ELSEVIER

European Journal of Operational Research 119 (1999) 427–439

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/orms

A system for the design of short term harvesting strategy [1]

R. Epstein [a,*], E. Nieto [b], A. Weintraub [a], P. Chevalier [c], J. Gabarró [a]

[a] *Department of Industrial Engineering, University of Chile, P.O. Box #2777, Santiago, Chile*

[b] *Forestal Valdivia, Agustinas #1070-6° Piso, Santiago, Chile*

[c] *Université Catolique de Louvain, Institut d'Administration et de Gestion, Place des Doyens 1, 1348 Louvain-la-Neuve, Belgium*

Abstract

Short term harvesting requires decisions on which stands to harvest, what timber volume to cut, what bucking patterns (how to cut up the logs) to apply to logs in order to obtain products that satisfy demand and which harvesting machinery to use. This is an important problem in forest management and difficult to solve well in satisfying demand, while maximizing net profits. Traditionally, foresters have used manual approaches to find adequate solutions, which has shortcomings both in time spent by analysts and the quality of solutions. Since demand for timber products is defined by length, diameter and quality of each piece, this leads to a complex combinatorial problem in matching supply (standing trees) and demand. We developed one of the few reported approaches for solving the short term harvesting problem based on a computerized system, using a linear programming approach. Determining adequate bucking patterns is not trivial. We develop a column generation approach to generate such patterns. The subproblem is a specially designed branch and bound scheme. The generation of bucking patterns implemented within the LP formulation led to a significant improvement of solutions. We believe this is the first system implemented with this level of detail. This system has been advantageously implemented in several forest companies. The results obtained show improvements obtained by the firms of 5–8% in net revenues over traditional manual approaches.

Keywords: Natural resources; Forest management; Linear programming

1. Description of the problem

Forest companies use standing timber in their forests to satisfy demand at pulp plants, sawmills and also exports as logs. In our case, the firms handle pine plantations which mature in a cycle of 22–28 years.

Forests are divided into reasonably homogeneous stands, were the similarity is given mainly by tree age, site quality and management state.

When trees are harvested, they are cut into several products or pieces. This operation is called bucking. The pieces obtained must satisfy demand, which sets requirements in terms of volume (m^3) of pieces defined by their length and diameter, and often also average diameter of a whole lot. The

* Corresponding author.

[1] Research partially supporte by FONDEF under project FI-11.

0377-2217/99/$ – see front matter
PII: S 0 3 7 7 - 2 2 1 7 (9 9) 0 0 1 4 4 - 7

bucking process can be carried out on the ground and individual pieces transported to their separate destinations from there, or the whole log can be transported to a sorting center, where the bucking is carried out.

Decisions for short term harvesting involve the following.

(1) Which stands to harvest among those which have mature trees ready for harvesting and are already accessible by existing roads.

(2) What type of machinery should be used. Areas with steep slopes are harvested using yarders or cable logging while flat areas are harvested with skidders. Yarders are usually installed on a flat area overlooking a steep slope and large enough to allow a small stockpile of timber and room for trucks to be loaded. The yarder is rotated so that the cable is laid out in different directions to bring up all the timber reachable from the yarder. In some cases cable logging is done so that logs are brought down the hill, which is more complicated due to the danger caused by gravity. Skidders carry logs to the roadside or to landings, where they are loaded into trucks.

(3) Volume to be cut each week. The volume harvested is in direct relation to demand. Stockpiles can be made for future use.

(4) Bucking patterns. Bucking carried out either on the ground or in sorting yards must follow simple instructions in terms of lengths, diameters and quality. A typical bucking pattern could be as shown in Fig. 1.

Instructions given to loggers are to obtain as many pieces as possible of each defined piece, in the given order. Thus, in the example of Fig. 1, the logger first tries to get a piece which at a distance of 12.10 m from the bottom of the tree has a diameter of at least 26 cm. If he obtains that, he tries again for a piece of length 12.10 and diameter 26 cm. If the diameter is below 26 cm, he tries to obtain a piece of length 4.10 m and diameter at least 24 cm, and so on.

The bucking patterns lead to specific products in length, minimum diameter and, if needed, average diameter of a complete lot.

(5) Delivery of products to destinations to satisfy demand. One usually needs to satisfy different sales, each of which requires certain products, defined by length and minimum diameter. The requirements for these products are in terms of total volume of the pieces. In addition, as mentioned, export sales and often sawmill demand also have average diameter constraints. An example is given in Fig. 2.

Most frequent sales or deliveries are as follows.

(a) Exports of high diameter logs to countries such as Korea, Japan and Turkey. Often a sale is composed of several products. The example in Fig. 2 can be seen as an order for a sale of 4 different products, with an overall average diameter of 26.7 cm and bounds on total volume.

(b) To local sawmills. These are shorter logs, of minimum diameter of 16 cm.

(c) To the pulp plant. The pulp plant will take almost any product.

The commercial value of these pieces usually follows the same order as defined in the bucking patterns. An export piece can be degraded to sawmill, a sawmill piece can be degraded to pulp timber. Tree trunks become narrower as the distance from the ground increases. This explains why the first piece (from the bottom up) is always the one with highest diameter. Thus it is usually possible to degrade to smaller diameter products. Degradation involves losses, that can be measured in sale price. For example, typical sale prices for

Piece	Length (m)	Minimum diameter (cm)	Destination
First	12.10	26	export
Second	4.10	24	sawmill
Third	4.10	22	sawmill
Fourth	4.10	16	sawmill
Fifth	2.44	8	pulp

Fig. 1. Example of bucking pattern.

