A two-stage decision model for log bucking and allocation

Guillermo A. Mendoza
B. Bruce Bare

Abstract

To ensure efficient and economic utilization of trees into finished products, log bucking and allocation decisions should be integrated and optimized simultaneously. This paper describes a conceptual approach for solving this problem by formulating the integrated wood resources allocation problem as a two-stage decision process. The first stage decision involves determining an optimal allocation policy for the logs and other products. The second stage decision consists of generating alternative log bucking policies. Models developed for each stage are interfaced using a column generation technique which closely parallels the Dantzig-Wolfe decomposition algorithm.

One of the major problems confronting the manager of an integrated forest products firm is how to determine a production policy that insures the efficient utilization of available wood resources. In order to achieve this, the manager must understand the different production processes and activities involved, from log production to the manufacture and sale of finished products.

Log bucking determines the "best" way of bucking trees into logs of different sizes. This bucking process essentially determines what portion of a tree can be allocated to specific forest products, such as lumber, veneer, or pulp. It also greatly influences the sizes and quality of the end products. These factors reflect the importance of log bucking and its significant effect on the profitability and economic utilization of a tree.

Log allocation, on the other hand, has been viewed traditionally as a problem of balancing the demand for forest products with the available supply of logs. A closer examination of the problem, however, indicates that it actually involves two related policies, 1) an allocation policy that distributes the logs to the different processing plants, and 2) a production policy for each plant.

When these two major activities (log bucking and allocation) are viewed from the standpoint of integrated utilization, it is clear that they are interdependent and affect each other in a direct way. As such, they should be examined and analyzed as two functionally related components of wood utilization. In other words, wood utilization should be viewed as a multistage decision process where the bucking of trees into logs is the first of a series of related decisions. Following bucking is the allocation of the logs to the processing plants, and the subsequent manufacture and distribution of the finished products. All these processes or activities should be simultaneously considered in order to achieve the most efficient and economic utilization of wood resources.

A few studies have attempted to examine this problem. One of the earliest attempts to integrate log bucking and allocation was reported by Pnevmaticos (8). In that study a dynamic programming model to optimize log bucking was developed. The optimal bucking policy generated by the dynamic programming model was used to determine the optimal log allocation policy through linear programming. One of the conclusions was that the solution generated may not be optimal because only one bucking policy was used. Moreover, the interface between log bucking and allocation was done by incorporating the optimal log bucking policies as technological coefficients of the allocation model formulated as a linear program. As such, the interface between log bucking and allocation operates only in one direction. That is, the optimal bucking policy dictates

The authors are, respectively, Assistant Professor, Dept. of Forestry, Univ. of Illinois, 110 Mumford Hall, Urbana, IL 61801; and Professor, College of Forest Resources AR-10, Univ. of Washington, Seattle, WA 98195. This paper was received for publication in November 1985.

Forest Prod. J. 36(10):70-74.

the amount of certain types of logs to be allocated, but the model does not permit the option to change the bucking policy for some trees even if the allocation policy indicates that a change may yield better results.

McPhalen (6) developed a linear programming model based on a column generating technique (5) that permits the coordination of log bucking and log sawing subject to market demand constraints for lumber. This linear programming model was structured to maximize lumber revenue by considering alternative bucking policies for trees with constraints on the desired quantity of different lumber sizes. New log bucking patterns, and hence new quantities of lumber sizes to be included in the linear programming model, are generated through a dynamic programming model that determines the optimal quantity of lumber for each log.

Faaland and Briggs (4) developed a dynamic programming model that can simultaneously optimize log bucking, sawing of logs into live-sawn lumber, and edging lumber into finished dimensions. Their model allows for variation in tree shape and quality. It integrates a log bucking model (2) and a sawing model (9). The model can evaluate the effects of changes in sawkerf, lumber thickness, and tree shape on the optimal conversion of finished lumber.

This paper defines integrated wood utilization that may include not only log bucking and sawmilling (4, 6) but other processes like veneer and plywood manufacture. Moreover, the determination of bucking policies is solved iteratively with log allocation instead of by independent and sequential solutions such as the approach described by Pnevmaticos (8).

Model structure and formulation

The general approach of the integrated wood resources allocation model described in this paper follows the decomposition principle (1, 3). This approach allows for the formulation of a linear programming problem (usually with a large number of decision variables) as a two-stage decision problem where two interrelated and functionally dependent problems are solved at each stage.

The first stage decision problem is a wood resources allocation model (WRAM) which includes log allocation and the production and sale of finished products. The second stage decision problem is concerned with determining the "best" log bucking policy. Models formulated for each problem are interfaced using the structure of the two-level optimization process of the decomposition principle.

