CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Supply Network Planning in the Forest Supply Chain with Bucking Decisions Anticipation

Satyaveer S. Chauhan
Jean-Marc Frayret
Luc LeBel

May 2008

CIRRELT-2008-16

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince, local 2642
Québec (Québec)
Canada G1K 7P4
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Université Laval · UQÀM · HEC Montréal · École Polytechnique Montréal · Université de Montréal

Supply Network Planning in the Forest Supply Chain with Bucking Decisions Anticipation

Satyaveer S. Chauhan[1] , Jean-Marc Frayret[2,3,*] Luc LeBel[2,4]

[1] Department of Decision Sciences and Management Information Systems, John Molson School of Business, University of Concordia,1455 de Maisonneuve West, Montréal, Canada H3G 1M8

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[3] Département de mathématiques et génie industriel, École Polytechnique de Montréal, C.P. 6079, succursale Centre-ville, Montréal, Canada H3C 3A7

[4] Département des sciences du bois et de la forêt, 2405, rue de la Terrasse, Pavillon Abitibi-Price, Université Laval, Québec, Canada G1V 0A6

Abstract. We consider a two-echelon timber supply chain in which the first echelon consists of several stands to be harvested and the second echelon consists of mills to be procured with logs of different length. This problem aims at minimizing harvesting and transportation costs for one production period, while satisfying demand expressed as mix of volumes of specific log types. Harvesting cost, which includes felling, bucking and hauling to roadside, depends upon the number of log type to be produced and sorted. Each stands to be harvested are modeled individually with a limited number of trees of various classes of diameter and total length, which affects the productivity factors of the bucking patterns to be used. To take these characteristics into account, we propose heuristics based on columns generation to solve the supply network problem at the forest level with an anticipation of bucking operations at the stand level.

Keywords. Supply network planning, bucking, integer programming, heuristic, forest supply chain, columns generation.

Acknowledgements. This work was funded by the FOR@C Research Consortium in E-Business in the Forest Products Industry. The authors would like to thanks Jean Favreau and Joseph Nader of the Forest Engineering Research Institute of Canada for their support.

---

* Corresponding author: jean-marc.frayret@cirrelt.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2008

1. Introduction

Due to various economical and environmental factors, forest companies in eastern Canada are faced with the need to improve simultaneously their ability to maximize timber recovery with regards to customer/mills needs (due to reduced allowable cut on public land) and minimize their cost of operations (due to fierce competition and increased Canadian Dollard value). If several directions should be investigated to support companies to do so, it seems necessary to develop a forest procurement system that is reactive and driven by mills' demand and, ultimately, by consumer needs. This paper proposes to address short term procurement planning in a context of multiple sources of raw material and multiple mills, in order to synchronize harvesting, bucking and transport operations.

The remainder of this paper is organized as follow. Section 2 presents a review of the literature in the domains of bucking and transportation optimization in the context of the forest supply chain. Section 3 introduces and models the specific problem that is tackled here. Section 4 presents the solution approach we propose. Finally, Section 5 presents a quantitative evaluation of the proposed approach.

2. Literature review

Bucking optimization problems in the forest supply chain concerns the optimization of the cross-cutting of stems into various types of logs. According to Laroze (1999), bucking optimization may be performed at the stem level, the stand level or the forest level.

At the stem level, the objective is to maximize the stem value by selecting the bucking pattern (i.e., sequence of logs to be cross-cut) that maximizes the potential revenue using anticipated market prices and considering, in general, only operational constraints. Stem level bucking, also called buck-to-value bucking optimization, can either be carried out directly in the field, using cut-to-length harvesting equipments, or at the mill, using more advanced scanners to capture the stems' geometrical profile. Because mills' scanners are more accurate than the profile models used in harvesters, bucking at the mill provides better lumber recovery factors then bucking in the forest (Forintek Canada 2005). Stem level bucking can be either modeled as an independent operation to optimize (Pnevmaticos and Mann 1972; Geerts and Twaddle 1984), or it can be integrated with other supply chain operations. For instance, Faaland and Briggs (1984) and Maness and Adams (1991) propose to synchronize bucking and sawing operations in order to improve the overall process of lumber production, with the specific goal of maximizing the potential lumber production value. Individual stem bucking can also be synchronized with a mill's demand through the in-the-field dynamic adjustment of the market price list (Kivinen and Uusitalo 2002; Murphy et al. 2004). These approaches basically compare for each new stem to buck, the log mix already produced with the target log mix and adjust the price list (or the definition of the log type characteristics, such as the small end diameter) in order to influence the mix of log to be produced.

In order to further improve the match between supply and demand, stand level bucking models both supply and demand and optimizes bucking operations in a planning mode, before it can be carried out in the field. The objective here is to maximize the value of the production of the entire stand by generating and assigning several bucking patterns to each class of stems. At this level, constraints regarding market restrictions (i.e., volumetric proportion of each log class, small end diameter) and stand characteristics (i.e., stem class distribution) are usually taken into account. Stand level bucking optimization involves generally some form of problem decomposition where stem level sub-problem optimization is coordinated a the stand level through various optimization techniques including shadow price mechanisms (Sessions et al. 1989), dual decomposition techniques (Eng and Daellenbach 1985; Mendoza and Bare 1986) and priority and attribute-based rules (Laroze and Greber 1997). As pointed out by Sessions et al. (1989), although these approaches are computationally correct, they lead to bucking rules that are difficult to implement in the field. In order to improve this aspect, Marshall et al. (2006) propose and test different buck-to-value decision models to implement a stand level buck-to-order plan. The authors show the importance of having a buck-to-order plan to obtain good order fulfillment.

Finally, bucking can be integrated with procurement activities, which includes the transportation of logs to mills. This type of bucking problem considers the harvesting of several stand simultaneously. It is called forest level bucking and involves defining a bucking program to assign to each stand in order to maximize a companies' profit, considering demand constraints (i.e., portion of each log type,), market restrictions (i.e., small end diameter, minimum and maximum volume per log type) and stands characteristics (stem diameter distribution) and size. For instance, Arce et al. (2002) proposes an approach that maximizes the net profit of harvesting each stem in each stand, considering stands' log class distribution and demand for each log class. This model does not, however, consider several mills with different demand profiles. Along the same line, Kivinen (2004), and more recently Kivinen (2006), present an approach which aims at coordinating the price matrix of several stand, considering their local characteristics and customer demand. In other words, the goal is to find stand specific price matrixes in order to maximize demand satisfaction. This approach may result in various demand profile allocated to each stand. Once again, the demand profile of a single mill is considered here. Finally, Laroze (1999) proposes a combined linear programming/tabu search procedure to coordinate the selection of bucking rules to apply to specific areas of each stand in order to maximize net profit, subject to market and supply availability constraints.

