Multi-commodity supply network planning in the forest supply chain

Satyaveer S. Chauhan, Jean-Marc Frayret, Luc LeBel

April 2005

Working Paper DT-2005-JMF-2

Research Consortium in e-Business in the Forest Products Industry (FOR@C)

Network Organization Technology Research Center (CENTOR),

Université Laval, Québec, Canada

Multi-commodity supply network planning in the forest supply chain

Satyaveer S. Chauhan[*], Jean-Marc Frayret [†‡], Luc LeBel[‡]

April 2005

Abstract

We consider in this paper a two echelon timber procurement system in which the first echelon consists of multiple harvesting blocks and the second echelon consists of multiple mills (e.g., sawmills), both distributed geographically. Demand is put forward by mills in the form of volumes of logs of specific length and species. Due to the impact of log handling and sorting on cut-to-length harvester and forwarder productivity ((15)), the total harvesting cost function increases non-linearly as the number of product harvested per block increases. The overall decision problem is a large scale mixed integer programming problem with the objective of minimizing combined harvesting and aggregated transportation costs, while satisfying demand. We first propose an algorithm based on the dynamic programming approach for a special case of this problem. Then, we propose another fast heuristic algorithm for the general case. An algorithm based on the branch-and-price approach is finally proposed for larger scale problems and comparison. Experimentations compare solutions found with the heuristic with the corresponding optimal solutions obtained with both Cplex (using the branch-and-bound approach) and the branch-and-price approach. Results demonstrate the good performance level of the heuristic approach for small scale problems, and of the branch-and-price approach for large scale problems.

Keywords : Supply planning, integer programming, dynamic programming, branch-and-price, Cut-to-length timber procurement.

1 Introduction

The forest supply chain is very specific and quite different from the traditional manufacturing supply chains for many reasons ((20), (7), (12), (10), (26)). For instance, forest management

[*]FOR@C Consortium and CENTOR research Center, Pavillon Adrien-Pouliot, Universite Laval, Quebec, Canada;

[†]To whom questions should be addressed.

[‡]Facult de foresterie et de gomatique, Pavillon Abitibi-Price, Universite Laval, Quebec, Canada. Jean-Marc.Frayret@forac.ulaval.ca,Luc.Lebel@sbf.ulaval.ca

and fiber supply planning involves many decision problems ranging from long term strategic forest management (e.g., forest treatment selection, main access road building, forest camp building) to tactical planning (e.g., localization and selection of forest blocks to harvest over a one year horizon, secondary access road building, on site inventory location selection) to short term harvest operations planning (e.g., machines allocation to blocks, bucking pattern rules selection, detailed volume allocation to mills). Furthermore, because it is difficult and costly (whenever possible) to have detailed and accurate information about supply availability (particularly regarding tree diameter, taper function, species, diameter and knot internal location), forest planning is done in a highly stochastic context where information quality is rather poor, especially in natural forest. Robust planning under such uncertainty is thus a challenge that must be addressed by foresters, although it is still mainly a subject for researchers. For instance, (6) address the problem of timber supply under risk of fire. Along the same line, (5) address robust planning taking into account multiple source of uncertainties related mainly to supply availability and machine capacity.

In this paper, we propose to address a supply network planning problem which is to decide what cut-to-length timber should be produced in pre-selected blocks in order to fulfill the short terms needs of many geographically distributed mills. The remainder of this paper is organized in five sections. The next section proposes a literature review of the problems addressed in this paper. Then, Section 3 and 4 introduce respectively the problem and its specific formulation. Section 5 is dedicated to the branch-and-price approach. Empirical evaluation is then presented in section 6. Finally, a conclusion summarizes the proposed approach and presents some research perspectives.

2 Literature review

Due to the divergent nature of the cut-to-length harvesting process (i.e., trees are broken down into various type of logs), Cut-to-length timber procurement planning must address two classic problems. The first problem is a multi-commodity supply planning problem with multiple sources of supply and multiple points of consumption. Then, because one also must decide how trees should be cut into logs of different length, the second problem is a

special case of the classical cutting-stock problem. In this context, not all supply plans are feasible because the heterogeneous nature of the forest and trees obviously constraints the supply volume and the type of logs that can be actually harvested. Consequently, these decisions should be somehow coordinated, if not taken simultaneously. It is typically a problem of integrated process planning and operation scheduling in a context of divergent transformation process.

Multi-commodity supply network planning problems have been known and studied by many authors. Typically, in the multi-commodity distribution problem, a sub-set of plants is identified from a given set in order to satisfy the demand of a list of customers for one or more commodities. This kind of problems is usually characterized by a set of costs, including transportation, production and a fixed cost of choosing a particular plant. Costs may be linear or non-linear. (18) proves that the problem is NP in the case where all plants have limited capacity. For an in-depth review of multi-commodity supply network planning literature, the interested reader is refer to (1).

Although its finality is similar, the problem addressed in this paper is quite different from the classical multi-commodity distribution problem. Indeed, it is different in the sense that there is no fixed cost for choosing any block. Instead, there is a variable production cost that depends upon the number of different product types to produce.

Similarly, cutting-stock problems have been addressed by many authors in many different contexts. For instance, (24) addresses the one-dimensional cutting-stock problem by proposing linear programming models and approaches based on column-generation decomposition. Addressing the same problem, (29) develop an approach for minimizing the number of patterns used for producing the desired products. Also, (2) propose two methods based on dynamic programming to address a large spectrum of industrial cutting-stock problems. For the cutting-stock problem involved in forest operations planning, also referred to as the forest-level bucking optimization problem, the reader is particularly referred to (3) and (19).

In the context of this paper, trees are cross-cut directly in the forest and timber (i.e., logs of various sizes) is supplied to different mills located sometimes far from the forest. Full length stems are thus cross-cut into logs of different sizes according to the mills demand (i.e.,

demand-oriented bucking selection problem). In this process, two costs play an important role: (i) the cost of harvesting (which includes bucking, handling, and sorting); and (ii) the cost of transportation of the timber to the mills. Because the introduction of a new product type (i.e., log of a specific length and specie) to separate at the stump in the final felling reduces harvester productivity by 1 to 4% and forwarder productivity by 3 to 7% ((8), (15)), harvesting cost is a non-linear function of the discrete number of product to be produced in each block. Given that blocks and mills are geographically distributed, there is an opportunity to reduce mills procurement cost by synchronizing bucking and transportation decisions so as to decide at the same time what to produce in each block and what to transport to each mill. In (3) and (19), the objective function of the forest-level bucking optimization problem is to maximize the net profit of using specific bucking patterns on trees of specific stands and classes. In these papers, net profit is calculated as the maximum of the sum of the profit per stem using the bucking patterns that satisfy aggregated demand. In other words, demand is only known per product or market type (e.g., export logs, saw logs, pulp logs) and is not segregated by location. Therefore, because net profit is a function of transportation cost, distance between mills and blocks should affect the selection of bucking patterns. This justifies the need to solve simultaneously the supply network planning problem and the bucking optimization problem.

