Testing the Use of Lazy Constraints in Solving Area-Based Adjacency Formulations of Harvest Scheduling Models

Abstract: Spatially-explicit harvest scheduling models to enforce maximum harvest opening size restrictions often lead to combinatorial problems that are hard to solve. This paper shows that the inequalities required by one of the three existing formulations, the Path Model are typically *lazy*. In other words, these constraints are rarely binding during optimization, especially if the maximum opening size is large relative to the average management unit size. By solving 60 hypothetical and eight real forest problems with varying maximum clear-cut sizes and to varying target optimality gaps, we confirm that applying the Path constraints only when they are violated during optimization leads to shorter solution times. While the lazy Path constraints performed better than the other formulation/solution approaches, the relative superiority of the method was more obvious at larger optimality gaps. Nearly 95% of the problem instances solved fastest with the "lazy" method at a target gap of 1%, and almost 92% solved fastest at 0.05%. At 0.01%, the Lazy Path approach was still superior in the majority of cases, but the percentage was much lower: 57%. This is a significant improvement compared to the 14, 10 and 19% shares of the other approaches. If only the real instances are considered, the Lazy Path approach performed best in 68% of the instances with 1% and 0.01% optimality gaps and in 61% of the instances with 0.05% gap. A closer analysis of the results suggests that the relative superiority of the approach increases with problem size and maximum clear-cut size.

Keywords: spatial forest planning, integer programming

Introduction

Spatially-explicit harvest scheduling models optimize the spatial and temporal layout of forest management actions in order to best meet management objectives such as profit maximization, even flow of products, and wildlife habitat preservation while satisfying a variety of constraints, including maximum harvest opening size restrictions. These models assign various silvicultural prescriptions, such as clearcuts, thinning or shelterwood treatments, to forest management units within a predetermined land-base. In addition, spatially-explicit decisions may also be modeled. These decisions, such as whether to treat a harvest unit or to build a road link in a given planning period, are typically represented with binary variables that can take only the values of 0 or 1. A variety of other restrictions, some spatially-explicit and some not, are also typically included such as timber-flow smoothing constraints (e.g., Thompson et al. 1994), minimum average ending age or inventory constraints (e.g., McDill and Braze 2000), and maximum harvest opening size restrictions (e.g., Meneghin et al. 1988).

The need for spatial specificity in these models, and the use of discrete optimization, has emerged primarily as a result of adjacency restrictions. Adjacency, or "green-up," constraints limit the maximum size of contiguous harvest openings. These restrictions, which are often required by law or policy in North America (e.g., Barrett et al. 1998, American Forest & Paper Association 2000, Boston and Bettinger 2002), have been promoted as a tool to mitigate the negative impacts of harvesting forested ecosystems (e.g., Thompson et al. 1973, Jones et al. 1991, Murray and Church. 1996a, 1996b, Snyder and ReVelle 1996a, 1996b, 1997a, 1997b, Carter et al. 1997, Murray 1999). Although maximum harvest opening size constraints do indeed disperse harvesting activities across the landscape, and thus reduce the concentration of this type of human disturbance, they have also been shown to fragment and disperse mature forest habitats (Harris 1984, Franklin and Forman 1987, Barrett et al. 1998, Borges and Hoganson 2000). To mitigate these negative consequences of these restrictions, Rebain and McDill (2003a, 2003b) proposed a 0-1 programming formulation that allows the forest planner to promote or to require the preservation, maintenance or creation of a certain amount of mature forest habitat in large patches over time in models with maximum harvest opening size constraints. A drawback of combining both harvest opening size and mature patch habitat constraints is that the resulting models are large, complex, and hard to solve. Considerable effort has been made to improve our ability to obtain high-quality solutions for these models within reasonable time frames such as a few hours. This study focuses on improving the performance of models with harvest opening size constraints. We show that the so-called Path constraints (McDill et al. 2002), which are required by one of the existing models to ensure maximum harvest opening size restrictions, are rarely active (binding) during optimization, especially if the size limit on harvest openings is large. Furthermore, since these constraint sets tend to be large, we hypothesize that putting these inequalities in *lazy constraint pools*, i.e., using them only when they are violated by a solution during optimization, can lead to dramatic improvements in solution times.

The rest of the Introduction discusses the existing exact optimization models for maximum harvest opening size restrictions and further explains our hypothesis about the "lazy" nature of Path constraints. In particular, the potential significance of this property with respect to the computational performance of harvest scheduling models is discussed. The empirical study described in this paper compares the solution times that can be achieved by the existing models with those of the Lazy Path approach using 60 hypothetical and eight real test problem instances, and different maximum harvest opening size levels.

The simplest type of maximum harvest opening size constraints prevent adjacent management units from being harvested within the same time period (McDill and Braze 2000). This case, referred to as the Unit Restriction Model (URM, Murray 1999), assumes that the combined area of any two units in the forest would exceed this maximum area. The Area Restriction Model (ARM, Murray 1999) is more general, allowing groups of contiguous management units to be harvested concurrently as long as their combined area is less than the maximum opening size. Depending on the average area of management units, the maximum harvest opening size, and the age-class distribution of the forest, the ARM formulation might allow for a significantly higher net present value (NPV) of the forest. Furthermore, the ARM approach gives harvest scheduling models more flexibility in building up treatment units in a variety of ways to meet different forest management objectives. Unfortunately, formulating and solving forest planning problems with ARM constraints is generally considerably more difficult than formulating and solving such problems with URM constraints.

URM constraints can be written in a number of different ways. McDill and Braze (2000) identify 16 different ways URM constraints have been formulated in the literature. The URM problem, which can be stated as selecting a subset of management units from a forest for logging in such a way that no two adjacent units are cut and that the net revenues are maximized, is equivalent to the well-researched maximum weight stable set problem (SSP). Nemhauser and Wolsey (1988, p259-265) provide a detailed discussion of the SSP. The equivalence of URM and SSP is evident if one considers the graph representation of the URM where the nodes correspond to the management units and the arcs represent the adjacency relationships among these units. If the weight assigned to a node represents the net revenues that are earned if the corresponding unit is cut, then the one-period URM problem is to identify a subset of unconnected nodes with maximum total weight. This is the maximum weight stable set problem. This equivalence is easily generalized to the n-period URM problem (Barahona et al. 1990).

There are two important implications of the equivalence of URM and SSP with respect to spatially-explicit harvest scheduling models. One is that harvest scheduling models, both URM and ARM, are *NP*–Hard. In other words, the solution times for these problems increase more than polynomially as a function of the number of constraints and variables that are required to formulate the models. This is because the ARM is a generalization of the URM, and the URM is equivalent to the SSP, which is known to be *NP*-Hard (Nemhauser and Trotter 1974). The other implication is that families of inequalities that have already been found useful for SSPs, such as those based on maximal cliques (Padberg 1973), can be useful for URM problems as well. The concept of maximal cliques – maximal sets of nodes in a graph that are mutually connected by edges – translates to maximal sets of mutually adjacent management units in forest planning. The useful combinatorial properties of maximal clique inequalities in URM problems has been mentioned in Murray and Church (1996a, 1997) and was later utilized by Goycoolea et al. (2005) and Murray et al. (2004) in solving ARM problems.

In contrast to the URM, ARM problems were initially deemed impossible to formulate in a linear model (Murray 1999) and only heuristics were employed to solve them (e.g., Lockwood and Moore 1993, Caro et al. 2003, Richards and Gunn 2003). However, McDill et al. (2002) identified two exact, linear, 0-1 programming formulations of the ARM. Their first formulation uses constraints that allow groups of contiguous management units to be harvested as long as their combined area does not exceed the maximum harvest opening limit. McDill et al. (2002) present an algorithm, which they call the Path Algorithm, that recursively enumerates all sets of contiguous management units whose combined areas just exceed the maximum allowable harvest level. The constraints created this way are similar to cover inequalities in 0-1 knapsack problems (c.f., Wolsey 1998, p147) and thus they are occasionally referred to as cover inequalities in this paper. The disadvantage of the Path/cover formulation is that the number of these constraints can be very large, and this number grows exponentially as the number of times the recursive algorithm that generates them calls itself. Thus, the number of constraints increases exponentially as the ratio between the average size of the management units and the maximum harvest opening size decreases. The advantage of the Path/cover formulation over the two alternatives, discussed next, is that it does not require the introduction of additional 0-1 decision variables. The potentially very large number of Path/cover constraints relative to the number of 0-1 variables suggests that with larger maximum harvest opening sizes, these constraints might be less likely to be binding during optimization. This behavior could be utilized to produce shorter solution times.

McDill et al.'s (2002) other formulation uses separate variables for each possible combination of contiguous management units within the forest whose total area does not exceed the allowable harvest opening size. McDill et al. (2002) refer to these combinations as Generalized Management Units (GMUs). These GMUs need to be enumerated before the model can be constructed. With this formulation, the same types of adjacency constraints as those used in URM models can be written on the set of GMUs. McDill et al. (2002) used pairwise constraints in their initial experiments, whereas Goycoolea et al. (2005) applied maximal cliques and found that these formulations performed better. Additionally, in a more recent work, Goycoolea et al. (2009) also provide theoretical evidence that the maximal clique GMU, or "Cluster," formulation is always at least as tight as the Path formulation in its approximation of the convex hull of ARM. In other words, the linear programming relaxation of the GMU model always leads to an objective function value that is at least as close, or closer to the objective function value of the true optimum as that of the Path model. This is an important result because tighter formulations often lead to shorter solution times. In contrast to the Path formulation, where the number of constraints grows exponentially as the ratio of the maximum harvest opening size is increased, with the GMU model the number of variables grows exponentially as the ratio of the maximum harvest opening size is increased.

The third exact 0-1 programming formulation of ARM, proposed by Constantino et al. (2008), is very different from the Path/Cover and GMU/Cluster formulations in that it does not rely on a recursive, potentially time consuming *a priori* enumeration of spatial constructs such as minimally infeasible (as in the Path Model) or feasible clusters of management units (as in the GMU Model). Since the number of clearcuts in a forest cannot exceed the number of management units (given that a management unit can only be harvested once) a parsimonious set of *clearcut assignment variables* can be defined that represent the decisions to assign management units to a particular clearcut (also referred to as a "bucket" in Goycoolea et al. 2009) in a given planning period. In the context of Constantino et al.'s (2008) model, a clearcut or *bucket* may comprise units that are disconnected. Additional constraints are present in the formulation to ensure that the area of these clearcuts never exceeds the maximum opening size and that two or more clearcuts never overlap and are never adjacent. Since the number of assignment variables in this formulation is bounded by $n \times n \times T$, where n is the number of management units in the forest and T is the number of planning periods, Constantino et al.'s (2008) model leads to smaller problems than the other two formulations when the maximum harvest opening size is large relative to the typical size of a management unit. Further, substantial reductions in problem size can be achieved by eliminating those assignments from the model where the area of the minimum-area path between the two management units involved is greater than the maximum harvest opening size. Constantino et al.'s (2008) model is significant because it keeps the size of ARM from growing exponentially with increasing maximum harvest opening sizes relative to the average unit size.

At least two other ARM constraint sets have been proposed. One can be viewed as an extension of McDill et al.'s (2002) Path model, and the other as a hybrid method that can be solved using exact optimization techniques but cannot guarantee solutions that do not require post-fixing for ARM-feasibility. Crowe et al. (2003) appended what they call "ARM clique constraints" to McDill et al.'s (2002) Path or cover inequalities, arguing that the "clique" concept can be applied to ARM models if the total area of a mutually adjacent set of management units exceeds the maximum opening size. Crowe et al.'s (2003) "clique constraints" are very similar to knapsack constraints, and are written for each mutually adjacent set of units, where the left-hand-side coefficients are the areas of the units and the right-hand-side is the allowable cut limit. Crowe et al. (2003) found that the appended formulation did not outperform McDill et al.'s (2002) Path approach computationally. It can be shown, however, that some of these ARM clique constraints cut off fractional solutions from the LP relaxation defined by McDill et al.'s (2002) Path/Cover formulation, and thus they could possibly be used to tighten the Path/Cover formulation (i.e., better approximate the ARM's integral convex hull). Crowe et al.'s (2003) results illustrate how obtaining a tighter formulation does not necessarily result in improved solution times. While additional constraints may tighten the formulation, they increase the size of the LP relaxation that must be solved at each node in the branch-and-bound tree, slowing down the rate at which nodes are processed.

Gunn and Richards' (2005) "stand-centered" constraints can also be used as an alternative or complement to McDill et al.'s (2002) cover inequalities. One stand-centered constraint is written for each management unit and period. The constraint prevents the harvest of the unit in a given period if the combined area of the adjacent units that are scheduled for harvest in the same period exceeds the cut limit minus the area of the unit. Gunn and Richards (2005) observe that these constraints do not prevent every possible harvest area violation, but they argue that these violations will be few when the areas of management units are not too small compared to the harvest opening area limit and that those that do occur can be easily detected and "post-fixed" at a relatively small loss in optimality. Although Gunn and Richards' (2005) constraint set is not an exact formulation of the ARM, it is attractive because (1) the number of stand-centered constraints needed is equal to the number of units in a forest, which is much less than the number of covers that might be needed, and (2) unlike finding McDill et al.'s (2002) covers, generating stand-centered constraints does not require a potentially very time-consuming recursive enumeration. However, Gunn and Richards' (2005) constraint set can be expected to be less effective as the ratio of the maximum harvest opening limit to the typical management unit size increases.