Before presenting the mathematical formulation of the model, the following points should be noted:

1. Trees are grouped into stem classes based upon tree length. Classification of stem classes can also be based on diameter sizes, qualitative criteria, or species;

2. Logs, like stem classes, are also classified mainly on the basis of length. However, the model does not preclude the inclusion of other quantitative and qualitative bases for categorizing or grading logs;

3. A bucking policy is defined in terms of the number of logs of different lengths produced. For example, consider a stem class whose length is 90 feet. Assume that there are four log classes that can be produced out of this stem class with the following lengths: 8, 10, 12, and 14 feet. Also assume that a 0.25-foot trimming allowance is provided. Two alternative bucking policies for this situation are described in Table 1.

The primary indicator or basis for classifying trees and logs is length. However, diameter and other qualitative criteria may also be considered in the classification so that tree or log classes can be a combination of different length and diameter sizes. Mendoza (7) adopted both length and diameter for classifying tree and log classes.

First stage decision problem (WRAM)

The first stage decision problem is formulated as follows:

$$Max\ R = \underbrace{\sum_{m=1}^{M_p} \sum_{p=1}^{P} P_{mp} Z_{mp}}_{\text{Returns from finished products}} - \underbrace{\sum_{k=1}^{K} \sum_{p=1}^{P} t_{kp} Y_{kp}}_{\text{Processing cost}} - \underbrace{\sum_{i=1}^{I} \sum_{j=1}^{J_i} C_{ij} X_{ij}}_{\text{Cost due to bucking waste}} \quad [1]$$

subject to:

$$\sum_{i=1}^{I} \sum_{j=1}^{J_i} a_{ijk} X_{ij} - \sum_{p=1}^{P} Y_{kp} = O, k = 1, 2, ..., K \quad [2]$$

$$\sum_{k=1}^{K} b_{mkp} Y_{kp} - Z_{mp} = O, \quad m = 1, 2, ..., M_p \quad p = 1, 2, ..., P \quad [3]$$

$$D_{mp} \leqq Z_{mp} \leqq Q_{mp} \quad m = 1, 2, ..., M_p \quad p = 1, 2, ..., P \quad [4]$$

$$\sum_{j=1}^{J_i} X_{ij} \leqq N_i \quad i = 1, 2, ..., I \quad [5]$$

$$X_{ij} \geqq O; i = 1, 2, ..., I; \quad j = 1, 2, ..., J$$
$$Y_{kp} \geqq O; k = 1, 2, ..., K; \quad p = 1, 2, ..., P$$
$$Z_{mp} \geqq O; m = 1, 2, ..., M_p; \quad p = 1, 2, ..., P$$

where:

R = measure of economic return

C_{ij} = cost of waste per unit of stem class i using bucking policy j

X_{ij} = number of trees in stem class i cut using bucking policy j

Y_{kp} = number of log type k's allocated to processing plant p (e.g., sawmill, veneer mill, pulpmill)

Z_{mp} = total amount of product m produced by plant p from all log classes

t_{kp} = cost of processing log type k when processed by plant p

P_{mp} = return (price) of product m processed by plant p

a_{ijk} = number of log type k's produced by cutting stem class i using bucking policy j

b_{mkp} = amount of product m (e.g., average conversion rates) produced from log type k processed by plant p

D_{mp}, Q_{mp} = minimum and maximum amounts of product m that should be produced by processing plant p

N_i = maximum number of trees available in stem class i

M_p = number of finished product types (or grades) produced from processing plant p

I = number of stem classes

K = number of log types or classes

J_i = number of bucking policies for stem class i

In this formulation, the assumed objective of management is the maximization of the net economic returns from the production and sale of wood products as formulated in Equation [1]. Equation [2] is a constraint defining material balance in log allocation. That is, the total amount of logs produced from stem bucking should be equal to the total amount of logs allocated to the processing plants. Equation [3] is also a material balance equation defining the amount of final products (e.g., lumber) obtained from intermediate products (e.g., logs). The constraint in Equation [4] describes the production limits for the different products, which may be due to contractual obligations from customers, or production capabilities of the processing plants. Finally, Equation [5] is a restriction concerning the available number of trees in each stem class.

Note that in this formulation, log market, either for purchasing or selling, was not included. However, if necessary or appropriate, depending on the scope of the problem addressed, the log market can easily be incorporated in the objective function and constraints. Hence, log market considerations such as the price of the log, will be included in the objective function; and log market restrictions such as contractual obligations to buy or sell a certain amount of logs will also be included in the constraints.

Second stage decision problem

The second stage decision problem involves determining the "best" bucking policy for each stem class. In reference to the formulation of WRAM, the stem bucking problem may be stated as: What set of a_{ijk} in Equation [2] produces maximum net economic returns? Note that if all possible bucking policies for all stem classes are initially generated and included in Equation [2], the second stage problem becomes irrelevant and unnecessary. However, the total number of bucking policies as defined and described in Table 1 is usually very large. Hence, the purpose of the second stage problem is to generate only the bucking policies that can potentially improve the total net economic return.