In the decision support model presented in this paper, forest-level bucking involves the integrated optimization of harvesting and aggregated transportation operations. In other words, we consider a multi-facility procurement system where planning decisions concern the allocation of demand from several mills expressed in terms of volume of logs, to several stands, considering both harvesting and transportation costs, as well as an anticipation of operational constraints including supply availability and bucking. In a context where several log types are produced in each stand, harvesting production cost plays a non-trivial role. Indeed, because the introduction of a new log type (i.e., log of a specific length and specie) to separate at the stump reduces harvester productivity by 1 to 4% and forwarder productivity by 3 to 7%, (Brunberg and Arlinger 2001; Gingras and Favreau 2002), harvesting cost is a non-linear function of the discrete number of log type to be produced in each stand (Chauhan et al. 2005).

Concerning transportation operations, the model proposed in this paper is limited to direct delivery from stands to mills. We do not consider the use of a sorting yard (Sessions et al. 2005), nor the use of backhauling, which could decrease significantly sorting and transportation costs (Carlsson and Rönnqvist 2005). These issues remain for future work.

The next section describes the problem and presents the mathematical formulation. Next, section 3 introduces a column generation-based solution approach with fast heuristic to solve the problem. The developed approach is then tested on randomly generated data set using stand profile distribution developed by the Forest Engineering Research Institute of Canada. The performance of the algorithm is presented in section 4. The last section concludes and future research directions are highlighted.

3. Problem description and formulation

This section first introduces the general problem tackled in this paper, and second, proposes a formal mathematical modelling of both aspects of the general problem: the procurement planning problem and the forest-level bucking problem.

3.1 Problem general description

As mentioned in the previous section, this paper addresses a short term multi-stand multi-facility procurement problem with non-linear harvesting cost, which includes a specific instance of the general forest level bucking problem (Figure 1).

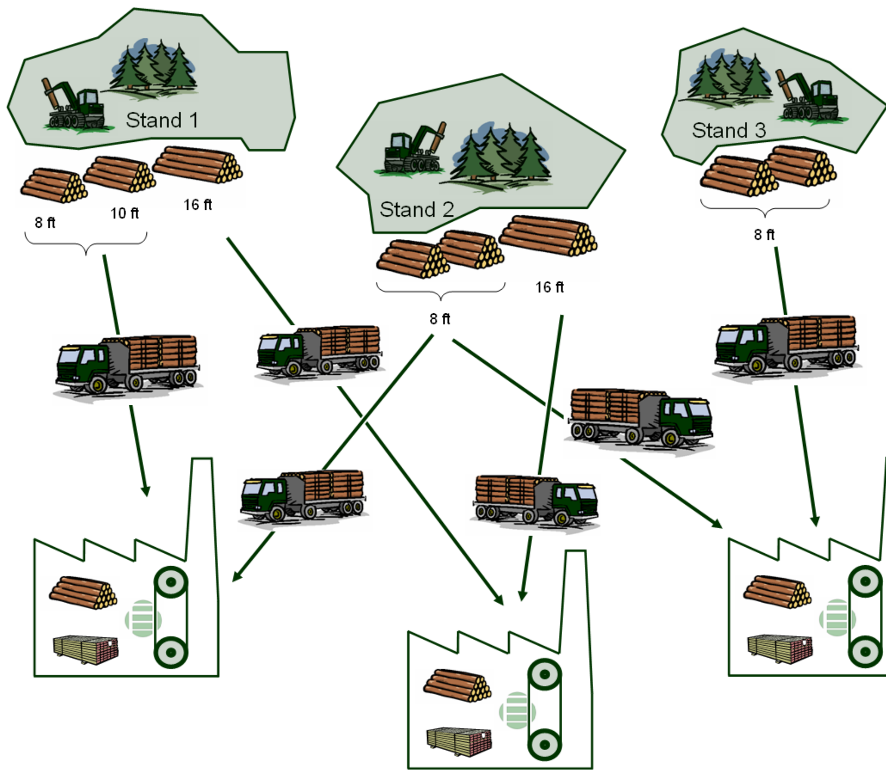

Figure 1: Procurement problem considered

This problem formally consider a set of forest stands $f \in F$ and a set of mills $s \in S$ procured from any of the stands f. All stands are managed by a single forest company which acts as the forest operation manager responsible coordinating the operations. We also assume that all stands f can be harvested during the period considered. In other words, stands are accessible by a network of roads. Furthermore, we do not consider the cost of moving harvesting equipments from one stand to another. We consider indeed that all stands have been selected at the tactical planning level taking moving costs into account, as suggested by Beaudoin et al. (2007). We also consider that the entire stand must be harvested. The decision to be taken is thus to select the log type and volume to be harvested in each stand, taking into account the demand for various log types of each mill, the distance between the mills and the stands, the stem class distribution in each stand, and bucking feasibility constraints.

The process of log procurement can be decomposed into several operations, including felling, bucking, sorting, hauling to roadside, and transportation to mills. In a cut-to-length harvesting system, once a tree is cut, its bucking is carried out with the same equipment and can produce several types of log that are sorted directly at the stump. Consequently, for each different log type (e.g., 8 ft, 10 ft, 12 ft) that is cross-cut, sorting logs forces the boom of the harvester to move over a pile of the appropriate log type in order to avoid mixing the logs. This process of moving the boom creates a discontinuity in the bucking process that, in turn, creates a discontinuous ineffectiveness that increases the production cost in a stepwise manner for each introduction of a new log type to produce (Gingras and Favreau 2002). Next, these piles of different logs are hauled to roadside and piled up in different locations where they are stored until they are loaded on log trucks. Hauling to roadside is also less efficient as it is constrained to consider each log type separately.

One consequence of this cut-to-length process is that producing a large number of log types in the same stand increases the workload of the forwarder and decreases the productivity of the harvester in a stepwise manner. In the model proposed in this paper, the costs of felling, bucking, sorting and hauling to roadside are aggregated into a non-linear production cost function. This procurement problem is thus non-linear since the objective function changes as the number and type of logs harvested in each stands change.

