. Cutting pattern 4 (96.8% recovery).

technology take their place in the furniture industry, dimension plants will need to position themselves competitively through the adoption of more efficient processing techniques. The procedures and algorithms described in this paper could provide the building blocks that can enhance the capability of dimension plants to achieve better efficiency in cutting furniture or dimension parts.

The models described in this paper offer exact-dimension cutting solutions. Hence, they can perform well in cutting particleboards with fixed or uniform sizes. For cutting lumber, obviously the algorithms will work best if the lumber is sorted into groups that are of approximately uniform sizes. From a practical standpoint, the results obtained from the algorithms can be most beneficial in situations in which lumber is sawn by batches so that sawing can be set up following the cutting patterns recommended by the algorithm. It should be noted, however, that the double knapsack algorithm can be used to optimally cut individual lumber. Hence, if lumber is cut at random, the double knapsack algorithm can be used to determine the optimal cutting pattern of each piece.

The algorithms described in this paper are somewhat generic. However, some heuristic procedures are also described showing modifications from the generic models to suit various production scenarios. Various algorithms are described for different cutting strategies, namely: 1) rip-first, 2) crosscut-first, or 3) a combination of both. Two algorithms are also developed for: 1) cutting lumber, and 2) cutting composite boards.

REFERENCES

BRUNNER III, C. C. 1984. A computer program to determine furniture cutting yields for both rip-first and crosscut-first sawing sequences. Ph.D. thesis, Virginia Polytechnique Institute and State University, Blacksburg, VA.

CARNIERI, C., G. A. MENDOZA, AND L. GAVINHO. 1991. Modified knapsack algorithms for cutting lumber into furniture parts. Working paper No. 2. Department of Forestry, University of Illinois, Urbana, IL.

FORONDA, S. U., AND H. F. CARINO. 1991. A heuristic approach to the lumber allocation and manufacturing in hardwood dimension and furniture manufacturing. European J. Operations Res. 54:151–162.

GIESE, P. J., AND K. A. MCDONALD. 1982. OPTYLD: A multiple rip-first computer program to maximize cutting yields. USDA Forest Service Res. Pap. FPL-412.

———, AND J. D. DANIELSON. 1983. CROMAX—A crosscut-first computer simulation program to determine cutting yields. USDA Forest Service General Tech. Rep. FPL-38.

GILMORE, P. C., AND R. E. GOMORY. 1965. Multi-stage cutting stock problems of two or more dimensions. Operations Research 13:94–120.

GILMORE, R. C., S. J. HANOVER, AND J. D. DANIELSON. 1984. Dimension yields from yellow poplar lumber. USDA Forest Service General Tech. Rep. FPL-41.

MARTENS, D. G., AND R. L. NEVEL. 1985. OPTIGRAMI: Optimum lumber grade mix program for hardwood dimension parts. USDA Forest Service Res. Paper NE-563.

MENDOZA, G. A., AND B. BARE. 1986. A two-stage decision model for log bucking and allocation. Forest Prod. J. 10:70–74.

SHUMANN, D. R. 1971. Dimension yields from black walnut lumber. USDA Forest Service Res. Paper FPL-162.

———. 1972. Dimension yield from alder lumber. USDA Forest Service Res. Paper FPL-170.