SALE 1 Export. Destination: Korea.						
Kind of export piece	Length (m)	Minimum Diameter (cm)	Average Diameter (cm)	Minimum Volume (m^3)	Maximum Volume (m^3)	Price ($/$m^3$)
Long	11.10	20	26.7	18,000	---	65
Medium	7.40	20	26.7	0	---	65
Short	5.50	20	26.7	0	2,400	65
Short	3.70	20	26.7	0	3,600	65
TOTAL	---	---	26.7	30,000	31,000	---

Fig. 2. Example of sale.

exports are about 65 $/m^3, 45 $/m^3 for sawmill timber and 25 $/m^3 for pulp timber. One of the main challenges in planning short term harvesting is to match supply of standing timber with demand, so as to minimize degradation.

For a given bucking pattern and a given stand, the volume obtained of each product is a random variable that will depend on the growth of each tree. Forests companies have developed product simulators based on taper functions which allow them to predict reasonably well, for a given bucking pattern, how much volume will be obtained for each product in the pattern. Typically, of the total volume harvested, export quality varies from 30 to 60%, sawtimber goes from 20 to 50% and pulp timber goes from 20 to 50%.

Models have been developed for harvesting and bucking problems. Typically these are LP models such as Burger and Jamnick (1991), Jamnick and Walters (1991), Morales and Weintraub (1991). When the stem bucking decisions are incorporated to the harvest planning problem, the difficulty increases. Because of the large number of possible bucking patterns, solution approaches incorporate different ways of generating a reasonable set of them either external to the LP (McGuigan, 1984), or internally, using Dantzig-Wolfe decomposition (Eng et al., 1986) or column generation (Mendoza and Bare, 1986). Dynamic programming (Briggs, 1989) and heuristics (Sessions et al., 1989) have also been used. Few harvest planning models incorporating bucking decisions have been applied. One of the few applications, for short term harvesting is in New Zealand, as shown by Garcia, 1990; Manley, 1993. However, we are not aware of implementations which allow the user to include machine scheduling and introduce bucking patterns, in spite of the fact these are levels of detail needed in our actual applications. Usually, stem bucking is considered at the processing facility to be done after harvest planning decisions have been taken and executed.

2. The traditional planning approach

The typical planning approach is carried out by an experienced planner. The planner analyses in the following manner.

(a) The demands for different products in the near future, one to three months ahead.

(b) The standing timber available in that period. Access given by existing roads plays an important role, as in this time frame, no roads are built. Road building is defined in a longer time framework, with time horizons of 2–5 years. Building a road takes several months and it is carried out in the summer season. Roads are either dirt or gravel. Gravel can be used year round whereas dirt roads are only useful in summer. Given any time of the year, it is known which roads can be used. There may also exist stocks of products stored in stocking yards.

(c) The capacity for harvesting. This depends mainly on the availability of harvesting equipment (yarders, skidders) and trucks for transportation.

With this information the planner schedules a stand sequence to be harvested and machinery to be allocated. Bucking patterns are defined such as to yield the products needed to satisfy demand each week, including stocks.

In this case, feedback is essential. As production advances, the planner looks each day at the results obtained. Suppose there is a shipment of 5000 m^3 to Japan, to be delivered in 2 weeks of pieces 12.10 m long, minimum diameter of 24 cm and average diameter of 27 cm. The instruction to the loggers could be to buck at 12.10 with a minimum diameter of 24 cm in a certain stand. Assume that after one week, production is 2000 m^3 with an average diameter of 25 cm. Clearly two changes need to be carried out: the pace of production needs to be accelerated, and the average diameter needs the be increased. The latter can be accomplished in two ways. One way is to move the harvesting operation to a stand with higher diameter trees. The second is to increase the instruction of minimum diameter, to 27 cm for example. This will increase the average diameter, but may lead to excessive harvesting, as more trees will be cut to obtain the required volume of 12.10 m pieces, and consequently to the need to harvest (and to degrade) more timber.

The feedback policy is necessary, as it is difficult to obtain the right combination of stands to harvest and bucking patterns. But this feedback leads to higher management costs, both in terms of excess volume in some products, for which there is no specific demand and which will have to be degraded. There are also increased operational costs in changing harvesting operations, and more timber will be harvested. This timber may be needed later, in particular at the end of the winter season when there is usually a scarcity of accessible stands. Another element of possible inefficiency in a manual approach is in a higher level of transportation costs.

3. The mathematical model

We present now a Linear Programming model that has been implemented to solve this problem.

Parameters

- REN_{ijk} — fraction of stand i that is transformed into product k, when using bucking pattern j.
- VOL_i — volume (m^3) of timber available in stand i.
- DM_k — average diameter (cm) of product k.
- DI_d — average diameter (cm) for sale d.
- CAC_t — production Capacity (m^3) in period t.
- DDAI_d — minimum demand (m^3) of sale d.
- DDAS_d — maximum demand (m^3) of sale d.
- DDAIP_d — minimum demand (m^3) for product k on sale d.
- DDASP_{dk} — maximum demand (m^3) for product k on sale d.
- PPED_d — set of periods in which it is feasible to produce timber for sale d.
- PROP_d — set of products in sale d.
- PV_{dk} — sale price (\$/m^3) for product k in sale d.
- COST_i — cost (\$/m^3) of harvesting in stand i. This cost includes cost of production, set-up for operations.
- CTA_{idk} — transportation cost (\$/m3) for product k between stand i and sale d.