Furthermore, eastern Canadian forests are natural and several species and age groups are typical. Consequently, each stand may have characteristics that make them fit for particular log types or combination of log types, while they may not be appropriate for others. Because mills are similarly different in terms of production capabilities, which make them tuned for specific log types, procurement decision must account for both specificities, as well as transportation cost. The considered problem is thus the minimization of the overall procurement cost, subject to demand satisfaction and bucking constraints. Forest level bucking is usually modeled in the literature as profit maximization decision problems because demand is defined in terms of minimum and maximum quantities to produce, between which the value produced must be maximize (Laroze 1999; Arce et al. 2002). Stands are characterized by the variety of trees of different species and taper profiles.

The overall objective is to minimize production cost which is the sum of harvesting and product sorting costs and the transportation costs.

3.2 Problem decomposition and modelling

Because in this context, both procurement planning and bucking problems are complex combinatorial optimization problems, we propose a modelling approach based on Schneeweiss (2003) hierarchical modelling framework of decision problems. This approach results in the modelling and solving of two interdependent problems. In this paper, we propose algorithms and procedures to optimize and combine these problems as previously introduced in Chauhan et al. (2005).

First, the general modelling framework is introduced. Then, we present the model of the procurement planning problem. The formulation of the bucking problem follows.

3.2.1 General modelling framework

As initially introduced in Chauhan et al. (2005), the formulation of a general approach that takes bucking decisions into account in procurement planning is outlined in

Figure 2. In brief, instead of designing a single centralized decision model that simultaneously optimize procurement and bucking decisions, we used some of the concepts and ideas of the distributed modelling framework introduced in Schneeweiss (2003). In this framework, both problems are modelled separately and solved iteratively until no improvement can be found.

The procurement problem, also referred to as the main problem, is first solved considering the productivity parameters of a few bucking patterns. For instance, these patterns can be calculated by solving the bucking problem with approximate resource (i.e., log type) cost. Then, for the current procurement solution, the dual revenue of harvesting each log type is input in the bucking problem, also referred to as the sub-problem, in order to identify for each stand independently new bucking patterns that can improve the procurement solution. Then, for each of these new patterns (if any), their productivity parameters (i.e, quantity of log type in a given pattern applied in a given stand) are calculated and then input into the procurement planning problem for another solving cycle.

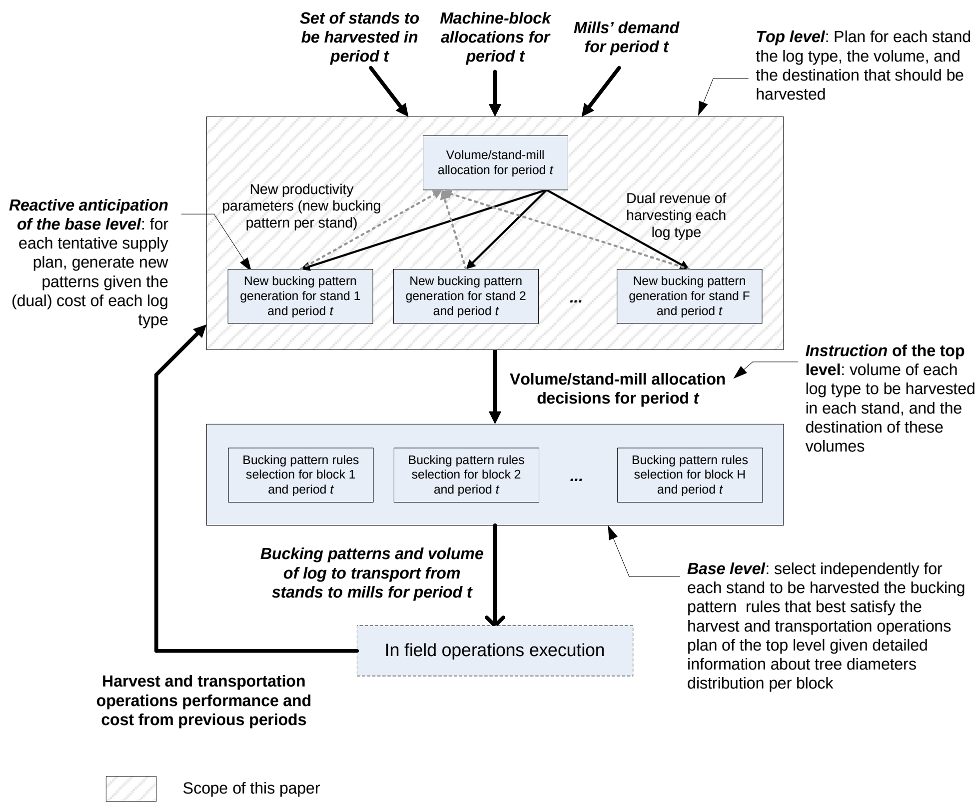

Figure 2: General modelling approach

This approach allows the system to generate new bucking patterns that can improve the procurement solution, instead of considering all possible bucking patterns, which would not be realistic given the combinatorial complexity involved (number of possible patterns x the number of stand).

Furthermore, this approach is not designed to generate both a procurement plan and a set of bucking rules. It is only concerned by the definition of a feasible procurement plan. Indeed, in Chauhan et al. (2005), only aggregated information about stands volume availability was considered. No detailed information about tree diameter and species distribution was considered, which could render the procurement plan infeasible. This extension of this early work proposes to take this type of information into account. However, the full complexity of the bucking optimization has not been addressed in order to simplify the problem. In other words, instead of considering all the complexity of how bucking is actually carried out in the field, the proposed formulation of the bucking optimization problem is a simplified anticipation of how bucking is carried out. Although, it does not represent perfectly bucking activities, it still allows the forest manager to take bucking into account when planning the procurement of several mills from several stands.

3.2.2 Procurement planning model

In the mathematical formulation of this problem, we assume that we have information for all possible patterns applicable to any stand. As said in the previous section, although demand is expressed in terms of volume of logs and minimum small end diameter (i.e., SED) requirement for each log, we ignore this last detail in the formulation of the procurement planning problem. Indeed, we assumed that the patterns generated by solving the bucking problem for each stand will provide logs that meet the required specifications. Similarly, at this level, we ignore the details regarding tree diameter and species distribution.

Notations

S Set of mills

F Set of stands

P Set of log types

$Q_{s,p}$ Quantity demanded by sawmill s for log type p.

$t_{f,d}$ Tree density for diameter group d in stand f.

