A Cutting Plane Method for Solving Harvest Scheduling Models with Area Restrictions

Nóra Könnyű[a], Sándor F. Tóth[a,*]

[a]*School of Environmental and Forest Sciences, University of Washington, Box 352100, Seattle, WA 98195*

Abstract

We describe a cutting plane algorithm for an integer programming problem that arises in forest harvest scheduling. Spatial harvest scheduling models optimize the binary decisions of cutting or not cutting forest management units in different time period subject to logistical, economic and environmental restrictions. One of the most common constraints requires that the contiguous size of harvest openings (i.e., clear-cuts) cannot exceed an area threshold in any given time period or over a set of periods called green-up. These so-called adjacency or green-up constraints make the harvest scheduling problem combinatorial in nature and very hard to solve. Our proposed cutting plane algorithm starts with a model without area restrictions and adds constraints only if a violation occurs during optimization. Since violations are less likely if the threshold area is large, the number of constraints is kept to a minimum. The utility of the approach is illustrated by an application where the landowner needs to assess the cost of forest certification that involves clear-cut size restrictions stricter than what is required by law. We run empirical tests and find that the new method performs best when existing models fail: when the number of units is high or the allowable clear-cut size is large relative to average unit size. Since this scenario is the norm rather than the exception in forestry, we suggest that timber industries would greatly benefit from the method. In conclusion, we describe a series of potential applications beyond forestry.

Keywords: OR in natural resources, integer programming, cutting planes, spatially-explicit harvest scheduling

1. Introduction

We propose a cutting plane algorithm to optimize area-based forest harvest scheduling. Harvest scheduling models, that are typically cast as integer programs, optimize the spatiotemporal

*Corresponding author. Tel.: +1 206 616 2738
Email addresses: nk6@uw.edu (Nóra Könnyű), toths@uw.edu (Sándor F. Tóth)

layout of harvests subject to a variety of logistical, economic and environmental constraints. Area-based models ensure that the contiguous size of harvest openings (i.e., clear-cuts) cannot exceed a maximum threshold in any given time period or over a set of periods called green-up. Area-based harvest scheduling problems are combinatorial problems that are often very hard to solve to optimality. The proposed cutting plane algorithm starts with a model without area restrictions and adds constraints only if a violation occurs during optimization. Before providing a formal definition of the algorithm, we give a brief background and literature review on harvest scheduling models.

The National Forest Management Act of 1976 was the first piece of legislation in the United States that imposed restrictions on the size of clear-cuts. The Act responded to public criticism, which emerged in the 1960s, that large clear-cuts compromised wildlife habitat and other forest ecosystem functions. Many states followed suit and established clear-cut size regulations on both private and state forestlands [3]. Forest certification standards such as those administered by the Forest Stewardship Council (FSC) or the Sustainable Forestry Initiative (SFI) also dictate various limits on harvest opening sizes [16]. The compliance of forest managers who enroll in an FSC or SFI program, is ensured by periodic third-party audits.

The intention behind the policy of restricting harvest opening sizes was to reduce the spatial and temporal concentration of harvest activities across the landscape. A possible side effect of this policy, however, a heterogeneous, patchy forest landscape, has been shown to have both positive and negative ecological consequences [15]. A forest with a spatially heterogeneous age-class distribution is more resilient against the spread of fire, but the increased amount of edge will increase the likelihood of wind-throw and compromise interior old forest habitat. Clear-cut size restrictions may also reduce timber revenues.

Computing the tradeoffs between timber revenues and landscape metrics is useful for policy makers, for the designers of forest certification standards and for forest landowners/managers who are interested in certification. However, such tradeoff analyses may be prohibitively expensive. The larger the opening limit relative to the average size of the harvest units, the harder it is to formulate and solve these models [18]. We give a brief overview of prior work on spatial harvest scheduling and discuss a real-life example that illustrates the computational issues that can arise.