Variables

- Y_{idkt} — volume of timber (m^3) transported from destination i to sale d, of product k in period t.
- K_{ijt} — volume of timber (m^3) produced in stand i, using bucking pattern j in period t.

Objective function

$$\mathrm{Max} \quad \sum_{i,d,k,t} (\mathrm{PV}_{dk} - \mathrm{CTA}_{idk}) Y_{idkt} - \sum_{i,j,t} \mathrm{COST}_i \cdot K_{ijt}.$$

Constraints

$$\sum_{j,t} K_{ijt} \leqslant \mathrm{VOL}_i \quad \forall i. \tag{1}$$

Total volume harvested is bounded by existing timber.

$$\sum_{i,j} K_{ijt} \leqslant \mathrm{CAC}_t \quad \forall t. \tag{2}$$

Timber harvested is bounded by machine capacity.

$$\sum_j \mathrm{REN}_{ijk} \cdot K_{ijt} - \sum_d Y_{idkt} \geqslant 0 \quad \forall i, k, t. \tag{3}$$

Volume transported is bounded by production of each product in each stand and period.

$$\mathrm{DDAI}_d \leqslant \sum_{k \in \mathrm{PROP}_d} \sum_{t \in \mathrm{PPED}_d} Y_{idkt} \leqslant \mathrm{DDAS}_d \quad \forall d. \tag{4}$$

Demand must be satisfied for each sale.

$$\mathrm{DDAIP}_{dk} \leqslant \sum_{t \in \mathrm{PPED}_d} Y_{idkt} \leqslant \mathrm{DDASP}_{dk} \quad \forall d, k. \tag{5}$$

Demand must be satisfied for each product of each sale.

$$\sum_i \sum_{k \in \mathrm{PROP}_d} \sum_{t \in \mathrm{PPED}_d} \mathrm{DM}_k Y_{idkt} \geqslant \mathrm{DI}_d \quad \forall d. \tag{6}$$

Average diameter requirements for each sale must be satisfied.

$$Y_{idkt} \geqslant 0, \; K_{ijt} \geqslant 0 \quad \forall i, d, k, j, t. \tag{7}$$

The model described is a simplified version of the model used in the forest companies. The model, embedded in a system called OPTICORT, considers the following extensions to the simplified version shown.

(i) There are different types of harvesting machinery and in each stand there exists a specific supply of timber for each type of machinery. Minimum and maximum bounds can be imposed on production by stand, type of machinery or global.

(ii) Macro-stands group sets of neighboring stands for the purpose of transport consideration. This reduces significantly the size of the model.

(iii) Demand in each sale can be divided into different length ranges.

(iv) Supply for a sale can be constrained to be regular over several periods.

(v) Average diameter constraints can also be imposed on specific products.

(vi) The opportunity cost of standing timber, which is not a financial one, is added to the objective function to avoid near sighted solutions, where more valuable trees could be harvested to save for example on transportation costs. While this would improve the short term profits of the horizon of this problem, it deteriorates the value of the firm in a longer horizon. This opportunity cost is not considered in the actual cash flow evaluation.

4. The product simulator

When the manager considers the problem of planning the production for the short term horizon, he needs the support of reliable information on the characteristics of the forest he wants to harvest. This information comes from product simulators, which are company specific, but rely on the same basic idea: simulate the application of a bucking pattern on a sample of trees from the stand. The shapes of trunks may be modeled through well defined functions that correlate the diameter of the trunk with the distance from the ground. These functions depend on parameters that can be measured on the trees, like the height and the age of the tree, the diameter of the trunk at 1.3 m from the ground, the height of the first branch, the height of a defect in the trunk if there is one, etc. Once these parameters have been obtained for each tree in the sample previously defined, one may calculate the diameter of the trunk at a certain height and subsequently evaluate the number of pieces it is possible to obtain and the volume and diameter of each piece of a product in a bucking pattern. This can be done either for each tree in the sample or for a diameter class if trees are previously grouped onto ranges of trunk diameter. Then the number of pieces, volume and diameter obtained for the sample can be extrapolated to the whole stand. Following the notation of the mathematical model exposed above, the inventory simulator yields the REN_{ijk} coefficients.

The model adopted for the tree representation is particularly significant. While there is a certain set of standard functions to render the shape of the trunk, the way of measuring the quality of the trees is peculiar to every company. Quality is affected by the defects one can find in the trunks and it constrains the products one can obtain for it. Some companies may consider it important to model every defect on the trunks, so the product simulator needs to know where the defect begins, where it ends, and the products one can get from the section below the defect and from the section

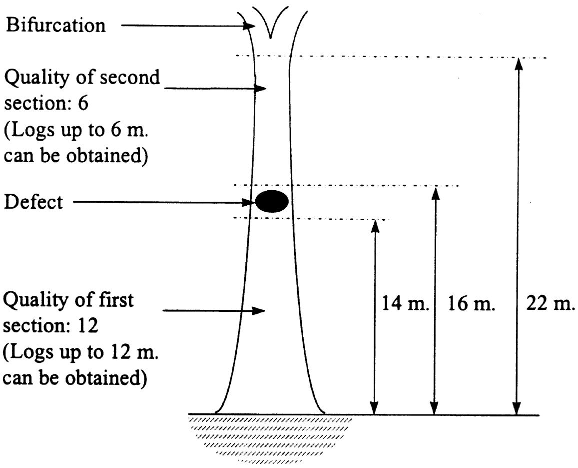

Fig. 3. Representation of simulation criteria in bucking patterns.

above it. This can be done for just one main defect on the trunk, or for any number of them. This approach leads to the evaluation of the simulator for each individual tree in the sample, and to longer processing times. Fig. 3 illustrates this way of representing quality.