$c_{n,f}$ Harvesting cost per product if n products types are cut in stand f ($/m^3).

χ_p Metric volume of one log of type p (m^3).

$q_{l,f,p}$ Quantity of log type p in pattern l applied in stand f (no dimension).

Z_f Estimated supply availability of stand f (m^3).

$m_{f,s,p}$ Transportation cost ($\$/m^3$).

z Average cost of a tree ($\$/m^3$).

$\omega_{s,p}$ Penalty or outsourcing cost for unsatisfied demand ($\$/m^3$).

$r_{l,f}^n$ Frequency of use of pattern l if n products are harvested in stand f.

$x_{f,s,p}$ Number of pieces of log type p transported from stand f to mill s

$\eta_{s,p}$ Volume of log type p sent to mill s from an alternative source (m^3).

I_f^n Binary variable taking value 1 if n log types are harvested in stand f

0 otherwise.

$J_{f,p}$ Binary variable taking value 1 if log type p is harvested in stand f

0 otherwise.

B Big number.

P1 Min
$$\sum_{f\in F}\sum_{n=0}^{N}\left[c_{n,f}.\sum_{l\in L}\sum_{p\in P}r_{l,f}^n.q_{l,f,p}.\chi_p\right]+\sum_{f\in F}\sum_{s\in S}\sum_{p\in P}m_{f,s,p}.x_{f,s,p}.\chi_p+\sum_{s\in S}\sum_{p\in P}\omega_{s,p}.\eta_{s,p}$$
$$+z.\sum_{n=0}^{N}\sum_{f\in F}\sum_{l\in L}r_{l,f}^n$$

Subject to

$$\sum_{l\in L}r_{l.f}^n.q_{l,f,p}\geq\sum_{s\in S}x_{f,s,p}\qquad\forall p\in P,\ f\in F,n=1,...,N\qquad(1)$$

$$\sum_{f\in F}\chi_p.x_{f,s,p}+\eta_{s,p}=Q_{s,p}\quad\forall\ s\in S,p\in P\qquad(2)$$

$$\sum_{n=1}^{N}\sum_{p\in P}\sum_{l\in L}\chi_p.q_{l,f,p}.r_{l,f}^n\leq Z_f\quad\forall\ f\in F\qquad(3)$$

$$\sum_{l\in L}\sum_{p\in P}q_{l,f,p}.r_{l,f}^n\leq B.I_f^n\quad\forall\ f\in F,\forall\ p\in P,n=1,...,N\qquad(4)$$

$$\sum_{n=1}^{N}\sum_{l\in L} q_{l,f,p}.r_{l,f}^{n} \leq B.J_{f,p} \quad \forall\, f\in F, p\in P \tag{5}$$

$$\sum_{p\in P} J_{f,p} \leq \sum_{n=1}^{N} n.I_{f}^{n} \quad \forall\, f\in F \tag{6}$$

$$\sum_{n=0}^{N} I_{f}^{n} = 1 \quad \forall f\in F \tag{7}$$

$$I_{f}^{n} \in \{0,1\} \ \forall f\in F, n=1,\ldots,N \tag{8}$$

$$J_{f,p} \in \{0,1\} \ \forall\, f\in F, p\in P \tag{9}$$

The objective of P1 is to minimize the harvesting, transportation, outsourcing costs as well as the cost of the trees harvested. Outsourcing cost serves as penalty in case there is no solution to the problem due to insufficient supply availability. In the formulation, we did not use the cost of trees since this cost is defined by the volume demanded by mills (which is a parameter) and a cost per unit volume of tree. We assume that this unit cost is similar for all stands, which is true if stands are not geographically distributed throughout large region.

Constraints (1) show that any quantity shipped from a production site to a mill is bounded by the production in the corresponding stand. Constraints (2) depict that the total supplies to any sawmill from all sources should match the mill's demand. In order to maintain feasibility, we have introduced variables $\eta_{s,p}$ in case supply availability is not sufficient or particular product specification cannot be produced within the given stands characteristics. Constraints (3) make sure that the volumes harvested in each stand do not exceed the aggregated available volume in these stands. Constraints (4), together with constraints (7), make sure that only one set of harvesting variables is used in order to use the appropriate harvest cost function in the objective function. Then constraints (5) and (6) are used to count the number of log types harvested in each stand. Finally, constraints (8) and (9) are used to force variables I and J to take values 1 or 0.

As mentioned before, the information regarding tree profiles, forest configuration and product/log specifications are not included in the above formulation for simplicity. These details are only needed the generation for of bucking patterns. In the above formulation, these patterns are formalized through the setting of the productivity parameters $q_{l,f,p}$. The formulation of the bucking problem follows.

3.2.3 Bucking optimization model

The general modelling framework presented previously proposes a simplification of the actual approach to bucking. More specifically, instead of considering the use of specific bucking patterns for each class of tree diameters, which is a common approach in Quebec, we consider that bucking is carried out in a bulk process (referred to as bulk bucking in the remainder of this paper). In other words, and for a given stand, the productivity of each pattern is calculated as the weighted average of the productivity resulting from the application of one pattern to all classes of trees considering their relative important in terms of frequency. For example, if for class A, pattern 1 produces 2 logs of 8 ft and 3 logs of 12 ft, while it produces 3 logs of 8 ft and 3 logs of 12 ft for class B, the productivity of pattern 1 for that stand is 2,5 logs of 8 ft and 3 logs of 12 ft (if there are as much trees in both classes).

Next, in order to take the heterogeneous nature of forest into account, statistical models of each stand are built using inventory data. To do so, trees are classified based on their species, diameters, as well as their heights. For each stand, the frequency (i.e., number of trees per class) and the geographical distribution are then input into a geographical information system and use in the optimization process. Finally, these inventory data are also exploited to define mathematical relationships (i.e., tapper function), for each species and location, between tree diameter and any height from the base of the tree. This information is required to know if a log bucked at a certain height of the tree will meet the SED requirement.

As explained in the general modelling framework, bucking optimization is carried out for each stand independently using this detailed information about the characteristics of stands and trees.

Also, we consider that stands are defined according to, among others, their homogeneity. In other words, within each stand, tree classes are uniformly mixed.