1.1. Area-based Harvest Scheduling in Forestry

Harvest scheduling models seek to maximize timber revenues or other outputs subject to environmental, logistical or budgetary constraints by assigning harvest decisions to forest management units (contiguous groups of trees that share similar characteristics such as species or average height) over a planning horizon. "Environmental" or sustainability constraints might include ending timber volume (inventory) requirements, a balanced flow of timber revenues, or maximum harvest opening size restrictions. Harvest scheduling models that incorporate maximum harvest opening size constraints are called Unit- or Area Restriction Models [a.k.a., URM vs. ARM, 30] depending on whether or not the combined area of every pair of adjacent units exceeds the allowable area threshold. If it does, the problem is a URM that prevents adjacent forest stands from being harvested simultaneously or within a pre-specified timeframe called the green-up or exclusion period. Otherwise, the problem is an instance of the ARM.

The core of the URM, which was first formulated as a mixed integer program (MIP) by [20], and subsequently by [28, 31, 32, 44] is an instance of the Node Packing Problem, a.k.a. the Vertex Packing or the Maximal Weight Stable Set Problem in integer programming. The ARM is a more general model; it allows groups of contiguous management units to be harvested concurrently as long as their combined area is less than the maximum harvest opening size. The ARM typically arises when the maximum harvest opening size is large relative to the size of the units. Since the ARM is a generalization of URM, it can be viewed as a Stable Set problem where the requirement on the independence of the nodes is relaxed subject to a pre-specified threshold. While this threshold is defined as area in forest planning, it does not have to be - giving rise to potential applications beyond forestry. In portfolio optimization as an example, it can be defined as threshold covariance among a cluster of financial instruments. Being able to select a pair of correlated instruments whose covariance is below the tolerable risk of the investor can be advantageous given the difficulties of finding large independent sets in an increasingly globalized stock market [5, 6]. Since the Stable Set Problem has been shown to be NP-Hard [35], the ARM is also NP-Hard. Due to the computational difficulties that are typically associated with solving NP-Hard decision problems, the first methods that have been proposed for the ARM all involved the use of heuristic techniques to find good solutions [e.g., 3, 7]. It was not until the early 2000s, when the first exact models of the ARM appeared in the refereed literature [4, 27, 34]. The first model in [27], called Path Formulation, enforces harvest opening size restrictions by means of constraints only. The formulation of these constraints, which are structurally very similar to the 0-1 cover inequalities in knapsack problems, requires the enumeration of all contiguous clusters of management units whose combined area just exceeds the maximum opening size. Subsequent attempts to improve the Path Formulation include [11] who appended knapsack constraints to [27]'s model to enforce area restrictions, [40] who proposed a strengthening procedure for the path constraints and [39] who showed that the use of path inequalities in lazy constraint pools can lead to dramatic savings in solution times.

The second exact model in [27], the Generalized Managent Unit (GMU) or Cluster Packing Formulation also relies on an enumeration procedure [17, 23, 24, 27, 33]. Unlike the Path Formulation, however, which requires minimally infeasible clusters, it is the set of feasible clusters that are needed in this model. Feasible clusters are contiguous groups of management units whose combined area is less than or equal to the maximum opening size. The GMU model uses extra decision variables to represent feasible clusters that comprise more than one management unit. Pair-wise [27] or maximal clique-based [17] constraints can then be written to prevent the harvest of adjacent or overlapping clusters. [18] showed that the maximal clique-based Cluster-Packing Model provides a tighter approximation of the convex hull of the ARM than the Path Formulation and that it can produce superior computational results.