The goals of this paper are 1) to test empirically whether McDill et al.'s (2002) Path or cover inequalities are often *lazy* in a sense that most of them are rarely active (binding) in otherwise feasible integer solutions that are potential candidates for the true optimum, and 2) to test whether this property can be used to solve area-based harvest scheduling models more efficiently. Specifically, we test whether specifying the Path constraints as a *lazy constraint pool* leads to more efficient solution times (i.e., whether a target dual gap can be achieved more quickly or whether a tighter gap can be achieved within a given amount of time). While the construction of lazy constraint pools still requires the *a priori* enumeration of paths, or minimally infeasible clusters of management units, the constraints in the pool are only applied during optimization if they are violated by a solution that has the potential to improve upon the objective function value of the incumbent. Note that lazy constraints are different from *redundant* constraints in that the latter can never be active in any of the solutions because they are found outside of the feasible region. Lazy constraints are also different from cutting planes because they are required in order to fully identify the set of feasible solutions; without them, an infeasible integer solution would be allowed.

We also note that our proposed approach bears some resemblance to McNaughton and Ryan's (2008) integrated column and constraint generation method. Our method is markedly different in three ways. First, while the lazy constraint approach is applied to the Path Formulation, McNaughton and Ryan's (2008) technique is applied to the Cluster Packing (Goycoolea et al. 2006) or, equivalently to the Generalized Management Unit-based Formulation (McDill et al. 2002). Second, we do generate all of the adjacency constraints, which are in our case path constraints, upfront but use them only when needed during optimization. McNaughton and Ryan (2008) do not generate any of the GMU-based adjacency constraints upfront. However, they enumerate the GMUs and construct the associated GMU variables and constraints only on those GMUs that turn out to be involved in clear-cut size or green-up violations at particular solution candidates. At last but not least, one big advantage of our approach is that all it requires from the user for implementation is to label the path constraints as "lazy". While most of-the-shelf optimization packages, such as IBM's ILOG CPLEX, offers several options to define model constraints as "lazy", the efficacy of the approach in forest planning has not been investigated so far. The McNaughton and Ryan's (2008) approach requires setting up what is essentially a branch-and-cut-and-price algorithm for the ARM, which is a far more technical task.

The next section describes the computational experiment that was conducted to check if the Path constraints are indeed lazy in various problem instances, and to test whether and under what conditions the use of lazy constraint pools leads to shorter solution times compared to other methods. We also give formal, mathematical definitions of the models and algorithms that we used in the comparison.

Methods

The test forests

The "laziness" of the Path constraints and the computational efficiency that can be afforded by the use of lazy constraint pools was tested on sixty hypothetical and eight real forest planning problems, all of which are available in a public data repository at http://ifmlab.for.unb.ca/fmos/ (Integrated Forest Management Lab 2006). Multiple levels of maximum harvest opening size restrictions were used (see Table 3). Thirty of the hypothetical forests had 300 units and thirty had 500 units. The real forests, Kittaning4, FivePoints, PhyllisLeeper, BearTown, Pack, ElDorado, Shulkell and NBCL5 consisted of 32, 71, 89, 90, 186, 1,363, 1,019 and 5,224 units, respectively. In this paper, a management unit is simply the smallest contiguous pre-defined spatial unit that will be treated using a single prescription, i.e., it cannot be split. Adjacent management units may be aggregated, however, to create larger treatment units that will be collectively treated using a single prescription. The hypothetical problems had one forest type and one site class, while some of the real problems had four, five or six forest types and two, three, four or six site classes (Table 2). Forests in different categories exhibit different growth and yield patterns. The initial age-class distribution of the hypothetical forests mimics a typical Pennsylvania hardwood forest (Table 1). As the hypothetical forests comprise different spatial configurations of management units and the acreage of the individual units is predefined, the actual percentages of the age-classes might deviate slightly from the figures in the table. The hypothetical problems were generated in batches using a program called MakeLand (McDill and Braze 2000), which creates hypothetical forests consisting of contiguous irregular polygons that can be assigned different stand characteristics. MakeLand was instructed to randomly assign age-classes to the polygons of each randomly generated forest map in such a way so that the overall age-class distribution would approximate the one shown in Table 1. This random age-class assignment was done three times for each of twenty maps, resulting in the thirty 300-stand and thirty 500-stand problems. Neighborhood adjacency (the average number of adjacent stands, or vertex degree in the adjacency graph) was varied by changing the initial number of points that MakeLand was instructed to use to construct the polygons. The age-classes and yields of each unit in the real problems were based on on-site measurements.

The planning horizon was 60 years for the hypothetical models, and 50, 45, 40 or 25 years for the real problems. The length of the planning periods was 10 years for each problem except for El Dorado, Shulkell and Pack forests, where it was 5 years. The minimum rotation age was 60 for the hypothetical, 80 for the four small real problems from Pennsylvania, 45 for Pack Forest, 35 for El Dorado and Shulkell, and it ranged from 20 to 100 years for NBCL5, depending on the forest type. Since the initial age and the minimum rotation age of a management unit determine whether it can be cut during the planning horizon, and this in turn can have an impact on the difficulty of the harvest scheduling problem, we note that the percentage area of the forests that cannot be cut at all is zero for the majority of the test problems. More specifically, it is zero for the 60 hypothetical problems, Pack Forest and Shulkell and it is 6.18% for Kittaning4, 3.66% for FivePoints, 1.83% for PhyllisLeeper, 0.44% for BearTown, 1.27% for NBCL5 and 20.1% for El Dorado. The financially optimal rotation age, based on maximizing the land expectation value (LEV), was 80 years for the hypothetical, 50 years for the small real problems and Pack Forest, 90 years for NBCL5, 70 for Shulkell and 35 for El Dorado. The possible prescriptions were to cut the management units in period 1, 2, 3, 4, 5, 6 (in the hypothetical forests) or not at all. Maximum harvest opening sizes of 40, 50 and 60 ha were imposed on the hypothetical problems, 40, 50, 60 and 80 ha on the four smallest real problems, 24.28, 32.37, 40.47 and 48.56 ha on Pack Forest, 48.56, 60.70 and 72.84 ha on El Dorado, 40 and 60 ha on Shulkell and 21, 30 and 40 ha on NBCL5. Adjacent management units were allowed to be harvested concurrently as long as their combined area was less than the maximum opening size. All units were smaller than the maximum harvest opening size. In the case of Kittaning4, FivePoints, PhyllisLeeper and BearTown, units greater than 40 ha were divided into smaller units by a Pennsylvania Bureau of Forestry employee using contour lines, roads, trails, streams and shape. In NBCL5 and Shulkell, units greater than 21 and 40 ha, respectively were excluded as we had no site-specific knowledge to make meaningful delineations. We also excluded those units from NBCL5 that had no yield information. The average age of the forests at the end of the planning horizon was set to be at least half of the minimum rotation age. We used a 3% real discount rate for each formulation except for the four Pennsylvania forests where we used 4% and in Pack Forest, where we used 7% as prescribed by the respective administrators. The 3% rate was used to be consistent with Goycoolea et al. (2009).

Table 2 summarizes the spatial characteristics of each real problem, and each hypothetical problem batch. Apart from the minimum, maximum and mean unit sizes, the unit size distribution, the total forest area, as well as the average vertex degrees and the number of forest types, site classes and planning periods are listed.

To evaluate potential solution time savings of the Lazy Path approach, we formulated each problem three different ways: using (1) McDill et al.'s (2002) Path/Cover constraints, (2) Goycoolea et al.'s (2005) maximal clique GMUs (clusters), and (3) Constantino et al.'s (2006) clearcut assignment variables. We used a green-up exclusion period of one period length. This means that depending on whether a 5 or 10-year long planning period was used, 5 or 10 years were assumed to be long enough for a clear-cut to be replanted or naturally regenerated into a new stand that had adequate canopy closure and height. We assumed that adjacent units with a combined area above the maximum opening size can both be cut as long as there is at least one planning period between the two harvests to allow green-up. As a reference for the readers, we note that the length of the exclusion period ranged between 10 and 20% of the financially optimal rotation age in these test problems. We solved the Path formulation with and without treating the Path/Cover inequalities as lazy constraint pools. We did not test the lazy constraint approach with Goycoolea et al.’s (2005) and Constantino et al.’s (2006) models because those formulations don’t require exponentially large constraint pools; they require more variables. Lazy constraint pools are expected to work well only in cases where the number of lazy constraints substantially exceeds the number of variables and where only a few constaints in the lazy constraint pool are likely to be binding. The more constraints there are relative to the number of variables, the less likely that they will all intersect in the neighborhood of a new, potentially optimal solution, hence the “lazy” designation.

The following two sub-sections give formal definitions for each of the models and for each of the preprocessing algorithms that were used in this experiment.

Model formulations

The Path Model (a.k.a. the Cell or Cover Model, McDill et al. 2002)

The general structure of McDill et al.'s (2002) Path Model is as follows:

$$MaxZ=\sum_{m=1}^{M} a_m [c_{m0} x_{m0} + \sum_{t=h_m}^{T} c_{mt} x_{mt}] \quad (1)$$

Subject to:

$$x_{m0} + \sum_{t=h_m}^{T} x_{mt} \leq 1 \quad \text{for } m = 1, 2, ..., M \quad (2)$$

$$\sum_{m \in M_{ht}} v_{mt} \cdot a_m \cdot x_{mt} - H_t = 0 \quad \text{for } t = 1, 2, \ldots T \quad (3)$$

$$b_{l,t} H_t - H_{t+1} \leq 0 \quad \text{for } t = 1, 2, \ldots T\text{-}1 \quad (4)$$

$$-b_{h,t} H_t - H_{t+1} \leq 0 \quad \text{for } t = 1, 2, \ldots T\text{-}1 \quad (5)$$

$$\sum_{j \in C} x_{jt} \leq |C| - 1 \quad \forall C \in \mathbb{C} \text{ and for } t = h_i, \ldots T \quad (6)$$

$$\sum_{m=1}^{M} a_m [(Age_{m0}^{T} - \overline{Age}^{T}) x_{m0} + \sum_{t=h_m}^{T} (Age_{mt}^{T} - \overline{Age}^{T}) x_{mt}] \geq 0 \quad (7)$$

$$x_{mt} \in \{0,1\} \quad \text{for } m = 1, 2, ..., M \text{ and } t = h_m, \ldots T \quad (8)$$

where the variables are:

x_{mt} = 1 if management unit *m* is to be harvested in period *t* for $t = h_m$, ... *T*, 0 otherwise; when $t = 0$, the value of the binary variable is 1 if management unit *m* is not harvested at all during the planning horizon (i.e., x_{m0} represents the "do-nothing" alternative for management unit *m*), and

H_t = the total volume of sawtimber in m^3 harvested in period *t*, and

the parameters are:

h_m = the first period in which management unit *m* is old enough to be harvested,

M = the number of management units in the forest,

T = the number of periods in the planning horizon,

c_{mt} = the net discounted net revenue per hectare plus the discounted expected forest value at the end of the planning horizon if management unit m is harvested in period t,

M_{ht} = the set of management units that are old enough to be harvested in period t,

a_m = the area of management unit m in hectares,

v_{mt} = the volume of sawtimber in m^3/ha harvested from management unit m if it is harvested in period t,

$b_{l,t}$ = a lower bound on decreases in the harvest level between periods t and $t+1$ (where, for example, $b_{l,t}$ = 1 would require non-declining harvests and $b_{l,t}$ = 0.9 would allow a decrease of up to 10%),

$b_{h,t}$ = an upper bound on increases in the harvest level between periods t and $t+1$ (where $b_{h,t}$ = 1 would allow no increase in the harvest level and $b_{h,t}$ = 1.1 would allow an increase of up to 10%),

C = a set of management units, also called a cover or path, that forms a contiguous area just greater in size than the maximum harvest opening limit,

$\mathbb{C}$ = the set of covers (or paths) that arise from a forest planning problem,

h_i = the first period in which the youngest management unit in cover i is old enough to be harvested,

Age_{mt}^{T} = the age of unit m at the end of the planning horizon if it is harvested in period t; and

$\overline{Age}^{T}$ = the minimum average age of the forest at the end of the planning horizon.

Equation (1) specifies the objective function of the problem, namely to maximize the discounted net revenue from the forest during the planning horizon plus the discounted ending value of the forest. Constraints (2) are logical constraints. They require a management unit to be assigned to at most one prescription, including a do-nothing prescription. Harvest variables (x_{mt}) are only created for periods where the stand is old enough to be harvested (i.e., it is older in that period than the predefined minimum rotation age). Constraints (3) are harvest accounting constraints. They sum the harvest volume for each period and assign the resulting value to harvest accounting variables H_t. Constraint sets (4) and (5) are flow constraints. They limit the rate at which the harvest volume can increase or decrease from one period to the next. Constraint set (6) captures the maximum harvest opening size restrictions as minimal cover constraints generated by the Path Algorithm. These constraints assume that the exclusion period equals one planning period: once a management unit, or group of contiguous units, has been harvested, no adjacent management units can be harvested until at least one period has passed. The structure of these constraints is easy to generalize to alternative exclusion periods which are integer multiples of a planning period (see for example, Snyder and ReVelle 1997b). Constraint (7) is an ending age constraint. It requires that the average age of the forest at the end of the planning horizon is at least $\overline{Age}^T$ years. In the real forests with multiple forest types, such as NBCL5, one ending age constraint was written for each forest type. The target ending age was set to one half of the minimum rotation age associated with the forest type. These constraints help prevent the model from over-harvesting the forest during the planning horizon and define a minimum criterion for a desirable ending condition. Lastly, constraint (8) identifies the management unit variables as binary.