This type of complex problem can be formulated as a mixed integer programming (MIP) problem because the cost of harvesting is a non-linear function of the discrete number of product types to harvest. The difficulty to solve this type of problem is linked to the combinatorial complexity of the problem. The solution space indeed increases exponentially as the problem size increases. Several papers exist in literature dealing with a variety of MIP problems, which constitute a subclass of combinatorial optimization problems (16). Many problem specific algorithms exist for finding feasible solutions or even optimal solutions. Three basic methods, Branch-and-Bound (21), Cutting plane algorithm, and Dynamic programming are widely used for solving such problems (30), (22), (23). Sometimes these methods are used in conjunction with others. For example, when cutting plane method is used with Branch-and-Bound, the technique is known as Branch-and-cut algorithms (17), or

when column generation (13) is used in conjunction with Branch-and-bound, it is referred to as Branch-and-price (25). The algorithms using branch-and-bound require good starting bound which is usually obtained by LP relaxations. Langrangian relaxation (14) and (11) is one of the popular approaches to obtain such a good bound.

Solving combinatorial optimization problems requires a trade-off between computational time and quality of the solution. In this paper we present a heuristic approach in order to quickly find a good solution, which can be used as an upper bound for the Branch-and-bound algorithm developed to reach the optimal solution.

3 General problem introduction

Because of the inherent complexity of the problem introduced in the previous section (i.e., a combined problem of multi-commodity supply network planning and forest-level bucking optimization), we propose to decompose it into two inter-related problems to be solved iteratively until a solution is reached. The theoretical framework presented in (28) is used here to model the proposed planning process (see figure 1). In brief, Schneeweiss models a complex planning problem by decomposing it into a top (i.e., an aggregated decision problem such as the capacity planning of multiple facilities or a complex production system) and a base level (i.e., a more detailed decision planning such as the operations planning for a short time horizon) problems. Because the base level problem constraints and/or contribute to the objective of the top level problem, the top level must be able to adequately anticipate the influence the base level might have on the top level objective if a particular decision (i.e., top level instruction) is taken. To do so, the top level possesses some form of anticipation function of the base level. (28) identifies four generic types of such functions: perfect; approximate explicit reactive; implicit reactive; and non-reactive.

More specifically, in the context of our problem, the top level represents an extended version of the classic multi-commodity supply network planning problem to take into account the non-linearity of the production cost. Because taking into account at this level all the details about tree diameter distribution and taper function would result in a larger scale problem even more difficult to solve, these parameters are only modelled as aggregated

available fiber volume per specie and block. Furthermore, at this level no restriction regarding the product mix to be harvested is considered. In other words, the anticipation of the selection of particular bucking pattern rules (defined here as a set of bucking patterns specified for each particular class of tree diameter) only considers the unconstrained product mix resulting from the application of these rules. Indeed, many sets of such rules can generate the same product mix, although the resulting resource utilization efficiency may differ from one set of rules to the other. From a modelling perspective, this form of anticipation of the base level is non-reactive as it is not influenced by the instruction of the top level. In (28) terminology, this type of anticipation is called non-reactive because the top level is not explicitly aware of the objective function of the base level. Furthermore, only general features of the base level, such as the aggregated supply availabilities, are considered. Also, it does not guaranty that the instructions of the top level (i.e., a tentative supply planning solutions of the top level) will be feasible (i.e., there is at least one set of bucking pattern rules that satisfies the top level instruction). That is why the base level aims at finding for each block independently the best set of bucking pattern rules, also called the reaction of the base level, that satisfies the production plan (i.e., target volumes of certain products type for each block) of the top level taking into account detailed information about tree diameters distribution and taper function while minimizing resource utilization. The base level consequently consists in many independent bucking decision sub-systems. Due to the non-reactive nature of the anticipation, the instruction of the top level may not be always feasible because of an overestimation of resource availability. This is why, once the base level reaction completely computed for each block, the top level analyze at this stage whether the overall supply plan remains feasible or not. If not, the aggregated availability is adjusted for each block (i.e., increased if availability remains, or decreased in case of local infeasibility) in order to best match the base level reaction. Then, the top level can compute another supply plan that takes into account the new resource availability constraints. This iterative process continues until a feasible solution is found or mills demand must be adjusted.

Within this overall iterative planning process, this paper focuses on the top level normative decision problem. More specifically, the problem we propose to address is a single-period

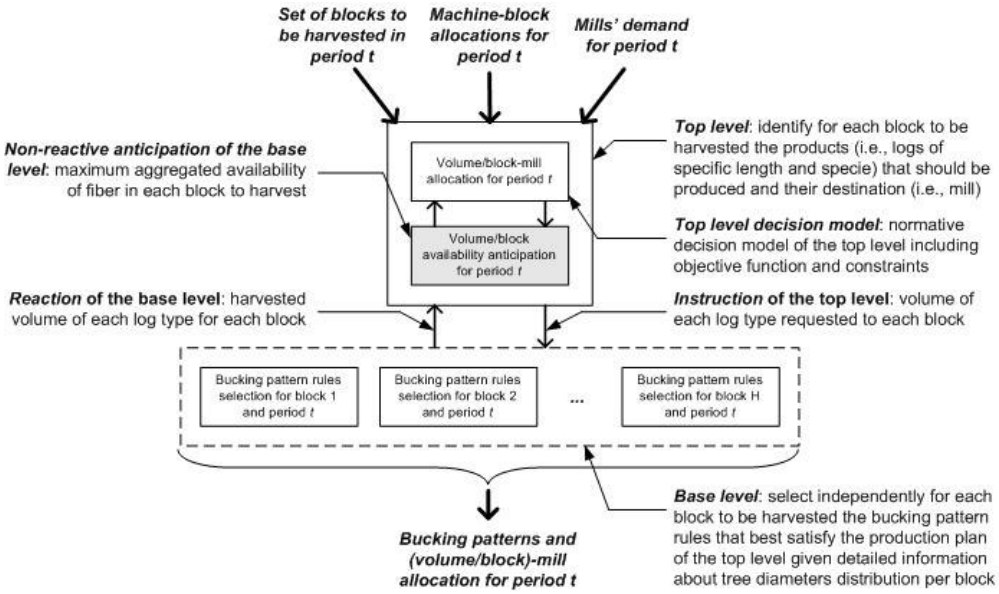

Figure 1: Supply planning

multi-commodity supply planning problem with multiple sources of supply (i.e., blocks to be harvested) and multiple points of consumption (i.e., mills) and with non-linear production cost. In this particular setting, blocks are completely harvested (i.e., no partial harvesting) during the planning period considered. The selection of the blocks to be harvested during this period to satisfy mills demand is part of another decision module that is not described in this paper. The interested reader is referred to (5) for more information.