The third model, Bucket Formulation [9], is very different from the previous two in that it does not rely on apriori enumerations of feasible or infeasible clusters. Unlike the Cluster Packing models or the Path Formulation, this model uses harvest assignment variables instead of harvest variables. Any one management unit can be assigned to initially empty sets of *clear-cuts* or *buckets* [18] in every planning period. While the management units that are assigned to the same clear-cut don't have to be adjacent, constraints are in place to ensure that the total area of each clear-cut is less than or equal to the maximum harvest opening size. Finally, there are additional constraints in the model to prevent the formation of adjacent or overlapping clear-cuts. What makes the Bucket Formulation very attractive is that unlike the other two models, it does not require potentially costly enumerations and that the size of the model is limited by the number of feasible clear-cut assignments. While extra preprocessing is needed to identify "infeasible" assignments, the potential reduction in the number of variables and constraints can be substantial. That said, problem size for the Bucket Formulation can increase exponentially as a function of the number of units depending on the efficiency of the preprocessing algorithm.

1.2. Problem Motivation

The size of the Path and the Maximal Clique-based Cluster Packing formulations is sensitive to the maximum harvest opening size, whereas the size of the Bucket model is sensitive to the number of management units. While problem size is not necessarily a good predictor of problem difficulty [41], it can make the problem formulation process prohibitively time-consuming, especially if cluster enumerations are involved. Moreover, solving large integer programs, such as the ARMs listed above, can also be hard [9, 17]. These issues are never more apparent than in tradeoff analyses where forgone timber revenues or changes in ecosystem metrics, such as forest habitat fragmentation, are to be forecasted as functions of alternative harvest opening size policies. Forest policy makers, landowners or forestry practitioners are all likely to be interested in how much compliance with a specific certification standard or a new regulation would cost. With the three existing ARM approaches, one has to formulate and solve a separate model for each opening size restriction of concern. This can be a time-consuming process if the forest in question has a large number of management units, or if the maximum harvest opening size is large relative to the average size of the units, or if many different harvest opening size policies are considered. Forest planning problems that involve thousands of management units are common. In fact, most authors argue that future research should focus on solving problems that comprise even more units [e.g., 17]. Forest regions across the globe where the allowable clear-cut size is large are not uncommon either. In the Canadian provinces of Ontario and New Brunswick, in central and north Quebec, and in some regions of Alberta, as well as in Victoria, Australia and in Russia, the maximum opening size varies between 100 and 260 ha [25]. In the Pacific Coastal states of Oregon and Washington, the limit is only 48.56 ha but contiguous clear-cuts up to 97.12 ha are allowed with special permission [25, 43]. Very small management units are the norm in these regions, especially in the U.S. private forest sector, where timber harvesting rights are typically allocated to willing buyers via auctions [2, 38]. Small-scale buyers can bid only on small sales that contain lower timber volumes, and as a result, forest managers often design small units [36].

The 1,708 ha Pack Forest, United States is a prime example of the computational difficulties that arise in harvest scheduling when unit sizes are small relative to the allowable clear-cut size. Pack Forest has 186 units with an average unit size of 9.18 ha and with feasible clusters that average 10.27 in cardinality. Washington State Forest Practices limits the opening size to 48.56 ha with a 5-year green-up requirement [43]. The administration of the Forest wants to weigh the pros and cons of acquiring FSC certification which requires 24.28 ha maximum clear-cut size subject to green-up rules that are driven by silvicultural factors such as tree height and canopy closure [16]. To estimate the opportunity costs of FSC compliance, the maximum timber revenues under the State's 48.56 ha clear-cut size rule need to be calculated in the absence of FSC requirements. Formulating the harvest scheduling model that could estimate these revenues using the Maximal Clique-based Cluster Model requires over ten million variables, whereas the Path Approach requires almost a million constraints. The cluster enumeration procedure alone, required by both models, takes several weeks to complete using the best available algorithms. While the Bucket Model takes only minutes to formulate, the optimality gaps achieved with set time-frames are much larger than those of the other two models.

Pack Forest is not the only organization facing this scheduling problem. There are thousands of other forest owners in Washington State whose holdings are large enough to require harvest scheduling models and all of them have to comply with the State's 48.56 ha restriction on clear-cut size. The nature of the problem is likely to be similar or more pronounced in other regions with significant private holdings and with similar policies (e.g., Oregon, U.S. [25]).