The Maximal Clique GMU Model

As discussed in the Introduction, the key step in constructing the maximal clique GMU or Cluster Model is to enumerate each possible combination of contiguous management units within the forest whose total area does not exceed the allowable harvest opening size. The choice variables x_{ut} in this model represent the decision whether all management units in GMU or Cluster u should be cut in period t or not. We note that these variables are defined for t=0 (the "do nothing" option) only if they denote a GMU that consists of one unit. This is necessary to ensure that the minimum average ending age constraint (15) functions as intended. As in Goycoolea et al. (2005), we used maximal clique constraints in this benchmark model to impose the maximum harvest opening restrictions:

$$MaxZ = \sum_{u} a_u [c_{u0} x_{u0} + \sum_{t=h_u}^{T} c_{ut} x_{ut}] \quad (9)$$

Subject to:

$$\sum_{u \in G_m} \left(x_{u0} + \sum_{t=h_u}^{T} x_{ut} \right) \leq 1 \qquad \text{for } m = 1, 2, ..., M \quad (10)$$

$$\sum_{u \in G_t} v_{ut} \cdot a_u \cdot x_{ut} - H_t = 0 \qquad \text{for } t = 1, 2, ... T \quad (11)$$

$$b_{l,t} H_t - H_{t+1} \leq 0 \qquad \text{for } t = 1, 2, ... T\text{-}1 \quad (12)$$

$$-b_{h,t} H_t - H_{t+1} \leq 0 \qquad \text{for } t = 1, 2, ... T\text{-}1 \quad (13)$$

$$\sum_{n \in K_{jt}} x_{nt} \leq 1 \qquad \text{for all } j \in J \text{ and } t = h_j, ..., T \quad (14)$$

$$\sum_{u,t} (Age_{ut}^{T} - \overline{Age}^{T}) \sum_{m \in u} a_m x_{ut} \geq 0 \quad (15)$$

$$x_{ut} \in \{0,1\} \qquad \text{for } \forall u \text{ and } t = h_u, ... T \quad (16)$$

where u = a generalized management unit (GMU or cluster): a set of management units that forms a connected sub-graph of the underlying adjacency graph, for which $\sum_{j \in u} a_j \leq A_{\max}$ (a_j = area of unit *j*, and $A_{\max}$ = maximum harvest limit),

G_m = the set of GMUs that contain management unit *m*,

h_u = the first period in which the youngest management unit in *u* is old enough to be cut,

G_t = the set of GMUs formed by management units that are each old enough to be cut in *t*,

K_{jt} = the set of GMUs that 1) contain at least one unit in maximal clique *j* of management units and 2) where all units comprising the GMU are old enough to be harvested in period *t*. A maximal clique is a set of mutually adjacent management units where no other units exist that are adjacent to all of the units in the clique,

h_j = the first period in which the youngest unit in clique *j* is old enough to be cut,

J = the set of maximal cliques of the management units, and

Age_{ut}^{T} = the age of GMU *u* in years at the end of the planning horizon if it is cut in period *t*.

The Bucket Model

To formulate Constantino et al.'s (2008) Bucket Model, define class K as a class of *clearcuts*. Each clearcut is uniquely indexed by a management unit (stand). Thus, $|\mathrm{K}| = M$, where M is the number of units in the forest. Further, the elements of a clearcut $K_i \in \mathrm{K}$ are management units defined by the following function (0-1 program). Function (11)-(14) assigns a set of units, (which can be the empty set) to each clearcut via the use of binary variables x_m^{it} that take the value of 1 if unit *m* is assigned to clearcut *i* in period *t*. The value of this variable is 0 otherwise.

$$MaxZ = \sum_{m=1}^{M} \sum_{i \in \mathrm{K}} a_m [c_{m0} x_{m0} + \sum_{t=h_m}^{T} c_{mt} x_m^{it}] \qquad (11)$$

Subject to:

$$\sum_{m} x_{m0} + \sum_{t=h_m}^{T} \sum_{i \in \mathrm{K}} x_m^{it} \leq 1 \qquad \text{for } m = 1, 2, ..., M \qquad (12)$$

$$\sum_{m=1}^{M} a_m x_m^{it} \leq A_{\max} \qquad \text{for } i \in \mathrm{K} \text{ and } t = h_m, ... T \qquad (13)$$

$$x_m^{it} \in \{0,1\} \qquad \text{for } i \in \mathrm{K}, m = 1, 2, ..., M \text{ and } t = h_m, ... T \qquad (14)$$

Equation (11), the objective function, is equivalent to Equation (1) in the Path Model. It maximizes the discounted net timber revenues from the forest over the planning horizon plus the discounted ending value of the forest. Constraint set (12) comprises the logical constraints for the Bucket Model. They allow a management unit to be harvested only once in the planning horizon or not at all. Constraints (13) prevent the formation of any clearcut *i* in class K whose area exceeds the maximum harvest opening size. Lastly, constraint set (14) defines variables x_m^{it} as binary.

Note that since constraint set (13) does not prevent clearcuts in class K from being adjacent or overlapping, it alone cannot prevent maximum harvest opening size violations. Additional constraints are necessary. To that end, Constantino et al.'s (2008) model introduces a new set of binary variables of form w_Q^{it} that take the value of one whenever a unit in maximal clique $Q \in \mathbb{Q}$ is assigned to clearcut *i* in period *t*. As with the GMU/Cluster Model, set $\mathbb{Q}$, the set of maximal cliques of management units, must be enumerated during the model formulation phase. The following two constraint sets, along with constraints (13) guarantee that the maximum harvest opening size is never exceeded. The contribution of constraint sets (15)-(16) is to ensure that the units in each maximal clique can only belong to at most one clearcut in any given planning period:

$$x_m^{it} \le w_Q^{it} \qquad \text{for } Q \in \mathbb{Q},\ m \in Q,\ i \le m \text{ and } t = h_m, \ldots T \qquad (15)$$

$$\sum_{i \in \mathrm{K}} w_Q^{it} \le 1 \qquad \text{for } Q \in \mathbb{Q} \text{ and } t = h_m, \ldots T \qquad (16)$$

$$w_Q^{it} \in \{0,1\} \qquad \text{for } i \in \mathrm{K},\ Q \in \mathbb{Q} \text{ and } t = h_m, \ldots T \qquad (17)$$

To account for harvest volumes in each planning period and to ensure a minimum average ending age, we modify constraint set (3) and (7) and add them to the Bucket Model (18-19). The harvest flow constraints are identical to constraint sets (4-5).

$$\sum_{m \in M_{ht}, i \in \mathrm{K}} v_{mt} \cdot a_m \cdot x_m^{it} - H_t = 0 \qquad \text{for } t = 1, 2, \ldots T \qquad (18)$$

$$\sum_{i \in \mathrm{K}} \sum_{m=1}^{M} a_m [(Age_{m0}^T - \overline{Age}^T) x_m^{i0} + \sum_{t=h_m}^{T} (Age_{mt}^T - \overline{Age}^T) x_m^{it}] \ge 0 \qquad (19)$$

The model defined by (11-18) and (4, 5) is identical to what Constantino et al. (2008) refer to as ARMSCV-C. We add a minimum average ending age constraint (19) to this model to prevent the forest from being overharvested. Finally, Constantino et al. (2008) propose a variety of pre-processing techniques that can improve the computational performance of the Bucket Model. We describe the algorithms that we used in a subsequent section titled “Pre-processing”.

The Lazy Path Approach

The Lazy Path approach solves the Path formulation (1-8) by specifying that constraints (6), i.e., the Path/cover inequalities, are placed in a lazy constraint pool. The integer programming solver is instructed to stop at each node in the *branch-and-bound algorithm* where a new feasible solution is found with an objective function value that is better than the current incumbent solution. The solver checks whether the solution at the node violates any of the Path inequalities in the lazy constraint pool. If none of the inequalities are violated and the solution is integer feasible, the solver designates the new solution as the incumbent and proceeds with pruning inferior nodes and processing any remaining unprocessed nodes in the branch-and-bound tree. If none of the inequalities in the lazy pool are violated, but the solution is fractional, the new node remains active for further branching. If, on the other hand, a violation is found, the violated constraints are added to the model and the sub-problem at the node is resolved. If the new solution is still feasible, integer, and has an objective function value that is better than that of the incumbent, then a new incumbent solution is found and, again, the branch-and-bound process is resumed. If the node has an inferior objective function value compared to the current incumbent after the violated constraint(s) has been added, it is pruned from the branch-and-bound tree. If the solution at the node is not integer feasible but still has a superior objective function value to the incumbent, it becomes an unprocessed node, and the branch-and-bound process is resumed. When there are no more nodes to explore, the algorithm terminates at the node that yields the best objective function value without violating any of the Path constraints that remain in the lazy constraint pool. We implemented the Lazy Path approach in IBM ILOG CPLEX 12.1 by using the “Lazy constraints” label for Path inequalities. To estimate how “lazy” the Path constraints were, we kept track of the number of lazy constraint violations that occurred during the course of optimization and these numbers were compared with the number of Path constraints that were needed to fully define the ARM. We note that CPLEX 12.1 offers several options for the user to define or label certain constraints as “lazy”. The options differ based on the modeling environment used, i.e., whether the Concert Technology, the Callable Libraries or other methods were used to access CPLEX.

Pre-processing

Each of the three models above requires pre-processing. The Path model, whether one uses the the Lazy Path approach or not, needs the set of paths or minimal covers to be enumerated before it can be formulated. The Maximal Clique GMU, or Cluster Model, requires the enumeration of both feasible clusters of units (GMUs) and the maximal cliques. The enumeration of maximal cliques is also necessary for the Bucket Model. In addition, the computational performance of the Bucket Model greatly benefits from the elimination of clear-cut assignment variables that can never take the value of one in a feasible solution.

For the simultaneous enumeration of both the clusters (GMUs) and minimal covers, we used "Algorithm I" as proposed by Goycoolea et al. (2009, p164). Following the recommendation in that paper, we utilized special computer programming structures such as hash tables and linked lists to store enumeration results and to check for repetitions. For finding the set of maximal cliques (mutually adjacent management units), we used the following algorithm:

Step 1: Pick a management unit and create a linked list of units that are adjacent to it. As an example, $A_1=\{2,3,5\}$ is the set of units that are adjacent to unit 1. Repeat Step 1 for each stand.

Step 2: Using an adjacency table or matrix that specifies which units are adjacent, check if $A_i \cap A_j = \varnothing$ for each pair of adjacent units *i*, *j* with $i \neq j$. If the intersection is empty, save {*i,j*} as a maximal clique. Otherwise, create a list of 3-member cliques of form {*i,j,k*} for $\forall k \in \{A_i \cap A_j\}$.

Step 3: For each 3-member clique {*i,j,k*}, check if $\left|A_i \cap A_j\right| = 1$. If $\left|A_i \cap A_j\right| = 1$, then save {*i,j,k*} as a maximal clique. Otherwise, create a list of 4-member cliques of form {*i,j,k,l*} for $\forall l \in \left\{A_i \cap A_j\right\}$ with $k \neq l$.

Step 4: For each 4-member clique of form {*i,j,k,l*}, check if $l \in A_k$. If the condition holds (i.e., units *k* and *l* are adjacent), then save {*i,j,k,l*} as a maximal clique.

Step 5: Go through all the saved maximal cliques and discard the redundant ones.

This algorithm could be extended for higher-order cliques (i.e., with more than four elements), but it was not necessary in this case, since adjacency was defined in this paper as sharing a common boundary, not just a point. In this case, the Four Color Theorem (Appel et al. 1977) guarantees that no cliques with more than four elements will exist.

Apart from enumerating the maximal cliques, pre-processing for the Bucket Model involves the identification of clear-cut assignments that can never be part of a feasible solution. For example, a management unit should never be assigned to a particular clear-cut (bucket) if the total area of the minimum area shortest path between this unit and the unit that indexes the clear-cut exceeds the maximum harvest opening size. In this context, paths are defined as contiguous sets of management units that connect a pair of units. Constantino et al. (2008) note that the vast majority of clear-cut assignments can be eliminated via a minimum-weight shortest path algorithm that determines, for each pair of units, whether they can form a feasible clear-cut or not. As an example, the following program, which is a modified version of the standard shortest path model, can, if solved, make such a determination. Given a directed graph representation of the forest, $G(V, E)$, where V is the set of units and E is the set of adjacencies or edges among the units, solve

$$z_{s,t} = min\left(\sum_{i \in V} a_i x_{ij} + a_t : \sum_{j \in A_i} x_{ij} - \sum_{j \in A_i} x_{ji} = \begin{cases} 1 \text{ if } i = s \\ -1 \text{ if } i = t \\ 0 \text{ otherwise} \end{cases} \quad \forall i \in V,\ x_{ij} \geq 0\ \forall ij \in E \right) \qquad (20)$$

for each pair of units $s,t \in V$ ("*s*" stands for source and "*t*" for terminal unit). As before, parameter a_i is the area of unit *i*, and A_i is the set of units adjacent to unit *i*. Variable x_{ij} represents the decision whether directed edge *ij* should be part of the minimum area path between *s* and *t*. If $z_{s,t} \leq A_{\max}$, then an assignment variable for *s* and *t* is necessary, otherwise it isn't. A potentially more efficient alternative that solves the minimum-weight shortest path algorithm for all pairs of units at once is the Floyd-Warshall Algorithm (Roy 1959, Floyd 1962 or Warshall 1962). This recursive, dynamic programming algorithm was used both in Constantino et al. (2008) and in this study to reduce the size of the Bucket formulation for the computational experiment.