The second stage optimization problem is formulated as:

For a given stem class i,

$$Max\ \overline{C}_j = C_j - \sum_{k=1}^{K} a_{jk}\ \pi_k \qquad [6]$$

TABLE 1. — *Alternative bucking policies for a 90-foot stem class.*

Log class	Bucking policy 1	Bucking policy 2
8.25 ft.	3	4
10.25 ft.	2	3
12.25 ft.	2	2
14.25 ft.	1	0
Total lengths	84 ft.	88.25 ft.
Waste	90 ft. − 84 ft. = 6 ft.	90 ft. − 88.25 ft. = 1.75 ft.

subject to:

$$\sum_{k=1}^{K} a_{jk}\ l_k \leq L \qquad [7]$$

$$a_{jk} \leq n_k \qquad [8]$$

a_{jk} are non-negative integers

where:

$\overline{C}_j$ = a measure [identical to the $(Z_j - C_j)$ objective function coefficient of a Simplex Tableau] used in evaluating the optimality of the generated bucking policy

C_j = cost of bucking (depending on the amount of waste) associated with bucking policy j

a_{jk} = the number of log type k's produced from the ith stem class using bucking policy j

l_k = length of log type k

L = length of the ith stem class

π_k = Lagrange Multiplier

n_k = maximum number of log type k's obtainable from the ith stem class

Problems of the form similar to the second stage decision problem, without Equation [8], are called *knapsack problems*. Several approaches have been devised to solve these types of problems. Gilmore and Gomory (5) described how dynamic programming can be used as a solution approach. They have also devised a computational procedure which is more efficient than dynamic programming. Mendoza (7) modified Gilmore and Gomory's procedure and devised a *modified knapsack algorithm* to solve the second stage problem.

Interfacing the optimization problems

Bradley et al. (1) showed how large-scale linear programming systems with a large number of variables can be solved efficiently using a method of column generation interfaced with a linear program. Due to the large number of variables, direct solution by the simplex method may be inappropriate. Aside from excessive computational requirements, simply generating all the technological coefficients usually prohibits this approach.

The column generation procedure very closely parallels the mechanics of the decomposition algorithm. The only difference concerns the definition of the *subproblem*. In the Dantzig-Wolfe decomposition algorithm, the subproblems are linear programming problems, whereas in the column generation approach, the subproblem need not be a linear program, but can be any type of optimization problem.

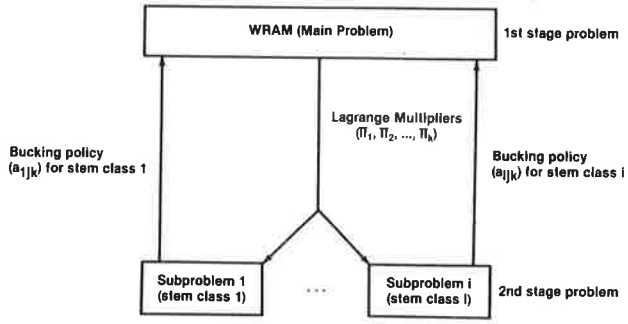

Figure 1. — A description of the two-stage decision problem.

The subproblem generates the columns of associated coefficients (a_{jk}'s) which are evaluated by the *main problem*. The interface between the two problems is made possible through the Lagrange Multipliers obtained from a solution of the main problem. For a more detailed presentation of the approach, interested readers are referred to Bradley et al. (1).

The direct analogy between the column generation procedure described in Bradley et al. (1) and the two stage decision problem described earlier is clear. The first stage decision problem (i.e. WRAM) constitutes the main problem, while the second stage decision problem constitutes the subproblem. These two decision problems are interfaced through the use of the Lagrange Multipliers as described in Figure 1.

Optimization in the integrated model

The interactive optimization procedure of Bradley et al. (1) applied to the two-stage decision problem is described in Figure 2. The flowchart may be described verbally by the following steps:

Step 1. Arbitrarily choose a specific bucking policy for each stem class (i.e. choose any feasible set of a_{ijk});

Step 2. Using these bucking policies, solve WRAM to determine the optimal allocation in terms of the optimal values of X_{ij}, Y_{kp}, Z_{mp}, and the Lagrange Multipliers, π_k for each constraint in Equation [2];

Step 3. Use current values of π_k to solve the subproblems in Equations [6]-[8] for all stem classes, and generate new set of bucking policies, a_{ijk};

Step 4. Evaluate the generated bucking policies, a_{ijk}, for all stem classes to determine if there are some which can potentially improve the value of the objective function. If there are, include them in the new set of a_{ijk}'s and go back to Step 2. If none, the iteration stops. The current set of log bucking and allocation policies are optimal.