1.3. The Proposed ARM Cutting Planes Approach

The cutting plane algorithm proposed in this study can bypass the computational issues listed above. More importantly, we show that it can complement the three exact models as it works best when the others are most likely to fail. The idea of using the path constraints from [27]'s Path Formulation as cuts stems from the expectation that path constraints should rarely be binding during optimization, especially if the maximum harvest opening size is large. This expectation, which we confirm empirically, is based on the fact that while the number of path constraints increases exponentially as a function of maximum opening size, area violations are less likely to occur since the ARM becomes more relaxed. We hypothesize that by creating only those path constraints that prevent the specific opening size violations that arise from solution candidates during optimization would not only lead to smaller IPs that are easier to solve, but it would also cut formulation times.

We note that the proposed cutting plane approach is an instance of what is often called *delayed constraint generation* (DCG). DCG was first reported by [12] in the context of the Traveling Salesman Problem (TSP), which, like the Path Model, requires a potentially enormous set of constraints. It was these, so-called *sub-tour elimination constraints* that [12] used as cuts to be the first to solve the 54-city TSP. Apart from the successful use of cutting planes in the TSP [see 22, for a review], the method was also found to be very efficient in vehicle routing problems [1, 10].

2. Methods

This section starts with a formal description of the Path, the Maximal Clique-based Cluster Packing and the Bucket Formulations that were used to benchmark the computational performance of the cutting plane algorithm. The proposed algorithm is described next, along with the computational experiment that was designed to compare both formulation and solution times. Both of these time components were included in the analysis since they cannot be separated in the cutting plane method where the formulation of Path constraints is imbedded in the optimization process.

2.1. Model Formulations

2.1.1. Terminology and Model Specifications.

Let P denote the set of planning periods and N the set management units associated with a given forest planning problem. Let p index set P and i index set N. Each management unit has the following attributes: area a_i, age in years at the end of the planning horizon if the unit is cut in period p, $t_{i,p}$, volume in the unit at the end of the horizon if it is cut in period p, $v_{i,p}$, expected net revenue in period p, $r_{i,p}$ and the set of units that are adjacent to unit i, D_i. We consider two units to be adjacent if they share a common boundary. We assume that a management unit with forest type

k cannot be harvested until it reached a minimum rotation age of R_k periods. Lastly, we assume that $R_k \geq P$: each unit can only be harvested at most once during the planning horizon.

All of the three benchmark models were set to maximize discounted net timber revenues subject to four sets of constraints: (1) logical constraints that allow each unit to be harvested at most once over the planning horizon; (2) harvest flow constraints that limit the harvest volumes in a given period to be both below and above a certain percentage, U and L, respectively, of the volume in the previous period; (3) average ending age constraints that force the area-weighted average age of the forest at the end of the planning horizon to be at or above a certain target age $\overline{ET}$; and (4) maximum harvest opening size constraints that prevent the harvest of contiguous clusters of management units whose combined area exceeds a threshold A_{max} in a planning period. In this study, we limited the temporal extent of the maximum harvest opening size restrictions (a.k.a., the green-up period) to be equal to one period. Lastly, we assumed that $a_i \leq A_{max}$ for any $i \in N$.

2.1.2. The Path Formulation [27]:

The Path Formulation requires the apriori enumeration of minimal covers (or minimally infeasible clusters, see Fig. 1). After [17], we let Λ^+ denote this set with C representing one particular element in Λ^+. A cover $C \in \Lambda^+$ is minimal if $\sum_{j \in C} a_j > A_{max}$ holds, but $\sum_{j \in C \setminus \{l\}} a_j \leq A_{max}$ for any $l \in C$. In the computational comparisons that follow, we used Algorithm I [18] to generate Λ^+ for the test problems. Algorithm I, which also generates the set of feasible clusters (Λ^-), uses the adjacency table, the areas of the management units and A_{max} as inputs:

$$\Gamma(M, A_{max}) : M \times A_{max} \mapsto \Lambda^- \times \Lambda^+ \tag{1}$$

where $M = [m_{i,j}]_{|N| \times |N|}$ is the adjacency matrix with $m_{i,j} = \begin{cases} a_i & \text{if } i = j \text{, else} \\ 1 & \text{if } i \text{ and } j \text{ are adjacent, and} \\ 0 & \text{otherwise.} \end{cases}$