The computational experiment

All pre-processing and model formulation tasks were automated using Java and IBM-ILOG CPLEX v. 12.1 Concert Technology (4-thread, 64-bit, released in 2009) on a Power Edge 2950 server that had four Intel Xeon 5160 central processing units at 3.00Gz frequency and 16GB of random access memory. The only exceptions were the Path and Maximal Clique GMU formulations of the Pack Forest problem with the 48.56 ha maximum harvest opening size and the Bucket formulations of NBCL5 and El Dorado. In these cases, a different, more powerful machine was used: a Power Edge 510 with two Intel® Xeon® x5670 CPUs at 2.93Gz frequency and 32GB memory. The operating system was MS Windows Server 2003 R2, Standard x64 Edition with Service Pack 2 (2003) on the Power Edge 2950, and it was MS Windows Server 2008 R2 Standard x64 Edition (2009) on the 510. As shown in the "Results and discussion" section, the fact that for a few problems the formulation times were measured using a faster machine had no impact on our conclusions because these formulation times were longer than those obtained with the alternative models using the slower machine. Finally, we note that the formulation time measurements included computer times that were required to write out the linear programming formulations into text files. The formulation times, the number of constraints and 0-1 variables that ensure the maximum harvest opening size restrictions, as well as the distribution of paths/minimal covers in terms of the number of units they contain are listed in Table 3 and 4 for each of the 68 problems. The information in these tables, along with Table 1 and 2, should allow readers to evaluate the results (e.g., solution times) in the context of the spatial and other attributes of the problems.

Every problem instance was solved on the Power Edge 2950 server with CPLEX 12.1. until a predefined target optimality gap or 6 hours of runtime was reached, whichever happened first. We set the target optimality gaps at three different levels, 1%, 0.05% and the CPLEX default of 0.01% to see how robust the results were with respect to this parameter. The use of a relatively loose 1% gap is illustrative of forest planning exercises where the input data already carries some error, there are simplifications in model development, perhaps because only rough first estimates or strategic benchmarks are sought, and it is not critical to identify accurate solutions. At the other end of the spectrum, model runs with the default gap of 0.01% will demonstrate the power of the proposed lazy approach to generate research-grade solutions that are assumed to be based on high-quality input data. Finally, the goal of the 0.05% runs is to strike balance between these two extremes. We present the 0.05% solutions in more detail and use worst-case analyses and other statistical tools to determine if these results were robust with respect to the 1% and the 0.01% gaps. All solver parameters were set to their default levels except the working memory limit which was set at 1GB. Since CPLEX allows only primal reductions for pre-processing formulations with lazy constraint pools, we set the "Primal and Dual Reduction Type" parameter to 1 (primal reductions only) for the Lazy Path approach. Solution times and constraint activity information for the Lazy Path inequalities (i.e., the number and percentage of lazy constraints that were found to be active during optimization) are listed in Table 5 for the eight real problems.

Results and discussion

The "laziness" of Path/Cover inequalities

On average only 0.20%, 0.33% and 0.54% of the Path/Cover inequalities were found to be active in the hypothetical problems with the 40 ha maximum opening size restriction and with 1%, 0.05% and 0.01% target optimality gaps, respectively (Table 6). The same measures were 0.04%, 0.08% and 0.13% for the same set of problems with 50 ha, and 0.01%, 0.02% and 0.03% with 60 ha maximum harvest opening size. The percentages varied more widely for the real problems (Table 6). While only fractions of a percentage of the constraints were found to be active during optimization for most of the Pack Forest, NBCL5 and El Dorado problems, as many as 23-24% of the constraints were active in some of the PhyllisLeeper or Kittaning4 instances at the 40ha max opening size. With a few exceptions, namely the PhyllisLepper, Kittaning4, FivePoints and BearTown problems with 40 or 50 ha max opening size settings, the Path/Cover inequalities were rarely active in the overwhelming majority of test cases. The activity rate ranged between 0 and 1.47% in the hypothetical and between 0 and 23.81% in the real problems. This empirical result suggests that in many cases only a fraction of the Path constraints might be necessary to find optimal solutions to area-based harvest scheduling problems. Not surprisingly, the results in Table 6 also imply that the larger the maximum harvest opening size, the less likely it is that a given path constraint will be active during optimization. As the evidence in the next section suggests, this implication could in turn lead to significant solution time savings. Before we move on to solution times, we note that the degree of "laziness" could also depend on other factors including the length of the green-up period or on the tightness of harvest flow and minimum average ending age constraints. The longer the green-up and the more relaxed the forest-wide constraints, the more likely it is that a given path constraint becomes active. Lastly, we would also like to point to the result that the proportion of active path constraints increases with tighter optimality gaps. More violations are likely during optimization if more accurate solutions are sought. As we will see, one implication of this result is that the proposed lazy approach is somewhat less effective with tighter optimality gaps.

Solution times

Table 7 lists the number and percent of "wins" for each of the three benchmark models and for the proposed lazy approach for both the real and the hypothetical problems at the pre-specified 1%, 0.05% and 0.01% target optimality gaps. We chose the number and percent of wins as the primary performance metric because not all problems solved to the desired gaps within the predefined 6 hours of runtime. We counted the "wins" based on the number of times a particular model/method solved the problem instance faster than any of the other models. If none of the models/methods were able to find a solution within the preset optimality gap and the 6 hours of runtime, we selected the "winner" based on the tightness of the optimality gap that was achieved. The model that led to the tightest gap for a given instance was considered to be the winner for that particular problem.

We start with the observation that the lazy approach far outperformed the three benchmarks at the 1% and at the 0.05% target optimality gaps for the hypothetical problems. At 1%, it solved 178 of the 180 (98.9%) instances faster than McDill et al.'s (2002) Path, Goycoolea et al.'s (2005) Maximal Clique GMU or Constantino et al.'s (2008) Bucket Model. At 0.05%, the proposed method "won" in 174 of 180 (96.7%) hypothetical cases (Table 7). The computational advantage of the Lazy approach was dramatic: it was at least one magnitude faster than the other methods in solving these problems. While the aggregate solution time at the 0.05% was less than an hour for the Lazy approach, it was more than 53 hours for the Path Model, more than 63 hours for the Bucket and more than 78 hours for the Maximal Clique GMU. And this comparison does not even account for the fact that the Maximal Clique GMU was not able to solve 7 of the hypothetical problems at the target gap of 0.05%. At the 1% target gap, the Lazy approach was also at least one magnitude faster on average, although this advantage was not as dramatic because most hypothetical problems solved in a matter of seconds. Nonetheless, it is worth pointing out that the total solution time was 1.37 minutes with the Lazy approach, while it was 18.65 minutes with the Bucket, more than half an hour with the Path and almost 13 hours with the Maximal Clique GMU. At the 0.01% gap, the advantage of the Lazy method in solving the hypothetical problems was still overwhelming although not as dramatic as it was at 1 or 0.05%. The proposed solution technique led to 99 "wins" out of the 180 hypothetical instances (55%) as opposed to the 21 (11.7%), 40 (22.2%) and 20 (11.1%) "wins" of the Path, Bucket and GMU models, respectively (Table 7). There were 26 cases when the Lazy approach was not able to find an optimal solution within the 0.01% gap in 6 hours. The number of such "timeouts" was 36, 78 and 49 for the Path, Bucket and the GMU models. To further illustrate the advantage of the Lazy approach in the 0.01% gap runs for the hypothetical problems, we created two charts (Fig. 1) that show the percent of "wins" for each approach by maximum harvest opening size and by the number of units. The top chart in Fig. 1 shows that the Bucket model "wins" the largest number of 300-unit instances when the smaller, 40-50 ha opening sizes are applied, but the Lazy approach gains as the opening size is increased and wins the most at the 60 ha opening size. In solving the 500-unit problems, the Lazy approach "wins" the largest number of cases for all opening sizes, and the result is increasingly strong as the opening size is increased (middle chart in Fig. 1). Noteworthy is the Maximal Clique GMU's relatively bad performance despite the fact that theoretical evidence exists that this formulation is tighter than either the Path (Goycoolea et al. 2009) or the Bucket models (Martins et al. 2011). Solution times are functions of both the number of branches that need to be created and processed by the solution algorithm and the complexity of the LP sub-problems. It is possible that the GMU model leads to harder and/or larger sub-problems at the nodes of the branch-and-bound algorithm due to the higher number of variables even though fewer branches might be required to reach the desired level of optimality.

As far as the real problems are concerned, the Lazy Path approach outperformed the other methods in 18 out of the 28 problems (64.3%) at the 1% gap, in 17 out of the 28 problems (60.7%) at the 0.05% and in 19 of the 28 problems (67.9%) at the default 0.01% gap. In the instances where the Lazy approach did not yield the shortest solution times or the tightest optimality gaps, it was almost always the original Path Model that performed the best (Table 7). The Bucket Model never led to better solution times or to better optimality gaps in any of the real problems. The Maximal Clique GMU did solve fastest in two cases (7.1%) of the 0.01% runs (Table 7).

A worst-case performance analysis, applied to all the experimental data we have, provides further evidence that the proposed Lazy approach had a distinct advantage in both the hypothetical and the real problems despite differences in the percentage of "wins". The bottom chart in Fig. 1 shows the proportion of times when each model/method performed the worst by different maximum harvest opening size categories: S (small), M (medium), L (large) and XL (extra-large). It is clear that the Lazy approach has the fewest "worst" performances, and the proportion of "worst" performances decreases as the relative maximum opening size increases. The Bucket model has the highest number of worst performances of all the approaches, regardless of the opening size. Surprisingly, this result gets stronger as the relative opening size increases.

Overall, the results suggest that the Lazy Path approach can improve solution times for area-based harvest scheduling problems - sometimes dramatically. This result appears to be robust regardless of the number and size of the units, the presence or absence of various forest types and site classes, the length of the planning horizon, the maximum harvest opening size, the vertex degree (Table 2) or the cardinality distribution of covers (Table 3). It also appears, especially in the hypothetical problem set, that the Lazy approach is particularly efficient in solving problems with greater maximum harvest opening sizes (Table 1). This is not surprising since the larger the max opening size, the less likely that a given Path constraint becomes active during optimization. It is also clear that in the instances where the Lazy approach was outperformed by the other models (e.g., in Kittaning4, FivePoints, PhyllisLeeper and BearTown – see Table 5), it was the low number of path constraints that was the common denominator (Table 4). Our conjecture, supported by empirical data, that the proposed Lazy approach performs the best when there are a high number of path constraints in the formulation is consistent with the pattern that the advantage of the method increases with greater opening sizes. Greater opening sizes and a greater number of management units both contribute to a higher number of adjacency constraints, which in turn makes it more likely that an individual constraint is lazy in the formulation.

Finally, we like to draw the reader's attention to the apparent lack of correlation between the number of units in a given problem and solution times. The instances that appear to be the most difficult to solve are very small (e.g., PhyllisLeeper or BearTown), whereas the largest models such as NBCL5 solve to the target optimality gaps in seconds. In a sense, this should not come as a surprise as McDill and Braze (2000) have already shown that the initial age-class distribution of a forest also has a role in determining problem difficulty. Further, Vielma et al. (2007) have shown that side constraints, such as volume flow constraints, can also have a significant effect. The idea that problem size (the number of stands is one of the primary determinants of problem size in harvest scheduling models) is only weakly related to problem difficulty is not new. Van Roy and Wolsey (1987) have made this point about mixed-integer programs a long time ago: " *in contrast with linear programming, size is a poor indication of difficulty. We believe that size is perhaps an even less reliable measure for mixed integer programs than it is for pure integer programs.*" (Page 45). We speculate that the reason why some of the smallest problems were the hardest to solve is due to a combination of factors. These factors likely include these forests' over-mature initial age-class distribution, which has been identified by McDill and Braze (2002) as a critical determinant of problem difficulty, and the fact that harvest flow requirements are harder to meet in an optimal fashion if the "volume blocks", i.e., the timber volumes associated with individual stands, are few in number and are large relative to the optimal levels of flow. We believe that the more "volume blocks" are available and the smaller they are relative to the sustainable periodic harvest flows, the easier it will be to find good solutions that satisfy the flow constraints. Since confirming these speculations on an empirical basis would require very large samples, likely thousands of test forests, we leave the question of problem difficulty to future research.

Formulation plus solution times

In this sub-section, we provide an analysis of "total times", the sum of formulation and solution times, to illustrate the role of the proposed Lazy approach in the context of formulating and solving ARM models. We only discuss the results in detail for the compromise 0.05% runs. At 1%, total times are dominated by formulation times because most problems solve very fast to this level of optimality. The Lazy approach does not have an impact on formulation times because it requires that all Path constraints are identified upfront. At 0.01%, the results with respect to total times are very similar to those of the 0.05% runs.