This model of pattern generation aims at maximizing the revenue from harvesting a particular stand, using a bucking pattern. Because bucking optimization is solved for each stand, we drop the indice f in the notation. The following additional notations are used for this problem:

T	Set of tree class (diameter class).
π_τ	Fraction of trees with class $\tau \in \mathrm{T}$ in a given stand ($\pi_\tau \in [0,1]$).
H_τ	Maximum height of tree of class $\tau \in \mathrm{T}$.
$H_{\tau(1)}$	Height of the tallest tree available in any diameter groups.
y_p	Dual cost corresponding to constraint (2) (which can be interpreted as the current price of log type p based on the available patterns in (P1)).
h_p	Length of log type p.
$h_{\tau,p}$	Maximum length from base to which log type p is available in class $\tau \in \mathrm{T}$.
k	Position of a cut from the tree base.
$m_{p,\tau,k}$	Binary variable taking value 1 if a log of type p is bucked in the k^{th} position in a tree of class τ, 0 otherwise.
$G_{p,k}$	Binary variable taking value 1 if a log of type p is bucked in the k^{th} position, regardless of the class, 0 otherwise.
$\delta^+_{\tau,k}$	Slack variable defining the remaining length of tree after the (k-1)$^{\text{th}}$ cut.

P2 $$\text{Max} \sum_{p\in P} y_p .(\sum_{\tau\in \mathrm{T}} \sum_{k=1}^{K} \pi_\tau .m_{p,\tau,k}) - z$$

Subject to

$$\sum_{p\in P} G_{p,k} \le 1 \quad \forall\, k = 1,2,...,K \tag{10}$$

$$\sum_{p\in P} G_{p,k} - \sum_{p\in P} G_{p,k+1} \geq 0 \quad k = 1,2,...K-1 \qquad (11)$$

$$\sum_{i=0}^{k-1}\sum_{p\in P} h_p.G_{p,i} + \delta^{+}_{\tau,k} = H_\tau \quad \forall \tau \in \mathrm{T}, k = 1,2,...,K \qquad (12)$$

$$H_\tau - \delta^{+}_{\tau,k} + m_{p,\tau,k}.h_p \leq h_{\tau,p} \quad \forall p \in P, \tau \in \mathrm{T}, k = 1,2,...,K \qquad (13)$$

$$m_{p,\tau,k} \leq G_{p,k} \quad \forall p \in P, \tau \in \mathrm{T}, k = 1,2,...,K \qquad (14)$$

$$m_{p,\tau,k} \in \{0,1\} \qquad (15)$$

The objective function of P2 aims at maximizing to the revenue per tree generated from using one bucking pattern applied to the considered stand, minus the average cost of a tree. Next, constraints (10) ensure that position k in a given pattern should not be occupied by more than one log type. (11) is used to make sure that there should not be any unoccupied position between two occupied positions in the pattern. Constraints (12) calculate $\delta^{+}_{\tau,k}$ the remaining length of tree after the (k-1)th cut. Constraints (13) check if any log at k^{th} position meets the corresponding SED and length requirements. Constraints (14) make sure that a log type is cut on a tree of class τ only if it meets constraints (10) to (13).

4. Solution approach

The section proposes algorithms to solve both of these problems. First, the procurement problem is addressed. A column generation approach is proposed to solve. Next, a heuristic is proposed to solve the bucking problem.

4.1 Procurement planning optimization

This problem, as described previously, is complex and cannot be tackled by commercial software using general MIP formulation. In fact, it is difficult to find feasible solution, for several instances, even in the case we generate all possible patterns and load into the system. There are indeed a huge number of possible patterns since trees are conical and each stand is different in terms of distribution of classes of diameter. In this work, we present an approach based on column generation to solve the above problem. First, we present an alternative formulation for the main problem and then discuss the solution approach.

4.1.1 Alternative formulation

Let $X_{i,j}^{n} = \{x_{i,j,1,1}^{n}, x_{i,j,1,2}^{n}, ..., x_{i,j,1,p}^{n}, ..., x_{i,j,s,p}^{n}\}$ be the i^{th} feasible production vector for the stand $j \in F$ in the production scenario n. Production scenario n means that n different types of logs are produced in stand $j \in F$. Other elements of this vector denote the quantity of particular log transported to a particular mill. Indexes have the same meaning as in the previous notation. Let $\lambda_{i,j}^{n} \in \Re^{+}$ be the variable associated with the vector $X_{i,j}^{n}$. Now problem (P1) can be reformulated as follows:

P1-2 Min $\sum_{j \in F} \sum_{n=1}^{N} \sum_{i=1}^{K_j} C_{i,j}^{n} . \lambda_{i,j}^{n} + \sum_{s \in S} \sum_{p \in P} \omega_{s,p} . \eta_{s,p}$

Subject to

$$\sum_{j \in F} \sum_{n=1}^{N} \sum_{i=1}^{K_j} \lambda_{i,j}^{n} . x_{i,j,s,p}^{n} + \eta_{s,p} = Q_{s,p} \quad \forall s \in S, p \in P \qquad (16)$$

$$\lambda_{i,j}^{n} \le 1 \quad \forall j \in F, i \in K_j, n = 1,2,..., N \qquad (17)$$

$$\lambda_{i,j}^{n} \le M \alpha_{j}^{n} \quad \forall j \in F, n = 1,2,..., N \qquad (18)$$

$$\sum_{n=1}^{N} \alpha_{j}^{n} \le 1 \qquad \forall j \in F \qquad (19)$$

$$\alpha_{j}^{n} \in \{0,1\} \qquad \forall j \in F, n = 1,2,..., N \qquad (20)$$

The objective function of this alternative formulation is basically similar to that of (P1). $C_{i,j}^{n}$ represent the cost vector associate to $X_{i,j}^{n}$. This cost vector consists of transportation and harvesting cost calculated considering the production scenario of the stand and quantity transported to the corresponding mills. Constraints (16) insure that demand of each mill must be satisfied.

Again, we include $\eta_{s,p}$ in the formulation in order to identify the product and corresponding quantities which cannot be met by the available volumes of the stands. Constraints (17) guaranties that a set of columns satisfying the harvesting scenario must form a feasible solution. Constraints (18) identify the production scenario in each stand, and constraints (19) ensure that at most one cost scenario associated to each stand should be considered. Note that in a feasible solution, we need only one harvesting scenario in each stand. The importance of this constraint will be discussed later. Finally last constraints (20) show that variables α_j^n are binary.

In this formulation, we have neglected the stand capacity in terms of volume available. However, constraints (17) indirectly control this capacity and our solution to problem P1-2 never violates the capacity constraint even when we have several vectors correspond to the same stand. This will be clearer later in the section.