Another approach is to assign a quality to the whole tree and then classify by quality. For example, one may have five ranges of quality. A product of length 11.10 m, which is a long piece for export, can only be obtained from trees of the first and second range, while a pulp piece can be obtained from any range. If trees are then classified by diameter classes, it can be known, for each diameter class and each quality class, how many trees there are in the stand, and a bucking pattern can be applied to a representative tree of each class. This approach leads to reduced processing times while the precision of the simulation, compared to the previous approach, is roughly similar. Reduced processing times are important in our case. For the column generation algorithm, time is a relevant constraint, as we intend to evaluate a very large number of bucking patterns.

5. Scheme for generating bucking patterns

The total number of bucking patterns is very large, so a user can only define a relatively small number of patterns to keep down the size of the model and more importantly, the amount of work needed to evaluate the timber yields for each product in each bucking pattern, using the inventory simulator. It is not trivial to define a set of bucking patterns for a stand which yields an adequate and varied choice of alternatives for the model.

We use a column generation approach to generate additional bucking patterns. First, the model is set with a number of bucking patterns defined for each stand. As already stated, a set of diverse patterns is defined for each stand, so that there is a good variety of choices for the model to pick. Patterns are built so as to include the products that are demanded. Given a solution to the LP problem, new bucking patterns are generated as follows.

In the original model, there are three sets of constraints that involve bucking patterns.

Dual Variables:

$$\sum_{j,t} K_{ijt} \leqslant \mathrm{VOL}_i \quad \forall i, \tag{8}$$

$$\sum_{i,j} K_{ijt} \leqslant \mathrm{CAC}_t \quad \forall t, \tag{9}$$

$$\sum_j \mathrm{REN}_{ijk} \cdot K_{ijt} - \sum_d Y_{idkt} \geqslant 0 \quad \forall i,k,t. \tag{10}$$

Let γ_i, δ_t and β_{ikt} be respectively the dual variables of constraints (8)–(10). Then the reduced cost of a bucking pattern variable K_{ijt} will be

$$\overline{C_{ijt}} = \mathrm{COST}_i - \left(\gamma_i + \delta_t - \sum_k \beta_{ikt} \cdot \mathrm{REN}_{kij}\right). \tag{11}$$

We need to find bucking patterns with value $\overline{C_{ijt}} > 0$. Note that bucking patterns are generated for each stand and period. Let us define

$$\alpha_{it} = \gamma_i + \delta_t. \tag{12}$$

The dual variables α_{it} represent the value assigned to the products already defined for stand i in period t, while β_{ikt} represents the value of a product k obtained in period t in stand i.

Before presenting the scheme for generating bucking patterns in a subproblem we need to define the characteristics of legitimate bucking patterns.

- There is a natural sequence in which products are typically defined, in decreasing value of length and diameter, and of commercial value.
- The first piece is typically an export piece, or for pruned stands, a knot free piece. The last piece is always destined for pulp.
- There is a minimum of 3 pieces and a maximum of 9 pieces that can be defined in a bucking pattern.

The model for the subproblem to generate bucking patterns is the following.

Let

$$X_{kf} = \begin{cases} 1 & \text{if the bucking uses product } k \text{ in} \\ & \text{position } f, \\ 0 & \text{if not,} \end{cases}$$

$k = \{1, \ldots, K\}$ represents the set of all products demanded. $f = \{1, \ldots, f_{\max}\}$, where $f_{\max}$ is the maximum number of pieces allowed in a bucking scheme.

When determining the timber yields for each product or piece we need to find the fraction of total volume of the log that corresponds to that particular piece. This volume will depend on the characteristics of the product (length and diameter), its position in the bucking pattern, but it must be noted that it also depends on the products that preceded it in the pattern. This can be seen in Fig. 4.

We have here 2 bucking patterns on a same log. Pattern 1 has as a first product of length 12.10 m and minimum diameter of 24 cm and a second product of length 4.10 m and minimum diameter 20 cm and then pulp.