Concerning demand, we consider that the supply needs of the mills are known and specified in terms of volume (i.e., m^3) of logs of different sizes in length, minimum small end diameter (SED) and species. In order to include the discussed base level anticipation, we assume that all blocks are capable of providing all length of products in limited amounts. In other words, the aggregated available volume for each species in each block is known and adjusted iteratively if necessary. In a context where detailed information about tree diameter distribution is not known in detail, using aggregated information is the easiest way to anticipate what could be harvested from specific blocks. The obvious limit of this approach is that the minimum SED allowed by mills for each product type makes it more or less difficult to realize every supply plan (i.e., find a set of bucking pattern rules for each). For

instance, producing a single product type in a block, while having a rather large allowed minimum SED for that product type (which is typical for long length log type), generally results in an artificially reduced resource availability (because the part of the tree above the minimum SED is not available to produce logs) that is only revealed during the base level analysis. That is why, the introduction of another product type in that block with a smaller minimum allowed SED may increase resource availability (and production cost as well). A trade-off between demand satisfaction and production cost is then necessary. Furthermore, although aggregated information about resource availability is often the only source of information available to foresters, more and more companies show interest in having more detailed information in order to better adjust their harvesting operations to their rapidly changing needs.

4 Specific problem formulation

The objective of the problem we address in this paper is to decide the mix and volume of product that should be harvested in each block so as to satisfy the demand of each mill at the minimum combined cost of harvesting and transportation to the mills. As mentioned earlier, the unit harvesting cost per block increases with the number of product types to be produced in the block. Consequently, we assume in this model that harvesting cost depends upon the number of product types produced in a block, as well as the volume of production for each product type. Next, transportation cost is linear with respect to the volume to be transported and depends upon the distance between blocks and mills. Given these specifications, the MIP formulation of this problem follows.

H	Set of harvesting blocks.
S	Set of mills.
L	Set of log/product types.
K	Set of species.
h, s, l, k	Indexes of blocks (i.e., harvesting units), mills, logs, and tree species respectively.
$x^h_{k,l,s,n}$	Linearized number of logs produced at h, of type l, using tree type k for mill s (decision variable) if n product types are produced in that block.
$a^h_{l,s}$	Unit transportation cost between h and s for log type l ($dollar/m^3$).
V^h_k	Maximum volume of wood type k available at h (m^3).
I^h_n	Binary indicator: takes value 1 if h produces n product-types; otherwise 0.
$J^h_{k,l}$	Binary indicator: takes value 1 if block h produces log type l of species k; otherwise 0.
α_l	volume of a piece of log of type l (m^3/log).
$b_{h,n}$	Unit harvesting cost if n product types are produced at harvesting block h ($dollar/m^3$)
$d_{l,k,s}$	Demand of sawmills for log type l of species k (m^3).

P_1 Min Transportation and production cost:

$$\sum_{n=1}^{N}\sum_{h\in H}\sum_{k\in K}\sum_{l\in L}\sum_{s\in S}(b_{h,n}+a^h_{l,s}).\alpha_l.x^h_{k,l,s,n}$$

Subject to:

$$\sum_{n=1}^{N}\sum_{h\in H}\alpha_l.x^h_{k,l,s,n}=d_{l,k,s}\quad\forall\ \ l\in L \text{ and } s\in S, k\in K. \tag{1}$$

$$\sum_{k\in K}\sum_{l\in L}\sum_{s\in S}x^h_{k,l,s,n}\le Z.I^h_n\ \ \forall\ \ h\in H\ \text{ and } n\in\{1,2,...,N\}. \tag{2}$$

$$x^h_{k,l,s,n}\le Z.J^h_{k,l}\quad\forall\, h\in H,\ k\in K, \tag{3}$$

$$l\in L, s\in S, n\in\{1,2,...,N\}$$

$$J^h_{k,l}=\{0,1\}, \tag{4}$$

$$\sum_{k\in K}\sum_{l\in L} J^h_{k,l} = \sum_{n\in 1,2,...N} n.I^h_n,\ \forall\ h \in H, \tag{5}$$

$$\sum_{n\in 1,2,...N} I^h_n \leq 1\ \ \forall\ h \in H, \tag{6}$$

$$I^h_n = \{0,1\}\quad \forall\ \ h \in H\ \ \text{and}\ \ n \in \{1,2,...,N\}, \tag{7}$$

$$\sum_{l\in L}\sum_{s\in S}\sum_{n\in\{1,2,...,N\}} \alpha_{k,l}.x^h_{k,l,s,n} \leq V^h_k \forall\ \ k \in K,\ h \in H. \tag{8}$$

$$x^h_{k,l,s,n} \geq 0\ \forall\ h \in H,\ k \in K,\ s \in S\ \ \text{and}\ n \in \{1,2,...,N\} \tag{9}$$

Constraint (1) stipulates that total quantity, corresponding to a particular product type, received by any mill from all harvesting blocks must match the corresponding demand of that product type. Constraint (2) guarantees that the decision variables corresponding to $n-$product type takes positive values only if the corresponding indicator (binary variable) I^h_n is positive. Constraint (3) and (4) states that if block h is involved in the harvesting of product of type l and specie k, then the corresponding binary variable $J^h_{k,l}$ should take a positive value. Constraints (5) to (7) are used to select the appropriate cost in the objective function. Finally, constraint (8) corresponds to the resource constraints which indicate that total harvested volume should not go beyond the aggregated resource availability.

Problem P_1 is a large MIP problem. The main challenge to solve it arises from the dependence of one binary variable over others. In other words, the setting of one binary variable to 1 forced all related binary variables to become 0, making the current solution invalid and forcing the need to optimize again. As stated previously, Branch-and-Bound is the only known approach to reach the optimal solution. In our approach we first present an algorithm to find a good and feasible solution that can be used as an upper bound for the branch-and-bound algorithm if the optimal solution is of prime importance. A Branch-and-Price approach is developed to solve the problem to optimality.

5 Properties and Algorithm

In this section we propose to analyze some of the properties of a local optimal solution, and develop algorithms to solve the problem.