We refer to Algorithm I as $\Gamma(M, A_{max})$ in this paper and direct the reader to [18] for a full description of the algorithm. In the Path Formulation (2)-(8), the decision variables, $x_{i,p}$, represent the choice whether unit i should be harvested in period p or not. Inequality set (3) captures the logical, sets (4)-(5) capture the harvest flow, (6) the minimum average ending age and (7) the adjacency constraints with u denoting the length of the green-ups in periods and t the planning periods. Constraints (8) define the binary restrictions:

$$\max \sum_{i,p} r_{i,p} x_{i,p} \tag{2}$$

Subject to:

$$\sum_{p} x_{i,p} \leq 1 \qquad \forall i \in N \tag{3}$$

$$\sum_{i} v_{i,p+1} x_{i,p+1} \leq U \sum_{i} v_{i,p} x_{i,p} \qquad \forall p \in P \backslash \{1\} \tag{4}$$

$$\sum_{i} v_{i,p+1} x_{i,p+1} \geq L \sum_{i} v_{i,p} x_{i,p} \qquad \forall p \in P \backslash \{1\} \tag{5}$$

$$\sum_{i,p} (t_{i,p} - \overline{ET}) a_i x_{i,p} \geq 0 \tag{6}$$

$$\sum_{i \in C} \sum_{t=p}^{min(p+u-1,|P|)} x_{i,t} \leq |C| - 1 \qquad \forall C \in \Lambda^{+}, \quad \forall p \in P \tag{7}$$

$$x_{i,p} \in \{0, 1\} \qquad \forall i \in N, \quad \forall p \in P. \tag{8}$$

Let (2)-(6) and (8) be denoted as the *core* of the Path Formulation.

2.1.3. Maximal Clique-based Cluster Packing Formulation [17]:

This model also relies on an apriori enumeration process. Unlike in the Path Formulation, however, this time the set of feasible clusters (Λ^{-}) is needed (Fig. 1). A contiguous set of management units g forms a feasible cluster if their combined area does not exceed A_{max}. Again, we will use [18]'s Algorithm I (1) to generate set Λ^{-}. The variables $x_{g,p}$ in this model represent the decision of whether all the management units in cluster g should be cut in period p or not. We note that these variables are defined for $p = 0$ only if they denote a one-unit cluster. This ensures that the ending age constraint functions as intended. Coefficients $t_{g,p}$ denote the age in years at the end of the planning horizon if cluster g is cut in period p. In the experiments that follow, we used the maximal clique-based version of the Cluster Packing Formulation [17], which requires the enumeration of the maximal sets of mutually adjacent management units (cliques): Π. A clique $K \in \Pi$ is maximal if adding one extra unit to the set would result in a group of units that are no longer mutually adjacent. The Maximal Clique-based Cluster Packing Formulation can be defined as follows:

$$\max \sum_{g,p} [x_{g,p} \sum_{i \in g} r_{i,p}] \tag{9}$$

Subject to:

$$\sum_{g \in G_i, p} x_{g,p} \leq 1 \qquad \forall i \in N \tag{10}$$

$$\sum_{g}[x_{g,p+1}\sum_{i\in g}v_{i,p+1}] \leq U\sum_{g}[x_{g,p}\sum_{i\in g}v_{i,p}] \qquad \forall p \in P\backslash\{1\} \tag{11}$$

$$\sum_{g}[x_{g,p+1}\sum_{i\in g}v_{i,p+1}] \geq L\sum_{g}[x_{g,p}\sum_{i\in g}v_{i,p}] \qquad \forall p \in P\backslash\{1\} \tag{12}$$