At 0.05%, the Lazy approach still comes out ahead of the other models on average in terms of total times for the hypothetical problems at each of the three maximum harvest size levels that were considered. The results with respect to the real problems are mixed (Table 4, 5). For FivePoints, PhyllisLeeper and BearTown, it was the Path and the Lazy Path approach that allowed the shortest formulation times. The four Kittaning4 instances on the other hand formulated 4-6 times faster with the Bucket Model than with the Path. Since Kittanning4, FivePoints, PhyllisLeeper and BearTown are all very small in size, and they can be formulated in the matter of seconds regardless of which method is used, it is really the solution times that set the alternative formulations apart. While both the Path and the Lazy Path approach solved Kittaning4 and FivePoints in seconds, the Bucket and the Cluster methods took several minutes, or in some cases, several hours of computer time before a solution with the target 0.05% optimality gap was found. Moreover, in one case (Kittaning4 at 80 ha Amax) the Bucket Model was unable to find a solution within the desired optimality gap in six hours of run time. As far as PhyllisLeeper is concerned, neither the Cluster nor the Bucket approach was able to find a solution within the 0.05% gap at any of the four maximum harvest size levels. While the Lazy Path method solved all four of the PhyllisLeeper models to the desired optimality, the original Path Model did so only at the 60 and 80 ha max opening size levels. Finally, none of the models were able to solve the 71-unit BearTown to the 0.05% gap. The tightest gaps were achieved by the Lazy Path approach in three of the four instances and it was the original Path approach that found the best solution for the fourth instance within the 6 hrs pre-specified runtime.

Formulation times ranged from a couple minutes to several days for NBCL5 depending on the maximum harvest opening size and the modeling approach (Table 4). The Maximal Clique GMU/Cluster Model allowed shorter formulation times (~3-11% shorter) than the Path Model for all three max opening sizes for this particular problem. Formulation times were excessive for the 5,224-unit NBCL5 with the Bucket Model even though the Floyd-Warshall Algorithm and other preprocessing techniques, suggested by Constantino et al. (2008), were utilized. While the Path or the Lazy Path approaches both solved the NBCL5 problem instances faster than the Maximal Clique GMU Model, this advantage was offset by the slightly longer formulation times at the 21 and 30 ha max opening size levels. The sum of formulation and solution times were roughly the same for these instances. At 40 ha, both the Path and the Lazy Path methods outperformed the Maximal Clique GMU model when the sum of formulation and solution times were used as the basis of comparison. The sum of formulation and solution times were excessive for the NBCL5 instances, due to the very long formulation times.

For the 186-unit Pack Forest, formulation times increased exponentially with increasing max opening sizes when the Path or the Cluster models were used (Table 4). Compare the 36.53 – 36.65 s formulation times at the 24.28 ha (60 ac) level with the 61.38 – 61.37 days at 48.56 ha (120 ac). The 24.28 ha (60 ac) maximum harvest opening size restriction corresponds to the Forest Stewardship Council's standard in the Pacific Northwest United States, whereas the 48.56 ha (120 ac) coincides with the Sustainable Forest Initiative's and the State of Washington's Forest Practices rules (Washington State Forest Practices Act 2010). With the Bucket Model, formulation times were stable (i.e., not exponentially increasing) and much shorter, except at 24.28 ha, than with the other models. This stability was expected due to the way the Bucket is formulated. Since none of the models could solve the Pack Forest problems to the target 0.05% gap, we were not able to compare the sums of formulation and solution times. In three of the four problems that were created based on four different maximum harvest opening sizes, it was the Lazy Path approach that reached the tightest optimality gaps within the pre-specified 6 hour runtimes (Table 5).

For the 1,363-unit El Dorado and the 1,019-unit Shulkell, formulation times were essentially the same regardless of whether the GMU/Cluster or the Path/Cover model was used. Formulation times ranged from about 25 minutes (at 48.56 ha max opening size) to 65 hours (72.84 ha) for El Dorado and from about 9 minutes (40 ha) to 20 hours (60 ha) for Shulkell (Table 4.). Formulation times were longer for the Bucket Model at the 48.56 and 60.70 ha levels in El Dorado and at the 40 ha level in Shulkell, likely because of the large number of management units involved. On the other hand, the Bucket Model formulated much faster for both problems at the highest, 72.84 and 60 ha maximum opening size levels.

In sum, our empirical results indicate that using lazy constraint pools for McDill et al's (2002) Path inequalities can lead to significant, sometimes dramatic cuts in solution times. Since the use of lazy constraint pools does not eliminate the need of an *a priori* enumeration of Path constraints, the proposed technique can only influence solution but not formulation times. As a result, the Bucket Model, which does not rely on costly enumerations, can outperform the Lazy Path approach in terms of solution plus formulation times in cases (e.g., Shulkell) where the maximum harvest opening size is large relative to the average size of the units and the number of units is not too high (like in NBCL5). Hence, we do not recommend the use of the Lazy Path approach for every single problem instance. We suggest instead that the forest planner tries to formulate the Path and Cluster models as a first step (using Goycooolea et al.'s 2009 Algorithm I) but abandons the process if it appears to be more time-consuming than what his or her timeframe allows. This scenario can occur when the maximum harvest opening size restriction is very large relative to the average size of the management units (see Pack Forest at 48.58 ha max opening size). If that is the case but the number of management units is not too large, then the Bucket Model is likely to be the most efficient choice in terms of formulation plus solution times. If the number of units is also very high (as in NBCL5), the Bucket Model might also become very large and cumbersome to formulate even if efficient pre-processing algorithms such as the Floyd-Warshall are employed. In this particular case, a cutting plane or delayed constraint generation method might be the best approach, where the path constraints are generated only during optimization and only if one or more ARM violations occur in a solution candidate. If the formulation of the Path/Cover/Cell and Cluster models is not too time-consuming, then it is safe to say based on the results of this study that the Lazy Path approach is the best choice to minimize solution times.

Finally, it must be noted that the formulation times reported in the present study should not be considered ironclad. Our goal was to give the reader a feel for the expected computational expense that is associated with formulating these models using the resources of an average analyst. We acknowledge that other programmers could improve these formulation times, perhaps significantly. The question is whether shorter formulation times would have an impact on our conclusions with respect to the performance of the Lazy Path approach. We argue that such an impact is very unlikely for the following reasons. First, since three of the four models that were considered in this study, the Path/Cover, the Lazy Path and the Cluster models all use the same formulation algorithm (Goycoolea et al.'s 2009 Algorithm I), a better computational implementation would have the same impact on all three formulation times. Second, while formulation times for the Bucket Model could potentially be improved to a greater extent than those for the other models, they would have to be improved by several orders of magnitude in order to outperform the Lazy Path approach. This is because the solution times afforded by the Lazy Path method are at least one magnitude shorter than those of the Bucket Model (Table 5).

Caveats

In this sub-section, we discuss a number of additional factors that might have an impact on how useful the proposed Lazy approach can be in solving harvest scheduling problems with area restrictions. As mentioned earlier, the efficacy of the method appears to depend on how lazy the path constraints are in a given formulation. If forest-wide constraints such as even flow or minimum average ending age constraints are present, and these constraints are set tight, it is more likely that a given path constraint is going to be lazy since the model is already very constrained. In practice, it is possible that harvest flow constraints are needed only at a scale broader than the one at which a spatially-explicit harvest scheduling problem is to be optimized. With that in mind, we removed the flow constraints from the 60 hypothetical problems and resolved them using the tightest allowable clear-cut size limit (40 ha) to see if this had any impact on "laziness" and on solution times. We found that the average number of lazy constraints per problem that were active during optimization went up from 71.45 to 719.60 (0.33% of total to 2.85%), which is almost a 10-fold reduction in "laziness". Nonetheless, 60% (36) of these problems still solved faster using the lazy constraints. This is a significant finding considering that the 40 ha max opening size was the tightest of the 3 settings that were used in the experiments. This means that even with the least lazy max opening size setting, the lazy constraint approach still maintained an edge even without even-flow constraints. As far as the impact of the minimum average ending age constraints is concerned, one could argue that these restrictions might force the models to leave old stands uncut during the planning horizon to make sure that the minimum average age is met. This in turn could have an impact on how active the path constraints are in problems that are severely constrained already. Our results for the hypothetical problems suggest, however, that this scenario never materialized. In our models, it was always optimal to cut the stands in the oldest age-classes during the planning horizon.

To illustrate how important (or unimportant) the maximum harvest opening size constraints were in restricting the forest managers' ability to maximize discounted timber revenues, we resolved the test problems at the 0.05% gap without path constraints. The percent reductions in NPV due to maximum clear-cut sizes are reported in the rightmost column of Table 5. The average cost of adjacency was a fraction of a percent for the hypothetical problems and it was less than 1% for most of the real problems. In a few real problems, however, as in FivePoints or Kittaning4 with 40ha max opening sizes, the cost was much higher at 11.89% and 7.78%, respectively. The cost of adjacency dropped rapidly as the max opening size was raised. The fact that the Lazy approach solved the FivePoints the fastest at 40 ha, but the original Path method was the best for Kittaning4 suggests that there might not be a strong correlation between the cost of adjacency and the efficacy of the Lazy method.

Conclusions

In this article, we showed empirically that the Path/Cover inequalities of McDill et al.'s (2002) Path formulation of the Area Restriction Model (Murray 1999) are often lazy. We exploited this property by removing these inequalities from the harvest scheduling model and placing them in a "lazy constraint pool". Each time the solver finds a potential solution it checks if any of the constraints in the pool is violated. If a lazy constraint is violated, we add it to the model. The process is repeated until the desired optimality gap is reached and no more violations occur. We tested the technique on sixty hypothetical and eight real problem instances with varying maximum harvest opening sizes and found that in most cases it outperformed the other three existing models in terms of solution times, often by a dramatic margin. An additional finding was that if the sum of formulation and solution times was used as a measure of efficiency, the Lazy Path approach still came out ahead of the other models on average.

In conclusion, we emphasize that while the Lazy Path approach offers significant improvements in solution times, it does not allow reductions in formulation times. The proposed technique still requires the complete enumeration of Path/Cover constraints prior to optimization, and as we have seen, this process can be extremely time-consuming. For future research, we plan to develop a cutting plane or delayed constraint generation technique that will enumerate a Path/Cover constraint only if a maximum harvest size violation is detected during optimization.

References

American Forest & Paper Association (AF&PA). 2000. Sustainable Forestry Initiative Standard-2000 Edition. Washington, DC. 5 p.

Appel, K., W. Haken, and J. Koch. 1977. Every Planar Map is Four Colorable. *Illinois Journal of Mathematics* 21: 439–567

Barahona, F., A. Weintraub, and R. Epstein. 1990. Habitat dispersion in forest planning and the stable set problem. Oper. Res. 40 (Suppl. 1): S14-S22.

Barrett, T. M., J. K. Gilles, and L. S. Davis. 1998. Economic and fragmentation effects of clearcut restrictions. For. Sci. 44: 569-577.

Borges, J. G., and H. M. Hoganson. 2000. Structuring a landscape by forestland classification and harvest scheduling spatial constraints. For. Ecol. Manage. 130: 269-275.

Boston, K., and P. Bettinger. 2002. Combining tabu search and genetic algorithm heuristic techniques to solve spatial harvest scheduling problems. For. Sci. 48(1): 35-46.

Caro, F., M. Constantino, I. Martins, and A. Weintraub. 2003. A 2-opt tabu search procedure for the multiperiod forest harvesting problem with adjacency, greenup, old growth, and even flow constraints. For. Sci. 49(5): 738-751.

Carter, D. R., M. Vogiatzis, C. B. Moss, and L. G. Arvanitis. 1997. Ecosystem management or infeasible guidelines? Implications of adjacency restrictions for wildlife habitat and timber production. Can. J. For. Res. 27: 1302-1310.

Constantino, M., I. Martins, and J. Borges. 2008. A new mixed-integer programming model for harvest scheduling subject to maximum area restrictions. Oper. Res. 56(3): 542-551.

Crowe, K., J. Nelson, and M. Boyland. 2003. Solving the area-restricted harvest-scheduling model using the branch and bound algorithm. Can. J. For. Res. 33: 1804-1814.

Floyd, Robert W. (June 1962). "Algorithm 97: Shortest Path". Communications of the ACM 5 (6): 345.

Franklin, J. F., and R. T. Forman. 1987. Creating landscape patterns by forest cutting: Ecological consequences and principles. Land. Ecol. 1: 5-18.

Goycoolea, M., A. T. Murray, J.P. Vielma, and A. Weintraub. 2009. Evaluating alternative approaches for solving the area restriction model in harvest scheduling. For. Sci. 55(2): 149-165.

Goycoolea, M., A. T. Murray, F. Barahona, R. Epstein, and A. Weintraub. 2005. Harvest scheduling subject to maximum area restrictions: exploring exact approaches. Oper. Res. 53(3): 490-500.

Gunn, E. A., and E. W. Richards. 2005. Solving the adjacency problem with stand-centered constraints. Can. J. For. Res. 35: 832-842.

Harris, L. D. 1984. The fragmented forest: Island biogeography theory and the preservation of biotic diversity. The University of Chicago Press. Chicago, IL. 211 p.

IBM ILOG Inc. 2009. CPLEX 12.1 Reference Manual. IBM ILOG Documentation.

Forest Management Optimization Site. 2006. Integrated Forest Management Lab, University of New Brunswick, Fredericton, New Brunswick, Canada. Available online at http://ifmlab.for.unb.ca/fmos/datasets/. Last accessed: April 15, 2011.

Jones, J. G., B. J. Meneghin, and M. W. Kirby. 1991. Formulating adjacency constraints in linear optimization models for scheduling projects in tactical planning. For. Sci. 37(5): 1283-1297.

Lockwood, C., and T. Moore. 1993. Harvest scheduling with spatial constraints: a simulated annealing approach. Can. J. For. Res. 23: 468-478.

Martins, I., F. Alvelos, and M. Constantino. 2011. A branch-and-price approach for harvest scheduling subject to maximum area restrictions. Comput. Optim. Appl. DOI: 10.1007/s10589-010-9347-1.

McDill, M. E., and J. Braze. 2000. Comparing adjacency constraint formulations for randomly generated forest planning problems with four age-class distributions. For. Sci. 46(3): 423-436.