In order to solve the problem, we need to generate all production allocations $X_{i,j}^n$. Note that to generate all possible sets, we first need to generate all possible patterns. The set that corresponds to all feasible allocations could be very large and may be very difficult to obtain a priori. The advantage of using column-generation based decomposition approach is that we can start the algorithm with a few feasible columns and generate other appropriate columns dynamically using the latest scenario. In other words, the advantage of this approach is that we may not need all possible columns to find the optimal solution. Furthermore, the required good columns can be generated by solving a small problem known as pricing problem.

The new problem with a few required columns is known as the restricted master problem (i.e., RMP) is denoted by (P1-2(RMP). Furthermore, problem (P1-2) is an integer program, and constraints (12)-(14) only make sure that the final solution is feasible. In the column generation approach we can drop these constraints as we can check dynamically if the solution is integer or not. In brief, we solve the relaxed version of (P1-2(RMP)), which only contains constraints (16) and (17). Hence, a column consists of a production allocation vector $X_{i,j}^n$ and an indicative vector for the stand, which is a 0-1 vector of F dimensions with all zeros except j^{th} place (constraint (17) correspond to this vector).

Now the algorithm used to solve the procurement problem has two main parts. Solving iteratively a pricing problem in order to add new columns and then branching the solution in order to obtain a feasible solution and bound.

4.1.2 Pricing problem

As stated earlier, (P1-2(RMP)) contains only a few columns which may not guarantee the optimal solution. In order to identify other columns, we solve pricing problems until we have no more new columns which can improve the solution. The pricing problem also takes cares of stand capacity constraints, product length, diameter, tree profiles and distribution in the stand through the embedded solving of the bucking problem.

As mentioned before this pricing problem generates a column which represents the information about the production and transportation allocations to various mills. Since production in any stand depends upon the combination of log type to harvest, we cut using different patterns, and moreover we are restricted by capacity constraints, it is mandatory to include pattern generation problem (2) in the pricing problem.

Let $\mu_{s,p}$ and ϑ_j be the dual variables correspond to constraints (16) and (17) respectively. Let $V_{f,i}$ denotes the i^{th} pattern applicable in stand f. Now the pricing problem can be written as follows:

P3(f) Min $(m_{f,s,p} - \mu_{s,p}).x_{f,s,p} + \sum_{n=1}^{N} \beta_n.c_{n,f}.x_{f,s,p} - z.r_{l,f} - \vartheta_j$

Subject to

$$\sum_{i \in \Gamma_f} r_{f,i}.q_{l,f,p} \geq \sum_{s \in S} x_{s,p} \quad \forall p \in P \tag{21}$$

$$\chi_p.x_{f,s,p} \leq Q_{s,p} \quad \forall s \in S, p \in P \tag{22}$$

$$\sum_{i \in \Gamma_f} q_{l,f,p}.r_{f,i} \le Z_f \qquad \forall j \in F \tag{23}$$

$$\sum_{s \in S} x_{s,p} \le M\alpha_p \tag{24}$$

$$\sum_{n=1}^{N} n.\beta_n = \sum_{p=1}^{N} \alpha_p \tag{25}$$

$$\alpha_p, \beta_n \in \{0,1\} \tag{26}$$

The goal of the pricing problem is to find an allocation in any stand f which minimizes the above objective function. Since the pricing problem is again solved with column generation, a starting feasible solution is necessary. Constraints (21) guarantees that a product allocation vector given by pricing algorithm respects the capacity constraints and has a associated set of patterns which will obtain it. Constraints (22) restrict the product allocation by the maximum demand. Constraints (23) restrict the number of harvested tree by the stand capacity. Finally (24), (25) and (26) are used to select appropriate harvesting/production scenario in the stand.

Reduced cost= $(m_{f,s,p} - \mu_{s,p}).x_{s,p} + \sum_{n=1}^{N} \beta_n.c_{n,f}.x_{s,p} - z.r_{l,f}$

If the reduced cost is negative then we add the new solution of the pricing problem (i.e., the new column) in (P1-2(RMP), otherwise we conclude that we have obtained the optimal solution of the relaxed problem. We can see that the pricing problem is an integer programming problem and solving it up to optimality is costly in terms of computing resources and time. Furthermore, this problem has to be solved for each stand, in each iteration. The effectiveness of the column generation approach depends upon how effectively we solve the pricing problem. In the next section, we present a fast heuristic approach which, in most of the cases, comes up with an optimal solution.

4.1.3 Heuristic algorithm for the pricing problem

The pricing problem is difficult because of there are several possible harvesting scenarios. In the algorithm, we start with a scenario and look if we can improve the solution by switching to the next or more advanced scenario, otherwise we stop the algorithm and calculate the cost corresponding to the latest harvesting scenario. We can see from the pricing formulation that once we fixed the scenario, constraints (18) to (20) can be dropped. Let us define new problem (P3R(f , n)) where $f \in F$ and n is production scenario (i.e., n log types harvested). The algorithm to solve (P3(f)) can be expressed as follows:

1. Set $n = 1$, Set $BestCost = BigNumber$
2. Solve $\text{P3R}(f, n)$ using CG approach and the solving of (P2).
3. $\text{NewCost=GetRealCost(P3R}(f, n))$.
   (GetRealCost is a function that calculates the real cost based on using a real production scenario.)
4. if Real_n=GetReal_n($P_3R(f, n)$).
   (GetReal_n is a function that computes the number of product types produced in the forest while setting was n.)
5. If $\text{Real\_n} \le n$ Stop.
6. If $NewCost \le BestCost$ then
   $BestCost = NewCost$.
   $n = n + 1$.
   Go to 2.
7. Stop.

During each iteration, we calculate the real cost corresponding to the current scenario and always keep the best cost (and hence its corresponding allocation vector). At the termination, we select the best solution kept by the algorithm. If $\text{Real\_n} = n$ then the solution is optimal.

We use column generation approach to solve the pricing problem. In other words, we have to generate patterns for the pricing problem by solving (P2) with in algorithm described later in this paper.

The column generation algorithm terminates for problem (P1-2(RMP) once there is no column with negative reduced cost. The solution may not be feasible for the global problem because it may contain more than one column associated with each stand. In order to obtain a feasible solution, and then optimal solution, we use the branch-and-bound approach. The following section explains the branching technique used in the algorithm.