Pattern 2 has as a first product length 8.10 m and minimum diameter 24 cm and as second product length 4.10 m and minimum diameter 20 cm and then pulp. Fig. 4 shows how the bucking pattern results for a specific log. In this case we can see that for pattern 2 two pieces of product 4.10 m in length and 20 cm of minimum diameter are obtained, while in pattern 1 only one piece of that product is obtained. So the yield of a given product k will depend on the bucking pat-

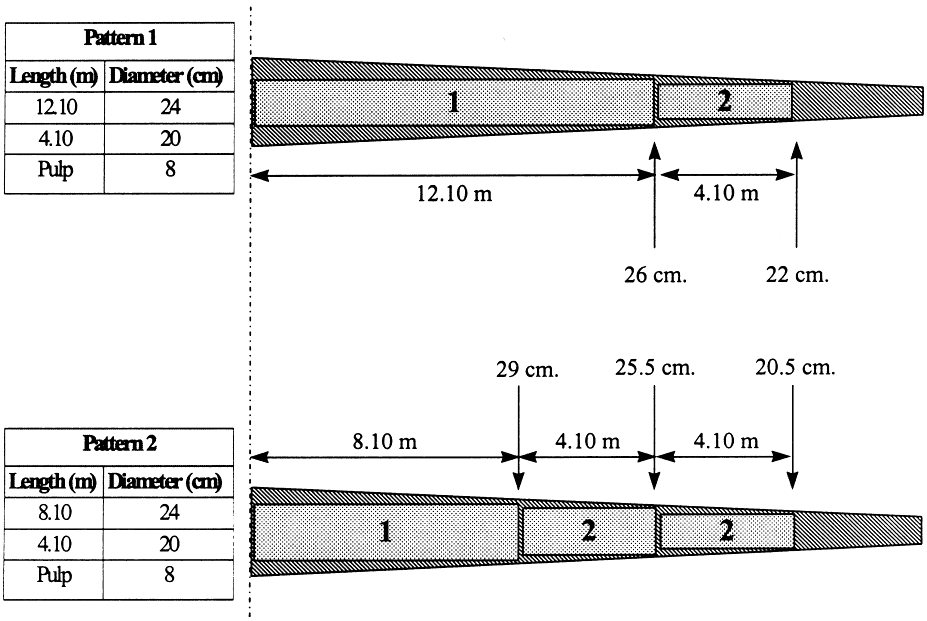

Fig. 4. Analysis of the use of different bucking patterns.

tern used, defined as $\mathrm{REN}_k(x)$, where $x = (X_{k1,1}, \ldots, X_{kr,r})$ defines the bucking pattern for r products with $r \leqslant f_{\max}$.

Note that the specific yields that will be obtained for each product in a given bucking pattern can be estimated through the use of the product simulator already described. The subproblem will then look for a bucking pattern X, with value

$$\sum_k \sum_f \beta_k \mathrm{REN}_k(X) X_{kf} \text{ larger than } \alpha_{it}.$$

To generate these patterns, a branch and bound algorithm was developed. In this case, nodes represent decisions on the next product to define, so that for each node there will be multiple branches representing possible products that can be added to the pattern at that point. The root node indicates a decision on the first product defined for the log. The maximum number of levels branching tree equals $f_{\max}$, the maximum number of products in a bucking pattern. Fig. 5 shows an example of a branching.

In each branch the length and minimum diameter of a product is defined. For example, branch 1–2 defines a product of length 12.10 m and minimum diameter 24 cm.

The value at each node is determined by the addition of the values of each branch leading to that node, e.g., the value of node 14 is defined by the value of products (12.10, 24 / 4.10, 30 / pulp). The yields of each product are determined by the product simulator. Suppose the yields for node 14 in the example have proportions (50%, 30%, 20%) then for given dual values $\beta_1, \beta_2, \beta_3$, for these 3 products and α for the corresponding stand all in period t, the value of node 14 will be

$$\alpha - 0.5\beta_1 - 0.3\beta_2 - 0.2\beta_3.$$

So, the basic algorithm for generating bucking patterns works as follows.

(1) Start at the root node.

(2) For a given node, evaluate its value, as shown above. If it is not a terminal node, determine a bound on the best value that can be obtained through that node. Depending on the value of the bound, either eliminate it from further processing or go to (3). If it is a terminal node its value corresponds to a feasible bucking pattern.

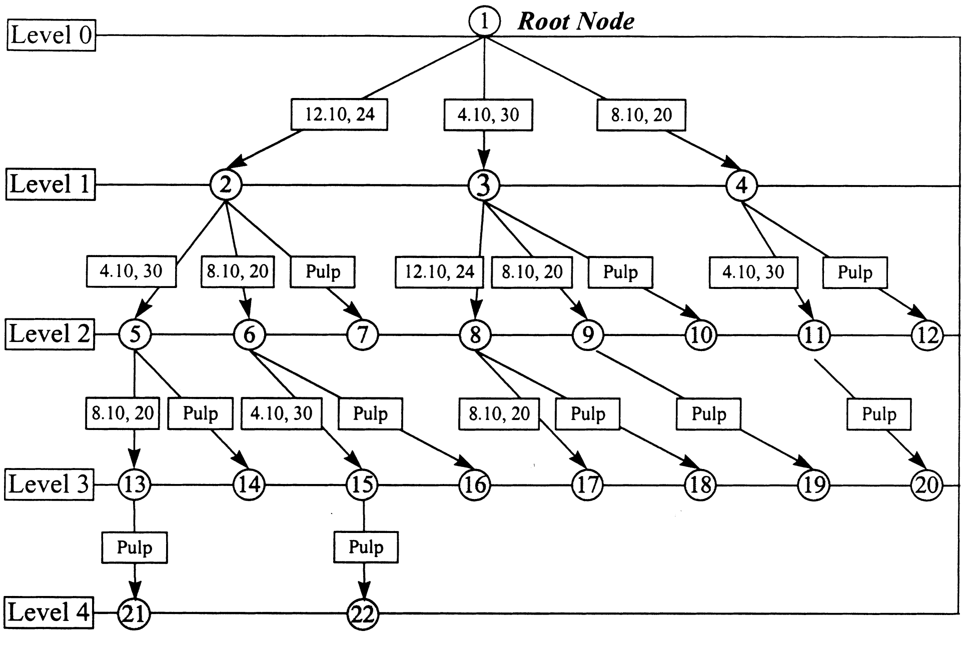

Fig. 5. Nodes in a branch and bound tree for the generation of bucking patterns.