McDill, M. E., S. Rebain, and J. Braze. 2002. Harvest scheduling with area-based adjacency constraints. For. Sci. 48(4): 631-642.

McNaughton, A.J. and D. Ryan. 2008. Adjacency branches used to optimize forest harvesting subject to area restrictions on clearfell. *Forest Science*. 54(4):442-454.

Meneghin, B. J., M. W. Kirby, and J. G. Jones. 1988. An algorithm for writing adjacency constraints efficiently in linear programming models. The 1988 Symposium on Systems Analysis in Forest Resources, USDA Forerst Service.

Murray, A. T. 1999. Spatial restrictions in harvest scheduling. For. Sci. 45(1): 45-52.

Murray, A. T., and R. L. Church. 1995. Measuring the efficacy of adjacency constraint structure in forest planning models. Can. J. For. Res. 25: 1416-1424.

Murray, A. T., and R. L. Church. 1996b. Constructing and selecting adjacency constraints. INFOR 34(3): 232-248.

Murray, A. T., and R. L. Church. 1996a. Analyzing cliques for imposing adjacency restrictions in forest models. For. Sci. 42(2): 166-175.

Murray, A. T., M. Goycoolea, and A. Weintraub. 2004. Incorporating average and maximum area restrictions in harvest scheduling models. Can. J. For. Res. 34: 456-464.

Nemhauser, G.L., and L.A. Wolsey. 1988. Integer and combinatorial optimization. Wiley. New York xiv, 763 p.

Nemhauser, G., and L. Trotter. 1974. Properties of vertex packing and independence system polyhedra. Mathematical Programming 6: 48-61.

Padberg, M. 1973. On the facial structure of set packing polyhedra. Mathematical Programming 5: 199-215.

Rebain, S., and M. E. McDill. 2003a. Can mature patch constraints mitigate the fragmenting effect of harvest opening size restrictions? Int. Trans. Oper. Res. 10(5): 499-513.

Rebain, S., and M. E. McDill. 2003b. A mixed-integer formulation of the minimum patch size problem. For. Sci. 49(4): 608-618.

Richards, E. W., and E. A. Gunn. 2003. Tabu search design for difficult forest management optimization problems. Can. J. For. Res. 33: 1126-1133.

Roy, B. 1959. Transitivité et connexité.. C. R. Acad. Sci. Paris 249: 216–218.

Snyder, S., and C. ReVelle. 1996a. Temporal and spatial harvesting of irregular systems of parcels. Can. J. For. Res. 26: 1079-1088.

Snyder, S., and C. ReVelle. 1996b. The grid packing problem: Selecting a harvest pattern in an area with forbidden regions. For. Sci. 42(1): 27-34.

Snyder, S., and C. ReVelle. 1997a. Multiobjective grid packing model: an application in forest management. Loc. Sci. 5(3): 165-180.

Snyder, S., and C. ReVelle. 1997b. Dynamic selection of harvests with adjacency restrictions: The SHARe Model. For. Sci. 43(2): 213-222.

Thompson, E. F., B. G. Halterman, T. J. Lyon, and R. L. Miller. 1973. Integrating timber and wildlife management planning. The Forestry Chronicle: 247-250.

Thompson, W., M. Halme, S. Brown, I. Vertinsky, and H. Schreier. 1994. Timber harvest scheduling subject to wildlife and adjacency constraints. Symposium on Systems Analysis in Forest Resources - Management Systems for a Global Economy with Global Resource Concerns, Pacific Grove, CA, Society of American Foresters.

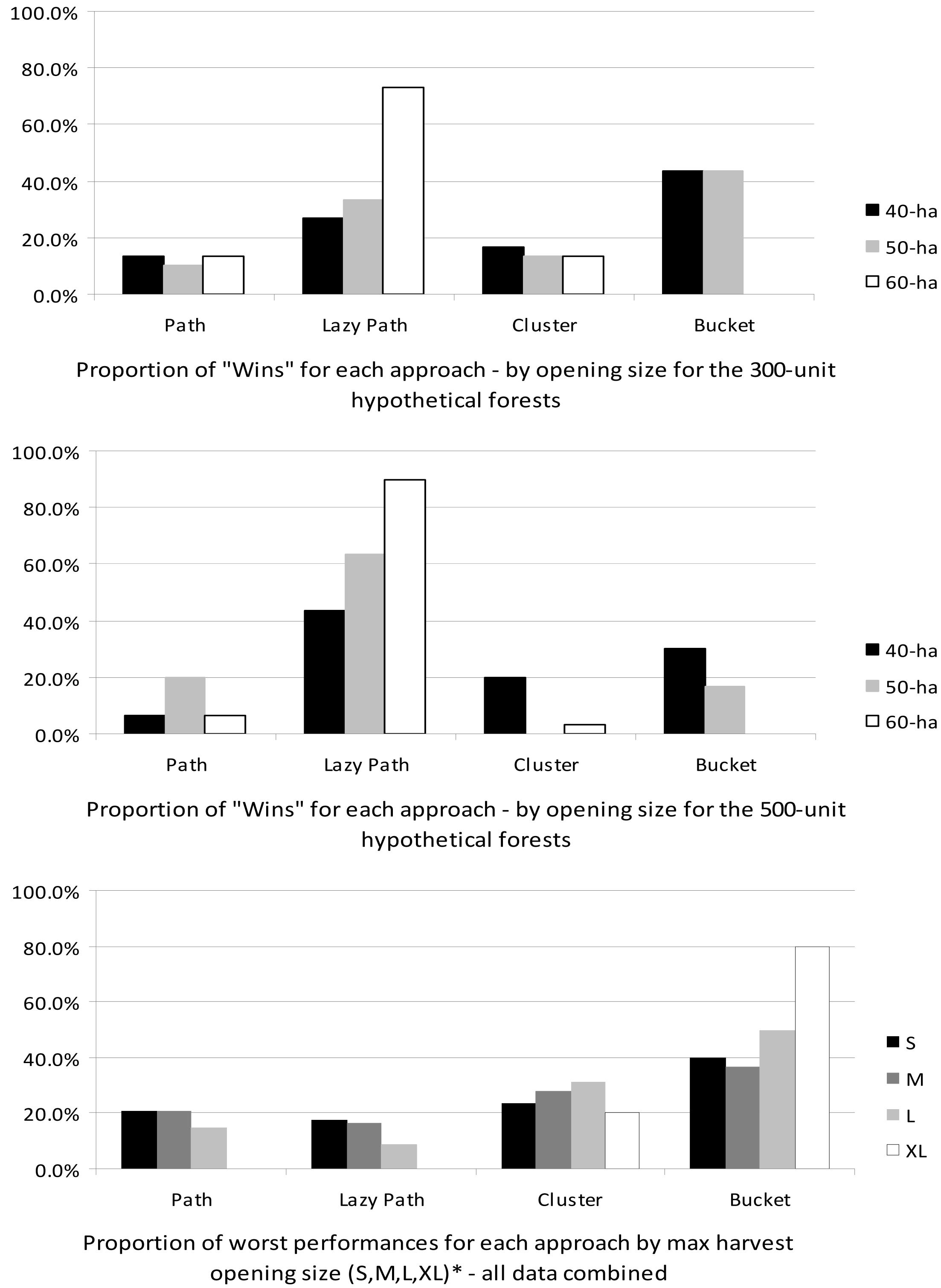
Tóth, S.F., M.E. McDill, N. Könnyű and S. George. 2012. A Strengthening Procedure for the Path Formulation of the Area-based Adjacency Problem in Harvest Scheduling Models. Mathematical and Computational Forestry & Natural-Resource Sciences 4(1):16-38.

Warshall, S. 1962. A theorem on Boolean matrices. Journal of the ACM 9 (1): 11–12.

Washington State Forest Practices Act. 2010. Washington State Department of Natural Resources. Chapter 76.09 RCW. Available online at http://www.dnr.wa.gov/Publications/fp_rules_76.09.pdf. Last accessed: April 15, 2011.

Wolsey, L. A. 1998. Integer programming. J. Wiley. New York. xviii, 264 p.

Figure 1. Best- and worst-case performance analysis - 0.01% target gap runs



*(S,M,L)=(40,50,60 ha) for the hypothetical forests, =(21,30,40 ha) for NBCL5, =(48.56,60.70,72.84 ha) for El Dorado; (S,L)=(40,60 ha) for Shulkell, and (S,M,L,XL)=(40,50,60,80 ha) for Kittaning4, FivePoints, PhyllisLeeper and BearTown and =(24.28,32.37,40.47,48.56 ha) for Pack Forest.

Table 1. Initial age-class distribution and yield table for the hypothetical forests

	Age-classes	Total Area (%)
1	0-10	8
2	11-20	8
3	21-30	3
4	31-40	3
5	41-50	2
6	51-60	2
7	61-70	13
8	71-80	13
9	81-90	24
10	91-100	24
Sum		100

Stand age	Yield (MBF/ha)	Annual value growth rate
10	0.0	N/A
20	0.0	N/A
30	3.7	N/A
40	12.4	0.1279
50	29.7	0.0915
60	61.8	0.0762
70	103.2	0.0526
80	144.6	0.0343
90	188.4	0.0269
100	232.3	0.0211
110	269.3	0.0149
120	306.4	0.0130
130	333.6	0.0085
140	360.8	0.0079
150	381.8	0.0057

Table 2. Test problem characteristics

	Problem IDs	Management unit size distribution (ha) 0-5	6-10	11-15	16-20	21-25	26-30	31-35	36-50	Total	Area (ha)	Unit size Min	Max	Mean	Planning Periods	Vertex^{o}	Forest Types	Site Classes
Real problems	Pack Forest, WA	62	56	31	16	21	0	0	0	186	1,708	0.55	24.27	9.18	9X5yrs	4.78	1	1
	NBCL5, Canada	2,833	1,577	623	211	0	0	0	0	5,244	34,739	0.99	20.23	6.65	4X10yrs	2.87	6	1
	El Dorado, CA	107	421	267	183	134	94	88	69	1,363	21,147	4.05	47.09	15.52	5X5yrs	5.30	1	1
	Shulkell, Nova Scotia	299	377	188	67	49	16	17	6	1,019	9,443	0.31	39.33	9.27	5X5yrs	4.05	6	6
	Kittaning4, PA	1	3	4	13	5	6	0	0	32	588	4.02	29.32	18.38		3.27	4	2
	FivePoints, PA	0	15	19	10	26	14	6	0	90	1,673	5.80	31.75	18.58	5X10yrs	3.71	5	4
	PhyllisLeeper, PA	6	3	15	30	21	13	1	0	89	1,597	1.25	30.46	17.95		3.19	5	3
	BearTown, PA	0	7	11	20	19	13	1	0	71	1,349	5.96	30.81	19.00		2.90	5	3
300-unit hypothetical problems	75-77	0	147	80	38	20	9	4	2		3,600	5.39	38.25	12.00		4.63		
	81-83	0	135	101	35	17	9	2	1		3,600	5.61	38.84	12.00		5.03		
	87-89	0	132	108	35	11	11	3	0		3,600	5.78	32.56	12.00		4.95		
	90-92	0	143	79	46	20	6	6	0		3,600	5.20	35.00	12.00		4.87		
	93-95	0	130	101	43	17	7	2	0	300	3,600	5.60	33.86	12.00	6X10yrs	4.93	1	1
	96-98	0	133	98	43	13	12	1	0		3,600	5.95	31.27	12.00		5.00		
	99-101	0	140	85	48	18	5	3	1		3,600	5.86	35.51	12.00		4.99		
	102-104	0	141	85	38	24	5	4	3		3,600	5.15	38.89	12.00		4.69		
	189-191	0	143	104	31	15	3	1	3		3,480	5.59	38.56	11.60		5.06		
	192-194	0	156	84	37	15	5	2	1		3,480	5.91	39.29	11.60		5.25		
500-unit hypothetical problems	108-110	0	233	170	45	34	12	6	0		6,000	5.56	34.98	12.00		4.94		
	111-113	0	241	151	72	21	4	9	2		5,725	5.15	39.97	11.45		4.79		
	120-122	0	189	161	82	33	22	9	4		6,750	6.93	39.79	13.50		5.29		
	135-137	0	295	122	58	13	7	2	3		5,300	5.40	39.31	10.60		5.28		
	141-143	0	242	164	56	19	9	10	0	500	5,800	5.82	34.89	11.60	6X10yrs	5.36	1	1
	144-146	0	256	142	48	33	15	5	1		5,800	5.70	35.67	11.60		5.44		
	150-152	0	299	131	39	20	5	3	3		5,300	5.43	39.87	10.60		5.50		
	153-155	0	280	146	55	11	5	2	1		5,300	5.37	35.20	10.60		5.47		
	159-161	31	270	126	53	14	4	1	1		5,000	4.78	38.73	10.00		5.46		
	168-170	0	209	150	88	29	14	9	1		6,300	6.35	36.34	12.60		5.46		

Table 3. Test problem formulation characteristics: cover / path size distribution