4.1.4 Branching scheme

Problem (P1-2(RMP)) is the relaxed master problem in which it is possible that α variables can take non-integer values. A simple way to branch would be to set α to 0 in one branch and 1 in the other. However, this may complicate the branch-and-bound tree because setting $\alpha = 1$ means we always want keep n number of products, (n =1,…,N) in the right branch. These n products can be chosen in ${}^{N}C_{n}$ ways. This simply means we need to solve the pricing problem either a maximum of this much of times to get all the columns with n number of products with negative reduced cost, or use integer programming approach to solve the pricing problem. Similarly in the left branch $\alpha = 0$ means we do not want n number of products in the column which could be possible in $2^{n} - {}^{N}C_{n} - 1$ ways. Consequently, although this branching scheme is rather simple to implement, it is very difficult to control as the branch-and-price tree becomes very big.

In the proposed branching scheme, we select a product, not already selected, whose contribution is minimal in the selected stand. We keep it in the right branch. Note that keeping the product in the right branch does not mean that we need to have a positive production for that product in all columns.

Let B_f be a set of basic columns correspond to stand f and let $l_f = |B_f|$. Let also assume that the production cost for at least one column does not correspond to the production cost associated with the rest of the columns. Now we compute :

$$b_{f,i,p} = 1 \quad if\ x_{i,j,s,p} > 0 \quad \forall i \in B_f, p \in P, s \in S, n = 1,..., N$$

$$b_{f,i,p} = 0\ otherwise$$

and we define:

$$w_{f,p} = l_f - \sum_{i \in B_f} b_{f,i,p}$$

We select the product and the stand for branching which has the highest $w_{f,p}$ value. In the case of ties (i.e., more than two products with the same $w_{f,p}$ in stand f or with the same product in two different stand), we define for each tied product, say p_1 and stand f_1, the relative production contribution :

$$rp_{f_1,p_1} = \frac{\sum_{i\in B}\sum_{s\in S}\sum_{n=1}^{N}\lambda_{i,f}^{n}.x_{i,f_1,s,p_1}}{\sum_{i\in B}\sum_{p\in P}\sum_{s\in S}\sum_{n=1}^{N}\lambda_{i,f}^{n}x_{i,f,s,p}}$$

and select the stand and product with the smallest rp_{f_1,p_1} .

The algorithm terminates when all the columns corresponding to each stand have the same production scenario as the number of log type harvested in the respective stand.

4.2 Bucking optimization

The pattern generation pricing problem is a binary integer programming problem. Since we solve this pricing problem for each stand in each iteration, the use of a commercial IP solver may slow the overall column generation algorithm. At the stand level, in practice, we do not harvest more than 4 to 6 products. Consequently, we can develop a fast optimal algorithm for the pricing problem that does not involve the use of IP commercial solver.

Algorithm

Define $stages = H_{\tau(1)}$. (*We assume that trees length is integer, otherwise we divide* $H_{\tau(1)}$ *by an appropriate number so that we have an integer number of equal length pieces)*

Define $c_\tau(p, Lb) = 1$ if $Lb + h_p \le h_{\tau,p}$, otherwise 0 (*we can generate* $h_{t,p}$ *using the statistical relationship for the stand and species mentioned previously)*

Define $F(I) = 0 \quad \forall I \le 0$.

Initialize $F(I) = 0$ for $I = 1$ to $stages$, $h_0 = 0$.

1. For $I = 1 \;\; to \;\; stages$
   Set $x_{\tau} = Max(0, H_t - H_{\tau(1)} + I) \;\; for \;\; \tau \in \mathrm{T}$.
   Set $Lb_{\tau} = Min(H_{\tau(1)} - I, H_{\tau}) \;\; for \;\; \tau \in \mathrm{T}$.
   Set $F(I) = Max_{\{0 \le p \le P\}} \left\{ y_p \sum_{\tau \in \mathrm{T}} \pi_{\tau} . c_{\tau}(p, Lb_{\tau}) + F(I - h_p) \right\}$
2. Stop.

The above algorithm is based on backward dynamic programming, in which $F(stages) - 1$ will be the objective function of (P2). The intermediate stages, which maximize $F(stages)$ in step 1.3, will also give the position of the different logs from the tree base. Finally, the new column is constructed by multiplying the resulting productivity factors by π_t for each tree group using the function c_{τ} as before.

5. Numerical experimentation

We have tested the above approach on several randomly generated instances. The numerical results of this section are divided in two parts. In the first part, we present the numerical experiments carried out for testing the performance of pricing algorithm. Note that we are not using IP solver to solve the pricing problem which is an integer program. In order to test the performance of the pricing heuristic we have generated several problem instances randomly and solved them using the proposed algorithm.

The objective value given by the algorithm is compared w.r.t. the exact solution. In order to get the exact solution, we solve $2^P - 1$ problems by exploring all possibilities. The algorithm performance is presented in table 1. The first column in the table shows the problem size. For each problem size, we have solved 10 different instances. The mean relative error is presented in the second column. Standard deviation of the mean error is presented in the third column. Finally in the last column, we present the relative time gain when using heuristic algorithm over the exact approach.

From the numerical testing it is evident that pricing heuristic propose a near solution optimal, at least for the problem size we face at the stand level. It takes only fraction of the time required for the exact solution. Furthermore this gain is more significant for relatively big problem size.

Table 2 presents the algorithm performance for various stand and log type combinations for three mills. Table 3 presents the algorithm performance for various stands and mills for four log types. In all cases, we have used 4 classes of tree diameter.

Table 1: Performance of pricing algorithm

Problem size (P, S)	Mean fraction error	Standard deviation of mean fraction error*	Time gain (%)
2,2	0.0001	0.0	89.66
2,3	0.0001	0.0	76.5
2,4	0.0001	0.0	87
2,5	0.0001	0.0	87
3.2	0.0016	0.0	51.14
3,3	0.0001	0.0	65.07
3,4	0.0001	0.0	54.46
3,5	0.0001	0.0	56.066
4,2	0.0007	0.0	30.35
4,3	0.0012	0.0	37.73
4,4	0.05	0.0016	20.477
4,5	0.0001	0.0149	44.01
5,2	0.0001	0.0	29.33
5,3	0.001	0.0	24.63
5,4	0.001	0.0	32.32
5,5	0.012	0.0	27.53
6,2	0.012	0.0	7.8
6,3	0.005	0.006	19.63
6,4	0.0025	0.0	18.61
6,5	0.00229	0.0	11.08
7,2	0.0049	0.0	13.39
7,3	0.0017	0.0006	12.47

7,4	0.0141	0.0017	3.30
7,5	0.001	0.00358	6.85

On average, the algorithm completion time increases relatively as size of the problem increases, except for some examples where one or two random problems have taken much longer than the other problems of the same group. That is why the standard deviation is high for that test group. As we increase the number of stand, the algorithm time increases because there are more pricing problems to solve (one for each stand).