(3) If the node has not been eliminated, branch on it, adding products consistent for that level.

(4) Find a new node to evaluate, go to (1). If none exists (all $\overline{C_{ijt}} \leqslant 0$ in Eq. (11)), end.

The process ends in principle when all nodes have been scanned. However, since we do not need an optimal solution except for the final iteration, the process may stop whenever one, or a set of improving bucking patterns have been found to be sent to the master LP.

What has to be defined are the following.

(i) The branching rules (given a node, which product is tested to be added to the pattern at that stage).

(ii) What node is to be evaluated next.

(iii) The bounding rules.

It is a clear that a commercial LP package, with 0–1 variables cannot handle this type of branch and bound process. So, we first developed a specific flexible Branch and Bound Shell.

This Shell, called Baobab (Chevalier, 1995) provides a framework in which the user can write individual branching and bounding rules.

6. The branch and bound shell

The branch and bound shell was conceived in an object oriented programming (OOP) approach, and written in C++, so that the user can write his or her own code for branching and or bounding rules. The default is to use a set of conventional branching rules and an LP code (CPLEX) for bounding.

The main target of Baobab is to provide a simple way of implementing branch and bound rules. This is done, using the OOP capabilities, by isolating certain problem dependent routines so the user can modify them without any further change in the code that manages the branching tree and the data of the problem. Structures have been created to handle the following.

The problem: All the data defining the problem is stored in one place, with routines that perform the calculations needed to evaluate a certain solution: objective function value, right hand side values, slacks, etc.

Relaxation: The problem has access to a structure that provides a link to the chosen relaxation code. This can be done, for basic implementations, with a commercial LP package such as CPLEX.

Nodes: Each node stores information about its relatives in the branching tree and provides routines to manage the upper and lower bounds. The bounds of a node can be updated according to the bounds of its descendants, thus passing the bounding information from lower to higher levels in the tree.

Branching rules: Each node has access to the branching rules which are clearly isolated from the rest of the program.

Solutions: Solutions can be stored at each node, providing information on which variables have been fixed and the values of all the variables, as well as the feasibility of the solution. Each node has access to its own solution.

Node lists: The program handles a list of the nodes that have been created and the priority by which they are going to be picked up for branching.

A main loop links all the data structures and performs the branch and bound algorithm. The program allows a user to build his or her own rules, like multiple branching, heuristics to determine the variable on which branching will be done, etc.

The column generation was implemented to generate sets of bucking patterns to add to the master program in each iteration. At initial iterations, more patterns are generated, quite fast. When approaching optimality, it becomes more difficult to generate improving patterns. At each iteration, one stand for a determined period is examined to find bucking patterns to add to the master problem. The sequence in which stands are evaluated is part of the computational strategy.

In our case, the branching rules were specifically adapted for this problem. Given a node, a set of heuristic rules determined best products to be added as a next choice. The rules were based on logical bucking patterns and an analysis of products most likely to be needed.

The selection of next node depended in part on user preferences for depth first or breadth first search. The selection of nodes, given those preferences, was based on the quality of each node, measured by the value of the node and the projection of the value of the remaining products that can be obtained in the remaining part of the log.

7. Computational implementation of the column generation

7.1. Branching rules

There are several ways in which nodes to branch are selected.

(a) Branch in the same order nodes are created. This corresponds to a breadth first approach.

(b) Branch on nodes in inverse order of creation. This approach can be attractive if the first branches created correspond to the most valuable products. It leads to a depth first branching.

(c) Branch on nodes with higher value. This also corresponds to a depth first approach.

(d) Develop all branches at the root level and then in inverse order of creation, a combination of breadth and depth first search.

(e) Branch entirely on the root node, and then in the node with higher value.

7.2. Bounding rules

The bounding for a node was based on observing, for that node, what products have already been defined, the length still available for new products and the diameter for the first additional product. With this information we can find a best idealized pattern for the remaining part of the log. For example, in node 6 of the branching tree of Fig. 5, we have a remaining length of 14 m (the two first pieces use up to 20 m) with a diameter of 20 cm. Defining for the remaining 14 m one product of length 8.10 m and diameter 20 cm and a final product of 6.10 m in length for pulp is an over estimation of the value of the node. This value provides an upper bound on the node for the remaining part of the branch.

A typical problem is described next in reduced form. Eight production zones contain a total of 28 stands available for harvesting. Ten products are demanded through 8 specific sales, with given prices and volumes required, as described in Fig. 6. Sale 1 is an export sale. Sales 2–4 demand sawmill timber of length 4.10 m and different minimum diameters. As none of these sales require average diameter or volume, they are grouped in one table. Sale 5 also demands sawmill timber, but it requires a minimum average diameter. Sales 6–8 correspond to different pulp plants.