$$\sum_{g,p}(t_{g,p} - \overline{ET})\sum_{i\in g}a_i x_{g,p} \geq 0 \tag{13}$$

$$\sum_{g\in G_K}\sum_{t=p}^{min(p+u-1,|P|)} x_{g,t} \leq 1 \qquad \forall K \in \Pi, \quad \forall p \in P. \tag{14}$$

$$x_{g,p} \in \{0,1\} \qquad \forall g \in \Lambda^-, \quad \forall p \in P \tag{15}$$

where G_i is the set of feasible clusters that contain unit i, and G_K is the set of clusters that contain at least one unit in K. Constraints (11)-(12) are harvest flow constraints, and constraint (13) is the average ending age constraint. Script u in (14) refers the length of green ups in periods.

2.1.4. The Bucket Formulation [9]:

Define class B, indexed by b, as a set of *clear-cuts* [or *buckets* after 18]. Since $|B| \leq |N|$, $b \in N$. Clear-cuts can be empty sets or they can comprise one or more management units that are not necessarily connected to each other (Fig. 1). The composition of clear-cuts is left to the model (16)-(25) to determine by means of assignment variables $y^b_{i,p}$, that represent the decision whether management unit i should be assigned to clear-cut b in period p or not. The following model maximizes discounted net timber revenues (16) subject to logical (17), harvest flow (18)-(19), minimum average ending age (20), and binary (22) constraints:

$$\max\sum_{i,b,p} r_{ip}y^b_{ip} \tag{16}$$

Subject to

$$\sum_{b,p} y^b_{i,p} \leq 1 \qquad \forall i \in N \tag{17}$$

$$\sum_{i,b} v_{i,p+1}y^b_{i,p+1} \leq U\sum_{i,b} v_{i,p}y^b_{i,p} \qquad \forall p \in P\backslash\{1\} \tag{18}$$

$$\sum_{i,b} v_{i,p+1}y^b_{i,p+1} \geq L\sum_{i,b} v_{i,p}y^b_{i,p} \qquad \forall p \in P\backslash\{1\} \tag{19}$$

$$\sum_{i,p}(t_{i,p} - \overline{ET})a_i y^b_{i,p} \geq 0 \tag{20}$$

$$\sum_i a_i y^b_{i,p} \leq A_{max} \qquad \forall b \in N \tag{21}$$

$$y^b_{i,p} \in \{0, 1\} \qquad \forall i, b \in N, \quad p \in P. \tag{22}$$

Constraint set (21) puts a limit on the total area of management units that can be assigned to the same clear-cut. Since adjacent clear-cuts might still have a combined area that exceeds A_{max}, constraint set (21) alone cannot prevent all harvest size violations. The Bucket Model uses two additional constraint sets, (23)-(24), and a set of indicator variables, $w^K_{b,p}$, that turn on if at least one unit in maximal clique K is assigned to clear-cut b in period p, to keep the clear-cuts disjoint:

$$y^b_{i,p} \leq w^K_{b,p} \qquad \forall K \in \Pi, \quad i \in K, \quad b \leq i, p \in P \tag{23}$$

$$\sum_b \sum_{t=p}^{min(p+u-1,|P|)} w^K_{b,t} \leq 1 \qquad \forall K \in \Pi, \quad p \in P \tag{24}$$

$$w^K_{i,p} \in \mathbb{R}^+ \qquad \forall K \in \Pi, \quad b \in N, \quad p \in P \tag{25}$$

Constraint set (23) defines the behavior of indicator variables $w^K_{b,p}$ based on the values of the assignment variables. Constraint set (24) says that each management unit in a clique, say clique K, can only be assigned to one clear-cut. Lastly, constraints (25) define the indicator variables as positive real. Constraints (24) allow greenups of length u in periods.