5.1 Optimal Algorithm for a special case

Let us consider the case of a single mill and multiple harvesting blocks such that the transportation cost is either negligible compare to harvesting costs or the same from all blocks. Let us also assume that the available volume of fiber in each block is sufficient to meet the mill's total demand. This last hypothesis is realistic in a context of plantation forest where the size of blocks is not necessarily limited. This algorithm would thus be useful to determine the size of the blocks to harvest. In this special case we can solve the problem to optimality using the algorithm described below. In this special case, the cost structure is only influenced by the unit harvesting cost of each block (itself being a function of the number of product type harvested) and not by the total volume harvested in these blocks. Indeed, because of the sufficient resource availability in each block, splitting the mills demand for a particular product type would only results in increasing the unit production cost of at least one block. Furthermore, this unit cost is only specified for a given product type number in terms of volume (m^3) of harvested fiber and not for a particular product mix. Consequently the objective becomes to decide how many product types to produce in each block. This special case of the problem can be solved using an adaptation of an algorithm, based on dynamic programming, presented in (9).

Let us define a function $\phi(h, a)$ where h is the block index and a denotes the number of product type harvested in this block. $\phi(h, a)$ denotes the minimum cost of harvesting a product types from blocks 1 to h. For example $\phi(3, 8)$ denotes the cost of harvesting 8 product types from the first three (indexed 1,2 and 3) blocks. We define $X[h, a]$ a two-dimension array which denotes the number of product types harvested in block h if a total of a product types are harvested from the first h harvesting blocks (i.e., from 1 to h).

The proposed recursive algorithm follows:

Algorithm 1

1. Set $\phi(0, a) = 0$ for $a = 0, 1, 2..., A$.

2. For $h = 1, 2, ..., H$

   (a) For $a = 1, 2, ..., A$

      i. Set $U = Min_{1 \leq x \leq a}\{\phi(h-1, a-x) + b_{h,x}\}$.

      ii. If $\phi(h-1, a) < U$ then set $\phi(h, a) = \phi(h-1, a)$ and $X[h, a] = 0$.
      else

      iii. Set $\phi(h, a) = U$ and $X[h, a] = x$
      In this case $X[h, a]$ contains the x for which U is found

3. Stop.

Explaination

Let us consider we are dealing with the step where h=3. At this step, the values of all $\phi(2, a)$ for $a = 1, 2, ..., A$ have already been calculated. Now at the step (i) of the algorithm, the cost of harvesting $a = 1, 2, ..., A$ products from the first three blocks is computed. Assume $a = 5$, we want to know how much it costs to produce five product types using the first three harvesting blocks (since h=3). The available options are to produce either one product type in the third block and four product types in the first two blocks, or two product types in the third block and three product types in the first two, and so on. Since the harvesting cost of all combinations of producing 1,2,...,5 product types using the first two blocks is already calculated, the cost of producing five product types using the first three blocks is easy to compute. At this step, it is possible that harvesting all five product types from the first two blocks is cheaper than sharing the product types among three blocks. If it is the case, at step (ii), $X[3, 5]$ is set to 0. In other words, it means that the third block produces nothing if the objective was to harvest five product types using the first three blocks. At step (iii), the cost is set in $X[h, a]$ if the third block produces at least one product type.

Computation of each $\phi(p, a)$ requires $A + 1$ steps. So the total steps involve in the algorithm are $\sum_{k=0}^{A} \sum_{h=1}^{H} (A + 1)$ or $A.H.(A + 1)$ to find the optimal value of the special case problem.

The above algorithm identifies how many product types should be harvested from each block. The next step is thus to find which specific product types and what volume should be harvested. This step is rather straight forward. To do so, we first sort the blocks selected for harvesting in increasing order of harvesting cost (note that the above algorithm will tell how many product types to be harvested from each block and therefore the harvesting cost for the block). Similarly, we next sort the product types in decreasing order of demanded volume. Finally, we select the first block on the list and allocate as many number of product types from the product type list as decided for that block. Then, we take the second block and do the same until no blocks are left in the block list.

5.2 General case

Here, at this point we drop the index n (correspond to cost type) from $x^h_{k,l,s,n}$. Now the following holds for the general case.

Proposition 1

There exists an optimal solution in which, for any mill s and for a given product type l of a specific specie k

$$x^h_{k,l,s} \in \{0\} \cup \{d_{l,k,s}\} \text{ for } h \in E(s) \tag{10}$$

where $E(s) \subset \{H\}$

or

$$x^h_{k,l,s} \notin \{0\} \cup \{d_{l,k,s}\} \text{ for } h \in \{1.2, ..., H\} \backslash E(s) \tag{11}$$

But in this case $V^h_k - \sum_{s \in S} \sum_{l \in L} \alpha_{k,l}.x^h_{k,l,s} < \alpha_{k,l}.x^h_{k,l,s}$ for $h \in H \backslash E(s)$ except may be for at most one h.

Proof

Let $X = \{x^h_{k,l,s}\}$, $h \in H$, $s \in S$ be a feasible solution to the problem. Assume that the supply corresponding to block $h_1, h_2 \in H$, respectively $x^{h_1}_{k,l,s}$ and $x^{h_2}_{k,l,s}$, do not verify the conditions of proposition 2 for sawmill s. In other words:

$$x^{h_1}_{k,l,s} \notin \{0\} \cup \{d_{l,k,s}\} \text{ and } V^{h_1}_k - \sum_{s\in S}\sum_{l\in L} \alpha_{k,l}.x^{h_1}_{k,l,s} \geq \alpha_{k,l}$$

and

$$x^{h_2}_{k,l,s} \notin \{0\} \cup \{d_{l,k,s}\} \text{ and } V^{h_2}_k - \sum_{s\in S}\sum_{l\in L} \alpha_{k,l}.x^{h_2}_{k,l,s} \geq \alpha_{k,l}$$

Assume that the cost per m^3 of harvesting and transporting from h_1 is higher than the cost from h_2, we have:

$$(b_{h_1,n} + a^{h_1}_{l,s}) \geq (b_{h_2,n} + a^{h_2}_{l,s}) \tag{12}$$

Now choose $\delta = Min(V^{h_2}_k - \sum_{s\in S}\sum_{l\in L} \alpha_{k,l}.x^{h_2}_{k,l,s}, x^{h_1}_{k,l,s})$ and set $y^{h_2}_{k,l,s} = x^{h_2}_{k,l,s} + \delta$, $y^{h_1}_{k,l,s} = x^{h_1}_{k,l,s} - \delta$. Now define the new solution set $Y = \{y^h_{k,l,s}\}$, by replacing in X, $x^{h_1}_{k,l,s}$ by $y^{h_1}_{k,l,s}$ and $x^{h_2}_{k,l,s}$ by $y^{h_2}_{k,l,s}$. The new solution obtained (i) remains feasible because the harvested volume is only switched from one block to the other with respect to availability constraints; (ii) the harvesting cost function corresponding to block h_2 remains the same as before since the above mechanism do not add any new product type in forest h_2 (i.e., only a volume of a product type and specie already harvested is added); (iii) the number of product types corresponding to block h_1 may reduces (as well as the harvesting cost corresponding to the rest of the supply harvested in k because of the non-linear increase in productivity discussed earlier) by one unit if $\delta = x^{h_1}_{k,l,s}$.