This problem was first set up with 51 bucking patterns per stand. It leads to an LP problem of 1138 constraints (886 are flow conservation constraints) and 3859 columns (1378 are transportation variables and 2256 are bucking variables).

A termination criteria of a maximum of 100 bucking patterns generated was used if no optimal solution was yet found. The set of starting bucking patterns assigned covers a reasonable variety of possibilities. Runs were made in a PC486 using CPLEX as the LP code. Fig. 7 shows the comparisons for the five branching rules.

Iterations are defined by calling the Branch and Bound Shell to solve the subproblem. Positive iterations are those in which at least one column is added to the master. In negative iterations no columns are added. As can be seen, both depth first approaches are very similar. The best strategy appears to be seeking first all the branches of the root node and then exploring the nodes with best value.

A choice has to be made on how many improving patterns are to be generated in each iteration (if any exist). The choice goes from stopping after a first improving pattern has been found, to solving the subproblem to optimality. Fig. 8 shows the tradeoffs involved in CPU time vs quality of solution for three alternatives: taking the first improving pattern found, taking the first three improving patterns found, and solving to optimality and then taking the best three patterns. Generating several bucking patterns per iteration is clearly better in terms of CPU time, while solving to optimality takes a significantly longer time to get a similar improvement in the objective function.

Fig. 9 shows, for a real, larger problem, how the objective value improves as new patterns are generated.

Sale 1.						
Length (m)	Minimum diameter (cm)	Average diameter (cm)		Volume (m3)		Price $/m³
		Minimum	Maximum	Minimum	Maximum	
11.10	20	26.3	∞	121,200	∞	65.0
7.40	20	26.3	∞	0	∞	49.0
5.50	20	26.3	∞	0	16,120	38.0
3.50	20	26.3	∞	0	24,180	38.0
TOTAL :		26.3	∞	0	202,000	

Sales 2, 3 and 4.						
Length (m)	Minimum diameter (cm)	Average diameter (cm)		Volume (m3)		Price ($/m³)
		Minimum	Maximum	Minimum	Maximum	
4.10	16	0	∞	0	∞	36.0
4.10	20	0	∞	0	∞	40.0
4.10	24	0	∞	0	∞	48.0
4.10	30	0	∞	0	∞	57.0
4.10	38	0	∞	0	∞	64.0
TOTAL :		0	∞	20,000	∞	

Sale 5.						
Length (m)	Minimum diameter (cm)	Average diameter (cm)		Volume (m3)		Price ($/m³)
		Minimum	Maximum	Minimum	Maximum	
4.10	20	25.0	∞	0	71000	48.0

Sales 6, 7 and 8.						
Length (m)	Minimum diameter (cm)	Average diameter (cm)		Volume (m3)		Price ($/m³)
		Minimum	Maximum	Minimum	Maximum	
Pulp	8	0	∞	0	∞	25.0

Fig. 6. Demand for products in the example problem.

		a	b	c	d	e
Positive iterations		61	100	100	56	100
Negative iterations		51	137	137	56	135
Average number of nodes scanned	per positive iteration.	23	34	35	27	28
	per negative iteration	133	121	121	127	136
Total CPU (sec)		975	2160	2175	1031	2285
Improvement in Objective Value (%)		4.33	5.17	5.17	4.33	7.86

Fig. 7. Comparison of node selection approaches.

It was observed that there are significant jumps in objective value when certain patterns are introduced.

Fig. 9 corresponds to a larger example, of a real problem, with 132 stands and 35 products, initially considering 10 bucking patterns per stand. The LP dimensions are 3348 constraints and 7584 variables. CPU time for solving the original LP was 150 seconds. The subproblem was solved using approach (e) for the branching. Pattern generation is carried almost to optimality either for the master problem (setting 3000 bucking patterns generated as termination criteria) as for the subproblem (setting 2000 nodes scanned as termination criteria

	1 pattern	3 patterns	To optimality
Positive iterations	100	39	44
Negative iterations	137	28	68
Number of nodes per it.	164	182	246
Total CPU time (SCS)	2282	716	1414
% Improvement in objective value	7.86	7.79	8.57

Fig. 8. Comparison of number of patterns generated per iteration.

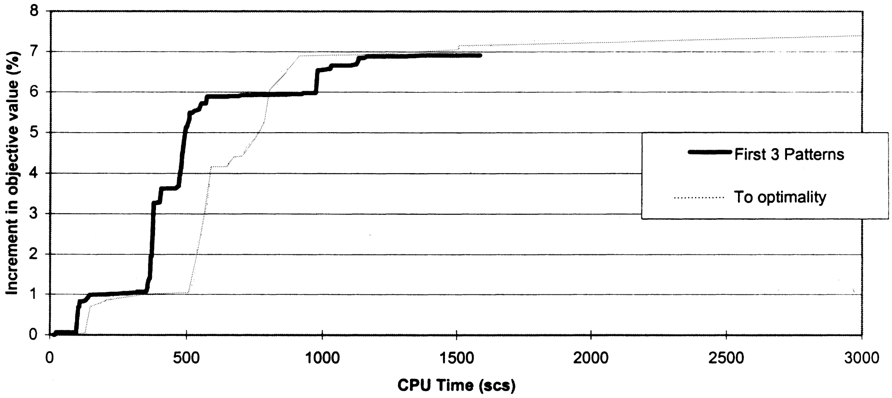

Fig. 9. Increment on the objective function trough time.

for each iteration) and, as Fig. 9 shows, solution time is significantly longer, while the improvement on the objective function is very small for most patterns created. The patterns that create large improvements in value correspond to those that include products which are in short supply to meet demand, and given the present solution belong to profitable and balanced combinations. As can be seen using the column generation scheme improves the original LP solution by 6% after 1000 seconds and by 8% after 10 000 seconds. The user needs to define how far he or she wishes to proceed in generating additional bucking patterns.