	Problem IDs	A_{max} (ha)	Cardinality distribution of covers/paths 15	14	13	12	11	10	9	8	7	6	5	4	3	2	Total
Real problems	Pack Forest WA	24.28	0	0	0	0	0	5	72	201	640	828	620	386	212	161	3,125
		32.37	0	0	1	18	304	1,063	2,908	4,305	4,020	2,349	1,175	477	302	68	16,990
		40.47	62	311	2,166	5,632	13,924	21,573	22,659	16,469	8,708	3,652	1,664	838	316	14	97,988
		48.56	3,212*	12,586	35,507	77,330	111,530	129,198	109,547	68,371	33,022	12,938	5,707	2,613	942	175	603,419
	NBCL5, Can.	21.00	0	0	0	0	0	0	0	0	62	463	1,867	4,148	3,749	1,566	11,855
		30.00	0	0	0	0	0	1	193	990	3,328	8,151	11,058	8,413	3,021	163	35,318
		40.00	0	0	0	26	528	2,620	9,051	27,885	41,540	34,432	20,893	6,554	537	0	144,066
	El Dorado, CA	48.56	0	0	0	0	0	0	0	3	536	3,476	6,626	6,205	3,127	657	20,630
		60.70	0	0	0	0	0	0	156	3,424	13,335	20,240	17,543	9,859	2,966	193	67,716
		72.84	0	0	0	0	36	1,958	18,700	47,749	65,734	54,688	31,672	10,609	1,971	18	233,135
	Shulkell, NS	40.00	0	0	5	44	105	183	290	790	1,674	3,014	3,603	2,042	845	378	12,973
		60.00	755**	1,626	2,747	2,782	5,328	12,870	21,376	26,141	22,532	13,902	6,798	2,066	394	181	119,734
	Kittaning4 WA	40.00	0	0	0	0	0	0	0	0	0	0	0	1	44	15	60
		50.00	0	0	0	0	0	0	0	0	0	0	0	4	63	1	68
		60.00	0	0	0	0	0	0	0	0	0	0	2	68	28	0	98
		80.00	0	0	0	0	0	0	0	0	0	7	117	33	0	0	157
	FivePoints WA	40.00	0	0	0	0	0	0	0	0	0	0	0	9	74	80	163
		50.00	0	0	0	0	0	0	0	0	0	0	4	41	188	28	261
		60.00	0	0	0	0	0	0	0	0	0	1	27	160	192	2	382
		80.00	0	0	0	0	0	0	0	0	5	84	278	462	26	0	855
	PhyllisLeeper WA	40.00	0	0	0	0	0	0	0	0	0	0	0	3	72	59	134
		50.00	0	0	0	0	0	0	0	0	0	0	4	17	201	8	230
		60.00	0	0	0	0	0	0	0	0	0	0	26	133	130	0	289
		80.00	0	0	0	0	0	0	0	0	2	35	359	290	0	0	686
	BearTown WA	40.00	0	0	0	0	0	0	0	0	0	0	0	0	58	47	105
		50.00	0	0	0	0	0	0	0	0	0	0	0	14	123	8	145
		60.00	0	0	0	0	0	0	0	0	0	0	5	91	99	0	195
		80.00	0	0	0	0	0	0	0	0	0	12	226	166	3	0	407
300-unit hypothetical problems	75-77	40.00	0	0	0	0	0	0	0	0	34	496	1,055	1,414	704	64	3,767
		50.00	0	0	0	0	0	0	0	210	1,485	2,892	3,764	2,230	350	12	10,943
		60.00	0	0	0	0	0	5	754	4,095	7,891	10,051	6,614	1,523	116	4	31,053
	81-83	40.00	0	0	0	0	0	0	0	0	0	172	1,126	2,228	837	49	4,412
		50.00	0	0	0	0	0	0	0	3	509	2,905	5,960	3,164	317	10	12,868
		60.00	0	0	0	0	0	0	18	1,382	8,090	15,955	11,177	1,893	87	3	38,605
	87-89	40.00	0	0	0	0	0	0	0	0	0	625	1,191	2,458	747	47	5,068
		50.00	0	0	0	0	0	0	0	166	1,960	3,916	6,007	2,808	340	6	15,203
		60.00	0	0	0	0	0	0	1,364	5,565	13,879	16,681	9,595	1,966	79	1	49,130
	90-92	40.00	0	0	0	0	0	0	0	0	0	206	1,067	1,891	650	68	3,882
		50.00	0	0	0	0	0	0	0	8	569	3,043	4,761	2,069	444	12	10,906
		60.00	0	0	0	0	0	0	76	1,650	8,376	12,509	6,897	1,944	165	0	31,617
	93-95	40.00	0	0	0	0	0	0	0	0	0	64	1,623	2,292	737	46	4,762
		50.00	0	0	0	0	0	0	0	0	295	4,075	6,461	2,670	348	5	13,854
		60.00	0	0	0	0	0	0	0	1,092	10,604	18,805	9,581	1,807	87	0	41,976
	96-98	40.00	0	0	0	0	0	0	0	0	0	208	1,393	2,193	829	45	4,668
		50.00	0	0	0	0	0	0	0	0	742	3,393	5,788	2,829	401	4	13,157
		60.00	0	0	0	0	0	0	97	2,631	8,855	15,742	9,817	2,203	72	0	39,417
	99-101	40.00	0	0	0	0	0	0	0	0	0	99	1,379	2,512	727	47	4,764
		50.00	0	0	0	0	0	0	0	0	397	3,894	6,868	2,778	307	8	14,252
		60.00	0	0	0	0	0	0	10	1,519	11,704	20,057	9,802	1,622	90	2	44,806
	102-104	40.00	0	0	0	0	0	0	0	0	12	45	767	1,694	656	71	3,245
		50.00	0	0	0	0	0	0	0	45	145	1,807	4,217	2,236	361	16	8,827
		60.00	0	0	0	0	0	1	68	494	5,074	10,673	7,354	1,513	146	3	25,326
	189-191	40.00	0	0	0	0	0	0	0	0	0	175	2,355	2,890	627	45	6,092
		50.00	0	0	0	0	0	0	0	0	875	7,209	8,317	2,725	213	13	19,352
		60.00	0	0	0	0	0	0	16	4,215	23,064	25,753	10,515	1,130	106	2	64,801
	192-194	40.00	0	0	0	0	0	0	0	0	0	408	3,392	2,611	693	52	7,156
		50.00	0	0	0	0	0	0	0	3	2,351	10,408	8,096	2,808	299	8	23,973
		60.00	0	0	0	0	0	0	115	11,656	33,098	26,782	10,438	1,692	112	0	83,893
500-unit hypothetical problems	108-110	40.00	0	0	0	0	0	0	0	0	0	120	2,113	3,395	1,297	93	7,018
		50.00	0	0	0	0	0	0	0	1	477	5,445	9,086	4,596	637	12	20,254
		60.00	0	0	0	0	0	0	14	1,732	14,482	25,964	15,732	3,090	189	1	61,204
	111-113	40.00	0	0	0	0	0	0	0	0	16	340	2,686	3,363	984	92	7,481
		50.00	0	0	0	0	0	0	0	56	1,586	7,087	9,140	3,700	454	23	22,046
		60.00	0	0	0	0	0	11	497	5,506	20,151	25,758	12,466	2,341	191	3	66,924
	120-122	40.00	0	0	0	0	0	0	0	0	0	7	603	3,398	1,471	199	5,678
		50.00	0	0	0	0	0	0	0	0	11	1,176	7,743	4,859	1,078	54	14,921
		60.00	0	0	0	0	0	0	0	17	2,557	17,450	14,754	4,852	536	8	40,174
	135-137	40.00	0	0	0	0	0	0	0	0	227	3,245	7,019	4,782	1,087	58	16,418
		50.00	0	0	0	0	0	0	7	2,711	13,440	23,861	16,291	4,920	320	9	61,559
		60.00	0	0	0	0	0	1,035	14,946	57,353	80,853	61,841	19,561	2,078	93	0	237,760
	141-143	40.00	0	0	0	0	0	0	0	0	0	513	4,033	4,986	1,169	101	10,802
		50.00	0	0	0	0	0	0	0	1	2,052	11,468	14,684	4,902	675	12	33,794
		60.00	0	0	0	0	0	0	63	7,723	33,014	44,575	19,502	3,637	146	3	108,663
	144-146	40.00	0	0	0	0	0	0	0	0	2	926	4,564	4,729	1,423	79	11,723
		50.00	0	0	0	0	0	0	0	90	3,489	14,110	14,962	5,778	566	14	39,009
		60.00	0	0	0	0	0	0	892	13,832	44,865	49,854	22,099	3,249	203	0	134,994
	150-152	40.00	0	0	0	0	0	0	0	0	1	3,299	10,800	5,633	819	71	20,623
		50.00	0	0	0	0	0	0	0	323	18,732	38,426	19,319	3,584	439	15	80,838
		60.00	0	0	0	0	0	0	6,213	95,252	139,898	69,352	15,972	2,525	146	0	329,358
	153-155	40.00	0	0	0	0	0	0	0	0	33	3,448	10,482	6,156	920	41	21,080
		50.00	0	0	0	0	0	0	0	628	18,197	37,199	20,878	4,507	293	4	81,706
		60.00	0	0	0	0	0	30	7,588	83,597	136,886	77,729	19,441	1,907	60	0	327,238
	159-161	40.00	0	0	0	0	0	0	0	13	1,371	7,076	11,369	5,785	818	34	26,466
		50.00	0	0	0	0	0	0	500	11,425	31,552	41,593	20,841	3,848	236	7	110,002
		60.00	0	0	0	0	15	8,550	65,482	143,837	154,467	78,924	17,876	1,460	72	0	470,683
	168-170	40.00	0	0	0	0	0	0	0	0	0	53	1,683	3,717	1,807	117	7,377
		50.00	0	0	0	0	0	0	0	0	245	3,957	9,772	6,452	948	16	21,390
		60.00	0	0	0	0	0	0	1	698	9,994	25,526	21,072	5,199	244	3	62,737

*: At Amax = 48.56 ha, Pack Forest has an additional 684 16-, 56 17- and one 18-unit cover **: At Amax = 60 ha, Shulkell has an additional 223 16-, and 23 17-unit cover

Table 4. Test problem formulation characteristics: problem size and formulation time