From the inequality (12) and the fact that costs are positive $(b_{h_1,n} + a^{h_1}_{l,s}).x^{h_1}_{k,l,s} + (b_{h_2,n} + a^{h_2}_{l,s}).x^{h_2}_{k,l,s} \geq (b_{h_1,n} + a^{h_1}_{l,s}).y^{h_1}_{k,l,s} + (b_{h_2,n} + a^{h_2}_{l,s}).y^{h_2}_{k,l,s}$. The above inequality shows that any solution not verifying the conditions of proposition 1 cannot be the optimal solution because there is potentially at least one better solution. This completes the proof. ■

Now, if two or more sawmills (for a particular product) are supplied from two or more blocks, then the satisfaction of proposition 1 does not guaranty that the solution is optimal, especially if the condition of the proposition 2 is satisfied.

Proposition 2 Assume that the conditions of proposition 1 apply, then if $x^p_{k,l,q} \notin \{0, d_{l,k,b}\}$ where $p \in \{h_1, h_2\} \subseteq H$ and $q \in \{s_1, s_2\} \subseteq S$ then the solution is not opti-

mal.

Proof

Let X be a set of feasible solution which do not satisfy the conditions of proposition 2. For any pair of h_1, h_2 and s_1, s_2, either of the following must hold.

$$(b_{h_1,n_{h_1}} + a^{h_1}_{l,s_1}) + (b_{h_2,n_{h_2}} + a^{h_2}_{l,s_2}) < (b_{h_1,n_{h_1}} + a^{h_1}_{l,s_2}) + (b_{h_2,n_{h_2}} + a^{h_2}_{l,s_1}), \tag{13}$$

or,

$$(b_{h_1,n_{h_1}} + a^{h_1}_{l,s_1}) + (b_{h_2,n_{h_2}} + a^{h_2}_{l,s_2}) \geq (b_{h_1,n_{h_1}} + a^{h_1}_{l,s_2}) + (b_{h_2,n_{h_2}} + a^{h_2}_{l,s_1}) \tag{14}$$

Now, define $\delta_1 = Min(x^{h_1}_{k,l,s_2}, x^{h_2}_{k,l,s_1})$ and $\delta_2 = Min(x^{h_1}_{k,l,s_1}, x^{h_2}_{k,l,s_2})$. If (13) holds (the opposite is true if (14) holds) then define a new solution set Y same as set X and set the following variables as follows: $y^{h_1}_{l,k,s_1} = x^{h_1}_{l,k,s_1} + \delta_1$, $y^{h_2}_{l,k,s_2} = x^{h_2}_{l,k,s_2} + \delta_1$, $y^{h_1}_{l,k,s_2} = x^{h_1}_{l,k,s_2} - \delta_1$ and $y^{h_2}_{l,k,s_1} = x^{h_2}_{l,k,s_1} - \delta_1$.

The new solution Y remains feasible because the harvested volume is only switched from one block to the other with respect to availability constraints. Also, the above procedure does not add any new product type to any block and may even reduce the number of product types from a block. In the resulting solution, the harvesting cost function associated with the blocks either reduces or remains the same. Even if the harvesting cost remains the same for all blocks. the total cost corresponding to the new solution reduces according to relation (13). ■

5.3 Scenario improvement heuristic

The propositions 1 and 2 provide necessary conditions for a solution to be optimal though all these properties are the properties of a locally optimal solution. In this section we develop a heuristic approach which always satisfies the above conditions and gradually improve the solution. The idea is to start with a given scenario and improve it in order to reduce its overall cost. In this algorithm we start with a feasible solution. We set the harvesting and transportation costs of a dummy sawmill to zero in order to absorb all resources from all blocks to maintain solution feasibility. Then, the remaining steps of the algorithm are as follows:

1. Introduce the harvesting cost of each block associated with the current solution.

2. Compute the reduced cost corresponding to all non-basic variables for the current solution. Select the M variables with highest reduce cost.
   For a given set of costs, the problem is a linear programming problem from which it is easy to calculate dual cost vector and reduced costs

3. For each M decision variable, using proposition 1 and 2 we check if the cost can improve by tentatively introducing it into the basis. Then we calculate the real cost (based on the real allocation) for each M cases.

4. We select the variable which improves the cost the most and introduce it in the basis. If this solution is better than the best solution we obtained so far, we replaced the best solution by the current solution. If the cost does not improve for all the selected M non-basic variables, then we stop, otherwise go to step 1.

In the algorithm we must check M non-variables because the cost does not remain the same when we change the number of product type to be harvested (i.e., as the allocation changes, the harvesting cost may also change). If the cost were constant, then selecting a variable with positive reduced cost would suffice to improve the solution (simplex method). We choose M variables to investigate based on the experience. For example, for the 80 binary variable problems we use M=50 and for 400 binary variable problems we used 500. Furthermore, we may use transportation simplex method by introducing a dummy sawmill for an ease in calculation of the dual cost and reduced cost.

The above approach is good to get a good starting upper bound for the problem. Since, the algorithm do not involve binary variables, the total number of variables required by the algorithm are $H * (S * K * P + K)$ whereas the number of ordinary variables required in the BIP formulation is $K * P * (H * S * K * P)$ and the number of binary variables are $H * (2 * P * K)$. Second advantage of the algorithm is that the solution is always feasible whereas the solution obtained by solving the relaxed problem of P_1, to get an initial lower bound, may not be feasible.

For real instances of the problem, the algorithm performed very well because the harvesting and transportation costs generally differ significantly from one forest to other. If these costs are similar to each other, then the algorithm performance tends to deteriorate. This motivated us to develop an algorithm which gives a better result. In the next section we propose branch-and-price approach for the exact solution. We will use the above heuristic to compute the initial starting solution and the upper bound.