Table 2: Performance forest vs Product Types

Log types Stands	2	3	4	5	6
2	0.51(0.17)	2.21(1.28)	2.59 (1.28)	2.37(0.92)	2.34 (1.39)
3	1.41 (0.93)	2.27(1.54)	5.53(7.37)	4.1(1.43)	3.01(3.5)
4	1.43(0.77)	2.33(0.76)	3.45(1.16)	4.48(2.67)	4.49(2.25)
5	1.47(0.57)	2.69(1.79)	4.29(2.71)	19.96(24.48)	7.24(5.3)
6	2.52(1.76)	4.6(2.27)	7.0(5.07)	8.21(3.23)	12.7(9.29)

Table 3: Performance Forest Vs SawMills

MiIls Stands	2	3	4	5
5	3.8(1.74)	4.35(2.99)	7.02(2.99)	5.3(1.06)
6	10.46(10.22)	5.85(4.58)	5.32(1.34)	7.19(2.46)
7	5.69(4.77)	16.12(19.75)	10.30(6.98)	15.18(14.56)
8	6.37(3.63)	5.25(1.8)	10.62(4.35)	11.8(2.32)
9	6.51(3.01)	8.3(2.06)	14.6(13.2)	13.21(2.16)
10	9.13(3.45)	11.12(5.57)	8.71(3.0)	11.38(4.47)

The algorithm was implemented in C++ using Visual C++.net on a desktop computer with 1.7 Ghz Pentium 4 processor with 512MB of RAM. For the Linear programming part, we use CPLEX 9.1 which interacted with the algorithm as needed.

6. Conclusion

In this paper, we have proposed an application of Operations Research techniques to a forest management problem. This problem involves the planning of harvest and transportation operations in a multi-stand and multi-mill context. In order to produce a plan that takes stands and trees characteristics, as well as the length of product demanded by mills into account, we propose a mechanism that anticipates bucking operations. The proposed algorithm is tested on the randomly data set which is generated around the standard tree specification available in the literature. The algorithm presented in this paper is easy to implement on any PC and are capable of solving forest level (size) problems in reasonable time.

Several extensions and improvement of this work include the development of a more realistic anticipation of bucking operations. Another interesting research direction would be to study the impact of harvest and transportation plans on the limited availability of the fleet of trucks used to haul the logs to the mills. This would allows forest managers to also take transportation capacity into account when planning procurement.

Acknowledgement

This work was funded by the FOR@C Research Consortium in E-Business in the Forest Products Industry. The authors would like to thanks Jean Favreau and Joseph nader of the Forest Engineering Research Institute of Canada for their support..

References

Arce, J.E., Carnieri, C., Sanquetta, C.R., and Filho, A.F. (2002). A forest-level Bucking optimization system that considers customer's demand and transportation costs. Forest Science 48(3): 492-503.

Beaudoin, D., LeBel, L., and Frayret, J.M. (2007). Tactical supply chain planning in the forest products industry through optimisation and scenario-based analysis. Canadian Journal of Forest Research 37(1): 128-140.

Brunberg, T., and Arlinger, J. (2001). Vad kostar det att sortera virket i skogen? (what does it cost to sort timber at the stump?). Resultat, Skogforsk 3.

Carlsson, D., and Rönnqvist, M. (2005). Supply chain management in forestry—case studies at Södra Cell AB. European Journal of Operational Research 163(3): 589-616.

Chauhan, S.S., Frayret, J.-M., and LeBel, L.G. (2005). Multi-commodity supply network planning in the forest supply chain, . Working paper, Centre CENTOR, Université Laval DT-2005-JMF-2.

Eng, G., and Daellenbach, H.G. (1985). Forest outturn optimization by Dantzig-wolfe decomposition and dynamic programming column generation. Operations Research 33(2): 459-464.

Faaland, B., and Briggs, D. (1984). Log bucking and lumber manufacturing using dynamic programming. Management Science 30(2): 245-257.

Forintek Canada, F. (2005). Efficacité du tronçonnage a la scierie et en foret a l'aide de multifonctionnelles. Technote Forintek June.

Geerts, J.M.P., and Twaddle, A.A. (1984). A method to assess log value loss caused by cross-cutting practice on the skidsite. New Zealand Journal of Forestry 29(2): 173-184.

Gingras, J.-F., and Favreau, J. (2002). Incidence du triage sur la productivité des systémes par bois tronçonnés. Avantage, Feric 3.

Kivinen, V.P. (2004). A genetic algorithm approach to tree bucking optimization. Forest Science 50(5): 696-710.

Kivinen, V.P. (2006). A forest-level genetic algorithm based control system for generating stand-specific log demand distributions. Canadian Journal of Forest Research 36(7): 1705-1722.

Kivinen, V.P., and Uusitalo, J. (2002). Applying fuzzy logic to tree bucking control. Forest Science 48(4): 673-684.

Laroze, A. (1999). A Linear Programming, Tabu Search method for solving forest-level bucking optimization problems. Forest Science 45(1): 108-116.

Laroze, A.J., and Greber, B.J. (1997). Using Tabu Search to generate stand-level, rule-based bucking patterns. Forest Science 43(2): 157-169.

Maness, T., and Adams, D. (1991). The Combined Optimization of Log Bucking and Sawing Strategies. Wood and Fiber Science 23(2): 296-314.

Marshall, H.D., Murphy, G., and Boston, K. (2006). Three mathematical models for bucking-to-order. Silva Fennica 40(1): 127–142.

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

Murphy, G., Marshall, H., and Bolding, M.C. (2004). Adaptive control of bucking on harvesters to meet order book constraints. Forest Products Journal 54(12): 114-121.

Pnevmaticos, S.M., and Mann, S.H. (1972). Dynamic programming in tree bucking. Forest Products Journal 22(2): 26–30.

Schneeweiss, C. (2003). Distributed Decision Making. Springer-Verlag, New York.

Sessions, J., Boston, K., Hill, R., and Stewart, R. (2005). Log sorting location decisions under uncertainty. Forest Products Journal 55(12): 53-57.

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