	Problem IDs	A_{max} (ha)	No. of 0-1 variables			No. of ARM constraints			Average formulation time (sec.)		
			CLUSTER	PATH	BUCKET	CLUSTER	PATH	BUCKET	CLUSTER	PATH	BUCKET
Real problems	Pack Forest WA	24.28	54,570	1,216	7,181	1,733	7,872	33,689	36.65	36.53	104.78
		32.37	344,110		12,626	1,808	34,302	59,695	2,752.73	2,846.41	135.68
		40.47	2,171,170		19,229	1,810	170,232	91,063	162,727.63	164,079.31	121.32
		48.56	15,643,562		26,178	1,818	924,133	122,709	5,303,282.04	5,302,396.93	176.00
	NBCL5, Can.	21.00	109,630	23,422	75,096	15,397	32,691	164,920	171.98	182.14	210,716.77
		30.00	337,965		126,431	15,495	88,248	277,596	2,829.54	2,931.58	789,070.12
		40.00	1,360,425		187,363	15,507	326,796	407,233	74,514.28	83,743.45	949,889.21
	El Dorado, CA	48.56	128,466	8,184	38,032	9,209	54,024	187,059	1,515.96	1,499.91	56,252.90
		60.70	426,012		55,780	9,230	164,401	274,575	20,144.21	20,160.48	64,383.68
		72.84	1,508,178		77,413	9,230	521,154	379,145	233,878.77	234,218.79	39,149.23
	Shulkell, NS	40.00	155,016	6,240	27,361	5,368	50,562	97,470	585.50	548.87	68,264.58
		60.00	1,726,446		65,827	5,370	425,760	222,448	72,297.81	72,823.93	24,555.11
	Kittaning4 WA	40.00	414	150	311	98	147	731	3.16	1.42	0.44
		50.00	594		420	101	165	954	3.02	1.38	0.22
		60.00	882		533	101	192	1,193	2.77	1.38	0.23
		80.00	1,836		734	101	307	1,553	2.52	1.38	0.30
	FivePoints WA	40.00	1,200	390	841	313	450	2,431	3.01	1.34	2.84
		50.00	1,914		1,260	324	653	3,626	2.49	1.34	3.47
		60.00	2,994		1,704	324	941	4,804	2.52	1.33	4.47
		80.00	6,960		2,566	324	1,847	7,057	2.85	1.59	6.64
	PhyllisLeeper WA	40.00	1,104	509	947	435	577	2,499	2.58	1.23	2.49
		50.00	1,734		1,452	440	951	3,977	2.36	1.22	3.67
		60.00	2,688		1,944	440	1,126	5,175	2.69	1.23	4.83
		80.00	6,474		3,020	440	2,421	7,774	2.68	1.44	9.79
	BearTown WA	40.00	756	411	718	325	487	1,787	2.61	1.49	3.08
		50.00	1,182		1,102	325	661	2,679	3.00	1.48	3.39
		60.00	1,668		1,427	325	883	3,416	2.77	1.48	4.11
		80.00	3,630		2,319	325	1,830	5,288	3.41	1.56	6.19
300-unit hypothetical problems	75-77	40.00	13,159	1,832	13,074	2,250	12,160	13,074	9.74	7.26	297.19
		50.00	68,859		21,928	2,253	31,510	13,599	119.12	113.10	1,493.94
		60.00	198,807		32,330	2,253	81,057	13,751	1,669.52	1,654.85	2,532.97
	81-83	40.00	13,237	1,823	13,220	2,578	13,954	13,220	9.83	7.76	285.69
		50.00	67,480		22,282	2,581	36,177	16,428	138.94	133.52	1,167.92
		60.00	200,221		33,861	2,581	97,560	16,579	2,468.89	2,453.76	3,139.36
	87-89	40.00	15,748	1,829	13,923	2,435	15,938	13,923	12.82	10.65	340.17
		50.00	79,681		23,276	2,437	42,398	15,849	223.73	215.85	1,371.53
		60.00	255,626		35,460	2,437	123,580	15,977	10,587.19	10,568.65	3,669.02
	90-92	40.00	12,960	1,828	13,122	2,365	11,982	13,122	10.33	7.18	269.28
		50.00	67,655		21,660	2,370	29,642	14,764	127.09	120.45	1,443.80
		60.00	197,974		32,534	2,370	76,758	14,968	2,779.65	2,765.56	3,273.09
	93-95	40.00	14,473	1,832	14,097	2,342	15,909	14,097	10.98	8.92	341.17
		50.00	72,968		23,458	2,342	41,615	15,426	170.21	164.22	1,819.45
		60.00	217,392		35,392	2,342	113,949	15,581	2,735.79	2,719.88	3,417.82
	96-98	40.00	14,685	1,838	13,778	2,524	16,540	13,778	10.35	8.23	331.70
		50.00	70,315		22,861	2,524	42,805	16,268	141.77	136.21	1,352.35
		60.00	209,713		34,613	2,524	117,748	16,386	2,178.78	2,164.06	3,413.22
	99-101	40.00	14,488	1,834	14,179	2,487	16,283	14,179	10.91	8.75	356.68
		50.00	74,011		23,903	2,487	44,479	16,161	171.21	165.87	1,738.68
		60.00	228,697		36,094	2,487	128,770	16,302	2,770.44	2,753.50	3,473.52
	102-104	40.00	11,452	1,830	12,547	2,217	10,406	12,547	6.61	4.75	281.53
		50.00	56,210		21,254	2,221	24,941	13,670	70.83	66.74	1,412.88
		60.00	159,019		32,050	2,221	63,440	13,909	1,070.91	1,060.27	3,251.74
	189-191	40.00	16,457	1,821	22,907	2,557	19,338	22,907	17.01	14.52	6,291.70
		50.00	93,170		38,913	2,560	54,718	25,992	1,253.17	1,246.39	14,301.59
		60.00	309,484		59,153	2,560	165,274	26,255	5,646.42	5,623.78	22,560.71
	192-194	40.00	17,699	1,810	24,824	2,768	21,102	24,824	25.95	23.04	6,674.27
		50.00	115,598		41,018	2,770	62,394	23,713	1,459.86	1,450.85	14,202.39
		60.00	395,360		61,627	2,770	195,118	24,103	10,613.52	10,584.38	19,862.15
500-unit hypothetical problems	108-110	40.00	22,643	3,068	18,644	4,085	23,888	18,644	28.17	24.33	2,448.78
		50.00	111,048		31,779	4,093	61,823	30,258	406.96	395.93	8,523.89
		60.00	330,575		48,247	4,093	169,752	30,884	6,920.04	6,890.28	17,748.60
	111-113	40.00	26,156	3,050	33,657	3,830	24,924	33,657	40.84	36.47	10,038.91
		50.00	137,074		57,368	3,838	65,817	30,250	608.00	595.16	20,697.16
		60.00	420,371		87,033	3,838	180,368	30,485	14,530.98	14,491.35	30,740.52
	120-122	40.00	17,220	3,057	27,181	4,642	20,436	27,181	15.02	12.24	7,688.99
		50.00	74,235		45,491	4,652	49,348	31,455	171.11	164.37	15,434.25
		60.00	198,940		69,320	4,652	122,099	31,676	3,647.06	3,628.28	24,365.35
	135-137	40.00	43,061	3,052	27,715	4,578	48,923	27,715	241.91	234.15	7,638.60
		50.00	310,037		47,271	4,588	160,994	32,571	4,936.36	4,905.92	15,683.86
		60.00	1,188,313		71,433	4,588	557,488	32,841	129,895.00	129,781.14	26,663.42
	141-143	40.00	30,129	3,051	35,538	4,763	34,699	35,538	81.19	75.86	10,143.24
		50.00	169,757		58,937	4,772	97,204	33,683	1,254.79	1,237.15	19,267.10
		60.00	546,063		88,894	4,772	283,289	33,949	35,702.50	35,645.51	31,542.25
	144-146	40.00	32,933	3,056	35,098	4,917	38,667	35,098	109.15	103.45	11,123.05
		50.00	194,096		58,726	4,924	115,408	33,397	1,932.03	1,912.50	21,404.01
		60.00	668,388		88,670	4,924	361,239	33,534	40,145.54	40,068.83	31,508.70
	150-152	40.00	49,090	3,046	39,553	5,102	63,527	39,553	2,973.76	350.95	13,224.18
		50.00	369,159		65,220	5,106	221,387	33,259	5,515.69	8,078.36	23,266.02
		60.00	1,501,787		98,045	5,106	811,518	33,415	160,533.47	160,338.75	33,898.84
	153-155	40.00	48,495	3,043	21,923	5,078	62,263	21,923	5,097.38	386.16	4,421.20
		50.00	372,582		37,677	5,080	213,411	32,807	6,107.62	10,629.55	9,334.47
		60.00	1,504,258		57,403	5,080	773,017	33,076	167,768.99	167,581.37	19,401.93
	159-161	40.00	63,985	3,044	15,425	5,018	76,663	15,425	789.36	775.33	442.03
		50.00	554,743		25,917	5,023	279,577	16,595	47,693.36	47,634.63	2,283.87
		60.00	2,417,709		38,767	5,023	1,060,839	16,772	423,982.86	423,672.59	4,092.80
	168-170	40.00	22,199	3,055	15,822	5,005	25,559	15,822	30.94	27.45	447.52
		50.00	107,177		26,623	5,010	67,463	17,996	1,665.73	1,656.44	2,365.92
		60.00	317,002		39,705	5,010	180,288	18,153	5,377.14	5,344.47	4,163.09

Grayed out cells represent formulation times obtained on a different, higher-performance computer

Table 5. Solution characteristics for 0.05% target gap runs: real problems

Test Problems	Number of stands	Maximum harvest opening size	Solution time (s) / Optimality Gap (%)				Reduction of NPV due to ARM
			Cluster	Bucket	Path/Cover/Cell		
					Conventional	Lazy	
Pack Forest, Washington	186	24.28 ha	0.44%	1.83%	0.21%	0.24%	1.35%
		32.37 ha	0.58%	0.95%	0.20%	0.19%	0.98%
		40.47 ha	0.92%	0.86%	0.44%	0.21%	0.96%
		48.56 ha	no solution	1.01%	0.55%	0.23%	0.27%
NBCL5, Canada	5,224	21.00 ha	21.27 s	532.14 s	11.23 s	25.56 s	0.70%
		30.00 ha	86.63 s	0.07%	22.63 s	11.78 s	0.28%
		40.00 ha	12,747.56 s	19,515.86 s	79.78 s	5.02 s	0.08%
El Dorado, California	1,363	48.56 ha	32.23 s	0.08%	20.16 s	96.5 s	0.71%
		60.70 ha	115.92 s	0.14%	75.61 s	36.67 s	0.55%
		72.84 ha	530.95 s	0.56%	3518.24 s	354.53 s	0.43%
Shulkell, Nova Scotia	1,019	40.00 ha	53.44 s	133.28 s	4.30 s	4.36	0.06%
		60.00 ha	7,315.89 s	3,339.63 s	52.56 s	8.06	0.01%
Kittaning4, Pennsylvania	32	40.00 ha	162.23 s	235.19 s	13.48 s	13.52 s	7.78%
		50.00 ha	473.14 s	2,724.56 s	8.92 s	3.99 s	0.74%
		60.00 ha	1,164.09 s	13,322.46 s	4.38 s	12.13 s	0.41%
		80.00 ha	138.88 s	0.27%	13.81 s	11.91 s	0.00%
FivePoints, Pennsylvania	90	40.00 ha	210.25 s	6.97 s	4.03 s	3.09 s	11.89%
		50.00 ha	461.71 s	7,074.70 s	0.56 s	0.72 s	4.52%
		60.00 ha	229.89 s	10,342.17 s	0.78 s	0.83 s	4.51%
		80.00 ha	2,426.52 s	35.297 s	0.33 s	0.66 s	-0.01%
PhyllisLeeper, Pennsylvania	89	40.00 ha	0.16%	0.18%	0.07%	0.05%	0.04%
		50.00 ha	0.16%	0.11%	0.08%	11,678.69 s	0.01%
		60.00 ha	0.15%	0.21%	19,553.28 s	1,117.45 s	0.01%
		80.00 ha	0.13%	0.20%	1,796.89 s	20,081 s	0.00%
BearTown, Pennsylvania	71	40.00 ha	0.18%	0.21%	0.15%	0.10%	0.15%
		50.00 ha	0.24%	0.14%	0.12%	0.07%	0.07%
		60.00 ha	0.14%	0.38%	0.14%	0.10%	0.07%
		80.00 ha	0.24%	0.51%	0.06%	0.09%	0.05%

Note: The negative sign for the percent NPV reduction due to the 80 ha maximum clear-cut size restriction for FivePoints is due to the fact that both the problem with and without ARM constraints was solved to 0.05% optimality. This is the reason why the profit maximizing objective value in the ARM can exceed the objective value of the problem without ARM by 0.01%.

Table 6. The number and percentage of path constraints used during optimization under three different optimality gaps

Test Problems	Number of stands	Maximum harvest opening size	Adjacency constraints in lazy constraint pools: 1%		0.05%		0.01%	
			No.	%	No.	%	No.	%
Pack Forest, Washington	186	24.28 ha	38	0.48%	93	1.18%	93	1.18%
		32.37 ha	23	0.07%	51	0.15%	51	0.15%
		40.47 ha	13	0.01%	37	0.02%	37	0.02%
		48.56 ha	8	0.00%	26	0.00%	26	0.00%
NBCL5, Canada	5,224	21.00 ha	1,009	3.09%	966	3.36%	962	2.94%
		30.00 ha	662	0.75%	669	0.92%	656	0.74%
		40.00 ha	382	0.12%	328	0.13%	366	0.11%
El Dorado, California	1,363	48.56 ha	564	1.04%	1,231	3.36%	601	1.11%
		60.70 ha	467	0.28%	402	0.92%	561	0.34%
		72.84 ha	824	0.16%	709	0.13%	931	0.18%
Shulkell, Nova Scotia	1,019	40.00 ha	42	0.08%	47	0.09%	63	0.12%
		60.00 ha	8	0.00%	8	0.00%	7	0.00%
Kittaning4, Pennsylvania	32	40.00 ha	35	23.81%	29	19.73%	29	19.73%
		50.00 ha	8	4.85%	12	7.27%	10	6.06%
		60.00 ha	3	1.56%	12	6.25%	10	5.21%
		80.00 ha	0	0.00%	6	1.95%	4	1.30%
FivePoints, Pennsylvania	90	40.00 ha	17	3.78%	45	10.00%	54	12.00%
		50.00 ha	19	2.91%	23	3.52%	31	4.75%
		60.00 ha	27	2.87%	16	1.70%	37	3.93%
		80.00 ha	5	0.27%	7	0.38%	128	6.93%
PhyllisLeeper, Pennsylvania	89	40.00 ha	41	7.11%	134	23.22%	134	23.22%
		50.00 ha	60	6.31%	126	13.25%	122	12.83%
		60.00 ha	33	2.93%	91	8.08%	94	8.35%
		80.00 ha	38	1.57%	74	3.06%	98	4.05%
BearTown, Pennsylvania	71	40.00 ha	76	15.61%	101	20.74%	101	20.74%
		50.00 ha	26	3.93%	78	11.80%	78	11.80%
		60.00 ha	39	4.42%	73	8.27%	73	8.27%
		80.00 ha	33	1.80%	39	2.13%	39	2.13%
Hypothetical problems (means)	300, 500	40.00 ha	44.45	0.20%	71.45	0.33%	116.80	0.54%
		50.00 ha	26.07	0.04%	45.07	0.08%	74.28	0.13%
		60.00 ha	14.80	0.01%	25.20	0.02%	45.77	0.03%

Table 7. Solution characteristics: the number of "wins" for each model / method

Target opt. gap	Test Problems	Cluster	Bucket	Path/Cover/Cell		Total
				Conventional	Lazy	
1%	Real	0 (0%)	0 (0%)	10 (35.7%)	18 (64.3%)	28 (100%)
	Hypothetical	0 (0%)	0 (0%)	2 (1.1%)	178 (98.9%)	180 (100%)
0.05%	Real	0 (0%)	0 (0%)	11 (39.3%)	17 (60.7%)	28 (100%)
	Hypothetical	0 (0%)	2 (1.1%)	4 (2.2%)	174 (96.7%)	180 (100%)
0.01%	Real	2 (7.1%)	0 (0%)	7 (25.0%)	19 (67.9%)	28 (100%)
	Hypothetical	20 (11.1%)	40 (22.2%)	21 (11.7%)	99 (55.0%)	180 (100%)

Figure Captions

Figure 1. Best- and worst-case performance analysis - 0.01% target gap runs

Table Titles

Table 1. Initial age-class distribution and yield table for the hypothetical forests

Table 2. Test problem characteristics (some of the information in this table is based on Table 1 in Tóth et al. 2012)

Table 3. Test problem formulation characteristics: cover/path size distribution

Table 4. Test problem formulation characteristics: problem size and formulation time

Table 5. Solution characteristics for 0.05% target gap runs: real problems

Table 6. The number and percentage of path constraints used during optimization under three different optimality gaps

Table 7. Solution characteristics: the number of “wins” for each model/method