8. Implementation of the system

The optimization part of the system proposed in this paper without the column generation scheme was implemented in a program called OPTICORT around 1992. This program was then implemented and used in four Chilean companies for 5 years, Bosques Arauco, Forestal Chile, Forestal Bío Bío and Forestal Mininco. Improvements have been obtained in the use of the system, as reported by the companies.

Better use of standing timber. Matching supply and demand for specific products led to less degradation of timber, easier satisfaction of minimum diameter constraints, more availability of timber at the end of the winter season. Savings based on better utilization of the timber in one specific firm, Bosques Arauco were estimated at about \$2000 000 US per year corresponding to a 5% less degradation on a production of 2 million m^3. In another company, Forestal Chile the use of the system led to an estimated lower harvesting of about 200 000 m^3 per year (more timber is left standing), and an improvement of about 5% of net income per m^3, as the model leads to a better mix of products and usually yields exact solutions in terms of average diameter. Note that for each sale, it is inadmissible to be below the requested average diameter, and to be above it means giving away money as the customer does not pay for the extra diameter. The system was also useful in assigning the harvesting machinery and in controlling daily production schedules, and ensuring that demand for each product was satisfied. Finally, the model has been used to evaluate potential new business, such as buying up timber from third parties and the impact of production limitations due to environmental constraints.

Similarly, in Forestal Bio Bio the system has been helpful in minimizing the amount of timber degraded, scheduling the production of the mix of products for local sawmills, regulating the average diameter for export logs and minimizing used the use of stocks of export logs at the port. Finally, the model has been used in evaluating the overall Operative Plans for each season as well as to estimate the likelihood of satisfying the demand for the remainder of the season at any given time.

The companies have also reported savings in transportation costs due to better planning of production. In the one company, Bosques Arauco, the model indicated that about 50% of the volume transported through intermediate sorting centers could be transported directly to destinations. Eliminating this added transference operation, plus a better allocation of timber to sales led to savings of about 250 000 US dollars per year, due to reduced hauling and loading operations. In one study (Villalobos, 1996) tests were made, based on real data, to compare the traditional, manual scheduling with results given by the model. The improvement given by the model, was between 3% to 7% depending on how tight the demand for products was. In situations were demand is loose, and there is ample flexibility to satisfy it with available timber, the manual approach has less difficulties in finding good situations.

The column generation scheme has been installed recently in two companies, and should lead to additional improvements as shown above.

References

Briggs, D.G., 1989. Tree Value System: Description and Assumptions. USDA Forest Service, General Technical Report PNW-239, Portland, OR.

Burger, D., Jamnick, M., 1991. Analysis of wood procurement strategies: Supplying multiple mills from multiple sources. In: Proceedings of 1991, Symposium on System Analysis in Forest Resources, Charleston, South Carolina, USDA Forest Service, General Technical Report SE 74, Asheville, NC, pp. 17–23.

Chevalier, P., 1995. Baobab, Instructions Manual. Department of Industrial Engineering, University of Chile, Santiago.

Eng, G., Daellenbach, H.G., Whyte, A.G.D., 1986. Bucking tree-length stems optimally. Canadian Journal of Forest Research 16, 1030–1035.

Garcia, O., 1990. Linear programming and related approaches in forest planning. New Zealand Journal of Forest Science 20 (3), 307–331.

Jamnick, M., Walters, K., 1991. Harvest blocking, adjacency constraints, and timber harvest volumes. In: Proceedings of 1991 Symposium on System Analysis in Forest Resources, (Charleston, SC, March 3-6, 1991), USDA Forest Service, General Technical Report SE-74, Asheville, NC, pp. 255–261.

Manley, B., 1993. New Zealand experience in linking short term and long term forest planning. In: Proceedings International Symposium on System Analysis and Management Decisions in Forestry, Valdivia, Chile, pp. 315–329.

Morales, R., Weintraub, P., 1991. A model for strategic planning in the management of forest industries. In: Proceedings of 1991 Symposium on System Analysis in Forest Resources, (Charleston, SC), USDA Forest Service, General Technical Report SE 74, Asheville, NC, pp. 114–117.

McGuigan, B.N., 1984. A log resource allocation model to assist the forest industry manager in process selection and location, wood allocation and transportation, and production planning. APPITTA 37 (4), 289–296.

Mendoza, G.A., Bare, B.B., 1986. A two-stage decision model for bucking and allocation. Forest Products Journal 36 (10), 70–74.

Sessions, J., Olsen, E., Garland, J., 1989. Tree bucking for optimal stand value with log allocation constraints. Forest Science 35 (1), 271–276.

Villalobos, R., 1996, Benchmarketing de Incorporación de Software de Optimización de Corte en Empresas Forestales, Engineering Thesis, Universidad de Chile, Santiago.