At $O(|N|^2 \times |P|)$ [9], the size of the Bucket Formulation can exceed the size of the Path or the Maximal Clique-based Cluster Packing models if $|N|$ is large [18]. However, the model can be reduced to a fraction of its size by creating assignment variables only for those pairs of units that can be connected by a chain of other units whose combined area plus the area of the two original units doesn't exceed the maximum opening size [9]. As in [9], a dynamic programming recursion was used [13, 37, 42] in this study as well to calculate the area of the *shortest area chains* between each pair of units:

$$s(i, j, l) = \begin{cases} \alpha(i, j) & \text{if } l = 1 \\ min\{s(i, j, l-1),\ s(i, l, l-1) + s(l, j, l-1)\} & \text{otherwise} \end{cases} \tag{26}$$

where $\alpha(i, j)$ is the combined area of units i and j with $a(i, j)$ being ∞ if the units are not adjacent, and $s(i, j, l)$ is the total area of the shortest area chain between units i and j going through intermediate units $1, 2, ...,$ and l. If, for a pair of units i and j, Function (26) returns an $s(i, j, |N|)$ that is greater than A_{max}, there is no need for variables that would assign unit i to clear-cut j, or vice versa. This simplification considerably reduces model size at a minimal preprocessing cost.

2.2. *The Cutting Plane Algorithm for the Path Formulation:*

As described in Section 1.2, the formulation time for the Path Model can be excessive due to the enormous number of cover constraints that might be needed to prevent all possible harvest opening size violations. To minimize formulation times associated with the cover enumeration process , we propose to use the cover inequalities (7) as *cuts* or *cutting planes*. Unlike conventional constraints, cutting planes do not have to be identified pre-optimization. The proposed cutting plane method, whose two variants are formally described in Algorithms (1)-(2), starts as the standard branch-and-bound algorithm [21] with the LP relaxation of the core Path Formulation being the root node (LP_0): (2)-(6) with $x_{i,p} \geq 0 \ \forall i \in N, \ \forall p \in P$. In other words, we start with the LP relaxation of the model that lacks the cover inequalities that would prevent harvest opening size violations. The branch-and-bound algorithm is instructed to stop whenever an LP sub-problem, say LP_k (where k indexes the sub-problems), is found with an objective value that is better than the best found so far (a.k.a., the incumbent solution). In either case, the cutting plane algorithm checks if the new solution contains any harvest opening size violations. We call this detection effort the *ARM Separation Problem*. If a violation is found, the algorithm creates a constraint of form (7) to prevent that specific violation and appends it to the problem. Otherwise, no action is taken. In either case, the branch-and-bound algorithm proceeds until another potentially optimal solution is found. The Separation Problem is invoked again and the process continues until no more violations are found and all of the existing LP sub-problems are solved.

Formally, the ARM Separation Problem is defined as a modified version of Function (1). Let $H_{k,p}$ denote the set of management units $m \in N$ such that $x_{m,p} = 1$ in LP_k. Then, the Separation Problem for LP_k is to evaluate if $\bigcup_p \Lambda^+_{k,p} = \{\emptyset\}$ for $\Gamma(M_{k,p}, A_{max}) : M_{k,p} \times A_{max} \mapsto \Lambda^+_{k,p} \ \forall p \in P$, where $M_{k,p} = [m_{i,j}]$ with $i, j \in H_{k,p}$. If $\Gamma(M_{k,p}, A_{max})$ returns $\{\emptyset\}$, then LP_k is ARM-feasible. Otherwise, one constraint of type (7) needs to be written for each $C \in \Lambda^+_{k,p}$ and appended to LP_k and to the other active nodes. While both the domain and the image of $\Gamma(M_{k,p}, A_{max})$ are different from Goycoolea's $\Gamma(M, A_{max})$, what is important is that, typically, $M_{k,p} \Subset M$, and as a result $M_{k,p}$ is much sparser than M. In other words, the units in $M_{k,p}$ are much less likely to form contiguous clusters than those in M. The computational implication of this, namely that evaluating function $\Gamma(M_{k,p}, A_{max})$ is in most cases trivial compared to $\Gamma(M, A_{max})$, is critical to performance of the proposed algorithm.