Note that in Step 4, the generated bucking policies are evaluated to determine if they can potentially improve the value of the objective function. Those bucking policies associated to one or more stem classes which might yield some improvement are included in the new set of bucking policies and the process goes back to Step 2. The iteration continues until there are no more bucking policies in Step 4 which might improve the value of the objective function. This happens when the C_j values of all stem classes in Step 3 are greater than zero, or the set of generated bucking policies in Step 3 are already in the current set of a_{ijk}'s.

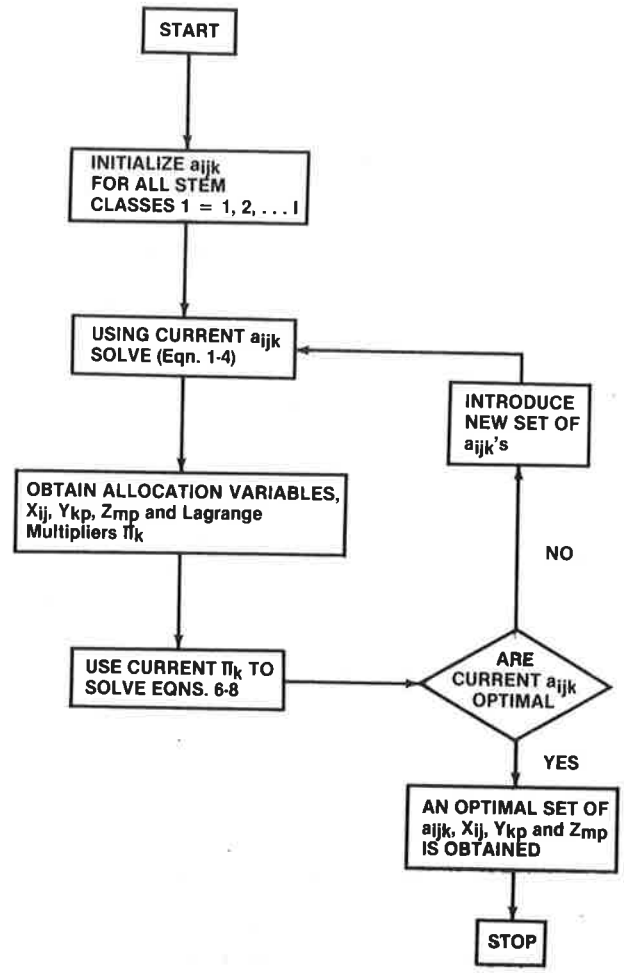

Figure 2. — Flowchart describing the procedure of the iteration process.

Summary

This paper has described a two-stage optimization procedure that simultaneously determines optimal log bucking and allocation. The procedure is based on the column-generation technique (1). Mendoza (7) has demonstrated the procedure using a sample problem adapted from Westerkamp (10).

The procedure works best when the planning problem involves a large number of log types or grades, and tree stem classes. In this situation, the model can improve computational efficiency by generating and evaluating only "promising" bucking policies, rather than enumerating and specifying all possible bucking policies.

Literature cited

1. Bradley, S.P., A.C. Hax, and T.L. Magnanti. 1977. Applied Mathematical Programming. Addison-Wesley Publ. Co., Reading, Mass. 716 pp.
2. Briggs, D.G. 1980. A dynamic programming approach to optimizing stem conversion. Ph.D. thesis, Univ. of Washington, Seattle, Wash.
3. Dantzig, G.B. and P. Wolfe. 1961. Decomposition algorithm for linear programs. Econometrica 29 (40).
4. Faaland, B., and D. Briggs. 1984. Log bucking and lumber manufacturing using dynamic programming. Management Sci. 30(2):245-257.
5. Gilmore, P.C., and R.E. Gomory. 1961. A linear programming approach to the cutting stock problem. Operations Res. 9(6):849-859.
6. McPhalen, J.C. 1977. A method of evaluating bucking and sawing strategies for sawlogs. M.S. thesis. Univ. of British Columbia, Vancouver, Canada.
7. Mendoza, G.A. 1980. Integrating stem conversion and log allocation models for wood utilization planning. Ph.D. diss. Univ. of Washington, Seattle, Wash.
8. Pnevmaticos, S.M. 1974. Optimal allocation methods for log production. Ph.D. thesis. Pennsylvania State Univ., University Park, Pa.
9. Tejavibulya, S. 1981. Dynamic programming sawing models for optimizing lumber recovery. Ph.D. diss. Univ. of Washington, Seattle, Wash.
10. Westerkamp, G.L. 1974. A linear programming approach to log allocation. M.S. res. pap. College of Forest Resources, Univ. of Washington, Seattle, Wash.