6 Branch-and-price approach

In this section we first propose a different formulation of the above problem and then discuss the solution approach. We also use additional notations to simplify the formulation. Let r denotes the total number of product-types demanded by the all sawmills i.e. $r = |\{\ 1 \mid d_{l,k,s} \geq 0, l\ \in\ L,\ k\ \in\ K,\ s\ \in\ S\}|$ where $|.|$ represent the cardinality of a set. For simplicity we define the demand vector $[B]_{r\ \times\ 1}$ whose elements $b_{k,l,s}$, sequentially represent the demand of product type (k, l) from sawmill s. In other words each element of B represent the demand of a particular item of particular sawmill. For example if we have 2 log lengths, 2 species and 3 sawmill then the first element of B will be $d_{1,1,1}$, 2nd will be $d_{1,2,1}$, third will be $d_{2,1,1}$, 5th will be $d_{1,1,2}$ and so on. We similarly define a column vector $[X^h]_{r\ \times\ 1}$, which represents the supply to sawmill s from block h. Each element in X represent the supply of a particular item to the corresponding sawmill. For example, the 5th element of $[X^h]$ will represent supply of wood type $(1, 1)$ to sawmill 2 by block h. Several columns (i.e., X^h) may be possible for a given block h. We denote by G^h the set of all feasible columns corresponding to block h. In order to identify each column of G^h, we introduce the index i (i.e., X_i^h where $i \in G^h$). We denote the elements of vector $[X_i^h]$ by $[x_{h,i,j}]$ where $j \in \{1, 2, ..., r\}$. Let us define a binary variables $z_i^h,\ i \in G^h$, corresponding to each column i associated with block h. z_i^h takes a positive value ($z_i^h = 1$) if the column, i, is selected or zero otherwise. The cost associated with column i of block h is denoted by c_i^h and is based on the supply represented by the column. Since our objective is to select at most one column from each block, we introduce a convexity constraint for each block. Let Y_i^h is $|H|$-elements column vector of 0s except the h^{th} element which is 1. The $j - th$ element of Y_i^h is denoted by $y_{h,i,j}$. Now the i^{th} column

corresponding to block h is composed of X and Y and can be denoted by $[X_i^h\ Y_i^h]$. Using the above notations we can formulate the problem as follows:

$$P_2 \qquad Min \ \sum_{h\in H}\sum_{i\in G^h} c_i^h.z_i^h$$

Subject to

$$\sum_{h\in H}\sum_{i\in G^h} x_{h,i,j}.z_i^h = b_j \ j \in \{1,2,...,r\}, \tag{15}$$

$$\sum_{i\in G^h} y_{h,i,j}.z_i^h = 1 \ j \in \{1,2,...,H\}, h \in H, \tag{16}$$

$$z_i^h \ \in \ \{0,1\} \tag{17}$$

The above formulation guaranties the optimal solution only if set G^h contains all the feasible columns corresponding to block h. Generating all the possible columns beforehand for each block may not be feasible. Therefore, only a limited version of the problem containing few columns is solved, while new profitable columns are iteratively introduced until the optimal solution is found. Since problem P_2 is a BIP (binary integer programming) model, to solve it we first optimize the relaxed version (relaxing constrains (16)) of the problem and then we use the branch-and-bound strategy to search for the integer solution. We denote by $P_2(RM)$ the restricted and continuous version of the problem P_2.

Let us assume the u_η, $\eta = 1,2,...,r+|H|$ denotes the dual variables associated with the current optimal solution of $P_2(RM)$. In order to generate a new column, say t, we must solve the following pricing problem:

$$Pr(h) \quad \text{Max} \quad \sum_{\eta=1}^{r} u_\eta.x_{h,t,\eta} + u_{r+h}.y_{h,t,h} - c^h$$

ST

$$y_{h,t,h} = 1 \tag{18}$$

$$\sum_{s\in S} \alpha_{k,l}.x_{h,t,f(l,k,s)} \leq V_k^h \ \forall \ k \ \in \ K, l \ \in \ L \tag{19}$$

where $f(l,k,s)$ generate the index according to the following formula.
$f(l,k,s) = (s-1).L.K + (k-1).L + l.$

The cost c^h depends upon the number of products harvested in h (see appendix for the constraints and objective function terms related to c^h). At each iteration we solve $Pr(h)$ for each block and we select the column corresponding to the highest positive reduced cost. We introduce the column in appropriate columns set i.e. G^h and solve again the $P_2(RM)$. We terminate the column generation algorithm if there is no column with positive reduced cost. The optimal solution of $P_2(RM)$ may not contains all z_i^h as integer and thus we use Branch-and-bound approach at this stage to search an integer solution. Since z_i^h represent a column we select the column for the left branch ($z_i^h = 1$) and remove the same column from the right branch ($z_i^h = 0$). In our approach we use depth first search strategy from left to right. For branching we select the column which has the highest number of positive elements i.e. $x_{h,i,j}, h \in H,\ i \in G^h, j \in \{1,2,...,r\}$. Mathematically, we identify a column i_1 of block h_1 such that:

$$|\{1 \mid x_{h_1,i_1,j} > 0, j \in \{1,2,...,r\}\}| \geq |\{1 \mid x_{h_2,i_2,j} > 0, j \in \{1,2,...,r\}\}|$$

where $h_1, h_2 \in H$, $i_1 \in G^{h_1}$ and $i_2 \in G^{h_2}$. The column i_1, h_1 is selected for the left branch and blocked for the right branch. It is evident at this stage that the computation effort to find a new column at each level, in the branch-and-bound tree, increases for the right branch and decreases for the left branch since in the left branch one block is always blocked and thus reduces the number of sub problem whereas in the right branch number of sub problem remains the same but with extra constraints.

For the detail overview of branch-and-price approach, branching strategy and useful results reader can refer (27) and (4).

The efficiency of column generation approach depends upon how effectively we solve the sub problem. To solve $Pr(h)$ using integer programming is very costly. Since the sub-problem is dealing with only one block we can develop a fast algorithm to find a near optimal solution. Let $G(k,l)$ represent a set of indexes of all x which correspond to product type

(k,l). Mathematically: $G(k,l) = \{f(k,l,s) \mid s = 1,2,...,S\}$. $C(G(k,l))$ gives the portion of cost associated with decision variables in $G(k,l)$ i.e. for a given h and χ (number of product-type)

$$C(G(k,l)) = \sum_{j \in G(k,l)} (\beta j - b_{h,\chi}).x_j.$$

Now the algorithm is as follows:

Let $X^* = x_1^*, x_2^*, ..., x_r^*$ is the set of decision variables to store the optimal solution. Define the sets $G(k,l)$ for all $k \in K$ and $l \in L$.

Algorithm2

1. Set $\chi = 0$, and set $BestCost = 0.0$
2. Set the $\chi = \chi + 1$ and solve the problem $P3$. Let $X^1 = x_1^1, x_2^1, ..., x_r^1$ be the solution.
3. Define $G(k,l)$ for the solution and set counter=0;
4. For k=1 to K, and For $l = 1$ to L
   (a) $Counter = Counter + 1$.
   (b) Compute $cost[Counter] = C(G(k,l))$. *First we solve the problem $P3(\chi)$ (see appendix) using liner programming solver and then use the function C(G(k,l)).*
5. Sort the array $cost[]$ in decreasing order and set $sum = \sum_{i=1}^{\chi} cost[i]$. *We keep in memory the indexes (k,l) corresponding to the first $1, ..., \chi$ best cost after sorting. We represent it by $(k^1, l^1), (k^2, l^2), ..., (k^\chi, l^\chi)$*
6. If sum < BestCost, stop.
   else
7. Set BestCost=sum.
8. For $i = 1, ..., \chi$
   (a) Set $x_j^* = x_j^1$ if $j \in G(k^i, l^i)$. Else
   (b) Set $x_j^* = 0$.
9. Go to 2

6.1 Numerical Example

In this section we present experimentation results for both the heuristic algorithm and the branch-and-price algorithm and compare them with the optimal result obtained with MIP solver. All the different problem instances were generated randomly using the following data. In all examples, harvesting cost is randomly generated between 2 to 5 such that it should increase as product-type increases. Similarly demand for each product type varies between 30 and 100, and transportation cost varies between 1.0 and 5.0. In all cases, heuristic, we have fixed M equal to 250. The algorithms are coded in C++ and implemented on P4 700Mhz computer with 512 MB RAM.

Table 1 presents the computational performance of algorithm 2 (to solve problem $P_r(h)$). In all examples we generated demand between 0 and 500, production cost for n product-type is the sum of random number between 1 and 4 and the production cost of $n-1$ product-types. u variables lies between 0-70, transportation cost is generated according to the following formula: *unit product volume*0.1+0.01*index of mill*. Ten instances of each problem size is solved. The mean percentage error and relative time gain *([Time taken by MIP-Time taken by heuristic]/[Time taken by MIP])*. Finally table 2 present the computational performance of the branch-and-price approach over the MIP formulation. We solve five instances of three different size problems. In all the problems we consider two blocks and two sawmills. Other parameters are generated randomly as for the previous examples. The algorithm is developed using C++ and use concert technology provided by ILOG-CPLEX.

For small problem instances, less than 5 products and up to 4 blocks, both MIP and heuristic performance is similar. But as the problem size grows MIP consumes more time and memory resources. Table 2 present the performance of all approaches (i.e. MIP, Heuristic, and B&P). The first data is correspond to the solution obtained by the algorithm and the data in bracket represent the time (in seconds) taken by the algorithm. Problems 16 to 20, we stop the MIP solver after 1000 seconds as by this time tree size grows to 1 GB. Consequently, in brackets we report the optimality gap reported by CPLEX at the end of 1000 seconds. We also terminate the branch-and-price approach when the optimality gap is less than or equal to 2.5%. From the result we can deduce that the lower bound provided

Table 1: Computational performance (Algorithm 2)

Number of mills	Number of product-types	Mean percentage error	standard deviation of mean percentage error	Average time saving in using algorithm 2 over MIP
5	15	2.15736	1.99564	0.682582
5	20	2.09111	2.28346	0.841711
5	25	3.01229	4.25334	0.855807
5	30	1.98382	2.67125	0.987875
10	15	3.0955	4.06499	0.868333
10	20	0.513287	0.813698	0.9353
10	25	2.34493	4.26834	0.950185
10	30	0.547781	0.747188	0.997277
15	15	2.32249	3.09511	0.923672
15	20	0.871089	1.26744	0.957821
15	25	2.19865	4.29367	0.973984
15	30	0.607191	1.0062	0.998478
20	15	1.51891	1.99983	0.952894
20	20	1.59909	1.66634	0.978066
20	25	1.46996	1.68545	0.984762

by the branch-and-price approach is tighter than the cplex.

7 Conclusion

This work presents a problem of multi-commodity supply planning in a cut-to-length timber procurement context. A MIP formulation is presented for the problem and solution approaches are developed. It is well known that large size mixed integer programs are difficult to solve optimally in reasonable time. We thus present two approaches to tackle the problem. The first approach is a heuristic approach which starts with a feasible solution and improves the solution up to a certain limit. The approach is fast and provides a good solution quickly. The obtained solution can then further be used as a good starting bound for the branch-and-bound algorithm if the optimal solution is of prime importance. In the second approach we present a branch-and-price technique to solve the problem to optimality. Since the problem is NP-difficult as the size of the problem increases, the complexity of the

Table 2: Computational performance

	Problem type (block, mill, product-type)	MIP	Heuristic	Branch and price
1.	2,2,5	108.53 (0.16)	108.53 (0.078)	108.53 (2.54)
2.	2,2,5	98 (0.14)	98 (0.079)	98 (1.5)
3.	2,2,5	106 (0.06)	106 (0.015)	106 (1.42)
4.	2,2,5	105.3 (0.08)	105.3 (0.016)	105.3 (1.687)
5.	2,2,5	82.7 (0.01)	82.7 (0.031)	82.7 (0.92)
6.	4,2,5	87.502 (0.19)	87.5 (0.062)	87.5 (3.359)
7.	4,2,5	111.2 (0.14)	111.2 (0.265)	111.2(2.703)
8.	4,2,5	99.8 (0.2)	101.2 (0.062)	99.8 (2.96)
9.	4,2,5	92.38 (1.05)	94.98 (0.218)	92.38 (4.062)
10.	4,2,5	106.652 (0.19)	111.8(0.093)	106.6 (3.4)
11.	4,5,5	259.41 (0.74)	259.41 (0.468)	259.41 (27.4)
12.	4,5,5	214.53 (3.77)	214.53 (0.5)	214.53 (22.92)
13.	4,5,5	290.79 (14.95)	294.06 (0.281)	290.79 (24.843)
14.	4,5,5	277.03 (10.67)	277.03 (0.203)	277.03 (20.156)
15.	4,5,5	287.94(1.58)	293.386 (0.234)	287.94 (23.484)
16.	4,5,10	550.57 [10.05%]	562.51 (0.718)	552.6(364.06)
17.	4,5,10	596.47 [8.2%]	634.97 (0.796)	596.43 (385.89)
18.	4,5,10	597.38 [7.57%]	644.27 (0.75)	595.7 (466.376)
19.	4,5,10	621.51 [6.89%]	631.13 (0.562)	624.508 (444.64)
20.	4,5,10	658.06[9.19%]	686.456 (0.515)	658.06 (412.127)

sub-problems also increase and hence cannot be utilize for a very big problem. But for a real world problem which do not involve more than 10-15 supply points and 10-15 receiving points both approaches works very well. Future research is still under progress to tackle the sub problems in a more efficient way.

8 Acknowledgement

This research project has been carried out with the support of Jean Favreau and Joseph Nader from the Forest Engineering Research Institute of Canada (FERIC) who have contributed to this research by their insightful comments and idea.

Appendix

$$Min \;\; c^h = \sum_{n=1}^{N} \sum_{k \in K} \sum_{l \in L} \sum_{s \in S} (b_{h,n} + a^h_{l,s}).x_{f(l,k,s),n}$$

S.T.

$$x_{f(l,k,s),n} \leq Z.J_{k,l} \forall \; k \; \in \; K; l \in L; s \in S, \tag{20}$$

$$\sum_{l \in L} \sum_{k \in K} \sum_{s \in S} x_{f(l,k,s),n} \leq Z.I_n \; \forall \; n \in \; 1, 2, ..., N, \tag{21}$$

$$\sum_{l \in L} \sum_{k \in K} \sum_{s \in S} x_{f(l,k,s),n} \leq Z.I_n \; \forall \; n \in \; 1, 2, ..., N, \tag{22}$$

$$\sum_{l \in L} \sum_{s \in S} \alpha_{k,l}.x_{f(l,k,s),n} \leq V_k, for \; k \in K, \tag{23}$$

$$\sum_{i=1}^{N} i.I_i = \sum_{k \in K} \sum_{l \in L} J_{k,l}, \tag{24}$$

$$\sum_{i=1}^{N} I_i <= 1, \tag{25}$$

$$I_n \in \{0, 1\}, n \; \in \; \{1, 2, ..., N\}, \tag{26}$$

$$J_{l,k} \in \{0, 1\}, \forall \; k \; \in \; K; l \in L. \tag{27}$$

Combining the c^h function in problem $Pr(h)$ we find that the cost associated with the variable $y_{h,t,h}$ is fixed (see the constraint which set the value 1) and the objective is to maximize the rest of the objective function. The objective function consists of decision variables for each scenario (scenario corresponding to every product-type cost) and the dual variables. Combining the dual variables with the scenario cost (transportation cost and the harvesting cost), we have $(d_{h,i} - a^h_{l,s} - b_{h,n})$. We use the same expression in the MIP formulation of pricing algorithm.

Let us define $\beta_{k,l,s} = d_{h,f(k,l,s)} - a^h_{l,s}$ now the problem $P3(\chi)$ is defined as follows:

$$P3(\chi) \qquad Max \sum_{k\in K}\sum_{l\in L}\sum_{s\in S}(\beta_{k,l,s} - b_{h,\chi}).x_{f(k,l,s)}$$

S.T.

$$x_{f(k,l,s)} \leq V_k^h \ \forall \ k \ \in \ K, l \ \in \ L \tag{28}$$

References

[1] C.H. Aikens. "Facility location models for distribution planning". *European Journal of Operational Research*, 22:263–279, 1985.

[2] Julien Antonio, Fabrice Chauvet, Chengbin Chu, and Jean-Marie Proth. The cutting stock problem with mixed objectives: Two heuristics based on dynamic programming. *European Journal of Operational Research*, 114:395–402, 1999.

[3] Edurado Julio Arce, Celso Carnieri, Roberto Carlos Sanquetta, and Afonso Figueiredo Filho. A forest-level bucking optimization system that considers customer's demand and transportation costs. *Forest Science*, 48, 2002.

[4] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46, 1998.

[5] D. Beaudoin, L. Lebel, and J.-M. Frayret. Tactical supply chain planning and robustness analysis in the forest products industry. *Canadian Journal of Forest Research*.

[6] D. Boychuk and D.L. Martell. A multistage stochastic programming model for sustainable forest-level timber supply under risk of fire. *Forest Science*, 42, 1996.

[7] David Bredström, Jan T. Lundgren, Mikael Rönnqvist, Dick Carlsson, and Andrew Mason. "Supply chain optimization in the pulp mill industry ip models, column generation and novel constraint branches". *European Journal of Operational Research*, 156:2–22, 2004.

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

[9] S.S. Chauhan, Anton V. Eremeev, Alexander A. Kolokolov, and Vladimir V. Servakh. On solving concave cost supply management problem with single manufacturing unit. *Proceedings of SCM Conference, Poland*, 2002.

[10] R. Epstein, R. Morales, J. Seron, and A. Weintraub. Use of or systems in the chilean forest industries. *Interfaces*, 29, 1999.

[11] Fisher.M. An application orientated guide to langrangian relaxation. *Interfaces 15*, 2:10–21, 1985.

[12] J.-M. Frayret, K. Boston, S. DAmours, and L. Lebel. The e-nabled supply chain : opportunities and challenges for the forest products industry. *Journal of Forest Products Business Research*.

[13] Dantzig G.B. and P. Wolfe. Decomposition principles for linear programming. *Operations research*, 8:101–111, 1960.

[14] A. Geoffrion. Langrangian relaxation for integer programming. *Mathematical programming study2: Approaches to Integer programming Balinski New York: North-Holland*, pages 82–114, 1974.

[15] J.-F. Gingras and J. Favreau. Incidence du triage sur la productivit des systmes par bois trononns. *Avantage*, 3, 2002.

[16] Erwin Hans. *Resource loading by Branch-and-Price techniques*. PhD thesis, Beta research school for operations management and logistics, 2001.

[17] Hoffman K. and M. Padberg. LP-based combinatorial problem solving. *Annals of Operations Research*, 4:145–194, 1985.

[18] J. Krarup and P.M. Pruzan. "The simple plant location problem: Survey and synthesis". *European Journal of Operational Research*, 12:3681, 1983.

[19] A. Laroze. A linear programming, tabu search method for solving forest-level bucking optimization problems. *Forest Science*, 45, 1999.

[20] David L. Martell, Eldon A. Gunn, and Andres Weintraub. Forest management challenges for operational researchers. *European Journal of Operational Research*, 104:1–17, 1998.

[21] L.G. Mitten. Branch-and-bound methods: General formulation and properties. *Operations Research*, 5:268–279, 1970.

[22] K.G. Murty. *Linear and combinatorial programming*. John Wiley & Sons.

[23] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience publication, 1988.

[24] Holthaus Oliver. Decomposition approaches for solving the integer one-dimensional cutting stock problem with different type of standard lengths. *European Journal of Operational Research*, 141:295–312, 2002.

[25] Vance P.H., C. Barnhard, E.L.Johnson, and G.L.Nemhauser. Solving binary cutting stock problems by column generation and branch-and-bound. *Computational optimization and applications*, 3:111–130, 1994.

[26] M. Rönnqvist. Optimization in forestry. *Mathematical Programming, Series B*, 97, 2003.

[27] M.W.P. Savelsbergh. A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, 45:831–841, 1997.

[28] C. Schneeweiss. Distributed decision making, 2sd. edition. *Springer-Verlag*, 20032. New York.

[29] Shunji Umetani, Mutsunori Yagiura, and Toshihide Ibaraki. One-dimensional cutting stock problem to minimize the number of different patterns. *European Journal of Operational Research*, 146:388–402, 2003.

[30] Wayne L. Winston. *Operations research application and algorithms*. Duxbury press, 1993.