As mentioned earlier, we developed two different strategies for imbedding the ARM Separation Problem in the branch-and-bound algorithm. In the first strategy (Algorithm 1), the Separation Problem is invoked whenever a solution, fractional or not, is found to an LP sub-problem with an objective value better than the incumbent's. The branch-and-bound algorithm starts as usual.

There is one active (or *dangling*) node, $LP_0 \in D$, where D is the set of active nodes, and an empty set of incumbent solutions (Z^*). The lower bound on the optimal objective value ($\underline{z}$), is set to $-\infty$. Next we evaluate if D is empty. If it is, we check if we have an incumbent in Z^*. If we do, the incumbent is optimal, otherwise the problem is infeasible. If D is not empty, another sub-problem, LP_k, is selected using the solver's built-in variable selection strategies, and solved as an LP. If the program is infeasible or if it is feasible but the solution has an objective value z_k that is less than the incumbent's, then LP_k is dropped from set D and another sub-problem is selected. Otherwise, we evaluate $\Gamma(M_{k,p}, A_{max})$. If $\Gamma(M_{k,p}, A_{max})$ yields an empty $\bigcup_p \Lambda^+_{k,p}$, i.e., LP_k is ARM feasible, we check the integrality of LP_k. If LP_k is integral, we drop it from set D along with all the other LP sub-problems that have an objective value less than or equal to z_k. We identify LP_k as the incumbent and set $\underline{z}$ to be equal to z_k. If LP_k is fractional, we instruct the solver to branch on a variable and create two LP sub-problems that are appended to D, replacing LP_k. If $\Gamma(M_{k,p}, A_{max})$ yields a non-empty $\bigcup_p \Lambda^+_{k,p}$, we append all sub-problems in D, including LP_k, with constraints (7) written for all $C \in \Lambda^+_{k,p}$ and $p \in P$. In other words, we add the path constraints as *global* cuts. We also tested the option of adding the constraints to only LP_k as *local* cuts, but this strategy proved to be computationally inferior to global cuts. Once the cuts are appended to the dangling nodes, the composition of set D is reassessed and the process continues in an iterative manner until no more violations are found and set D becomes empty, or if the gap between the objective value of the incumbent and the best node falls below a predefined threshold.

In the second strategy (Algorithm 2), the Separation Problem is invoked, i.e., $\Gamma(M_{k,p}, A_{max})$ is evaluated only if the solution to an LP sub-problem with an objective value better than the incumbent's is integral. If LP_k is fractional, we instruct the solver to branch on a variable and create two new LP sub-problems replacing LP_k in D as in the first strategy. If LP_k is integral and $\Gamma(M_{k,p}, A_{max})$ yields an empty $\bigcup_p \Lambda^+_{k,p}$, then LP_k is ARM feasible and dropped from set D along with all the other LP sub-problems that have an objective value less than or equal to z_k. We identify LP_k as the incumbent and set $\underline{z}$ to be equal to z_k. If $\Gamma(M_{k,p}, A_{max})$ yields a non-empty $\bigcup_p \Lambda^+_{k,p}$, the algorithm proceeds as in the first strategy until another LP sub-problem is found with an objective value better than the incumbent's.

There is a tradeoff between the two strategies. The Separation Problem is less frequently invoked in the second than in the first strategy leading to potential time savings. However, the identification of path constraints that cut off large amounts of fractional solution space might be delayed or missed with the second strategy. We test both strategies.

2.3. The computational experiment

Sixty hypothetical and eight real forest planning problems, including Pack Forest, were used to test the performance of the cutting plane algorithms in comparison with the benchmark models. We compared the formulation plus solution times (*total times*) and tested if the computational advantage of the cutting planes method was sensitive to different maximum clear-cut size restrictions. We hypothesized that the method would perform better with increasing clear-cut sizes.

2.3.1. The test forests:

The hypothetical problems were generated using a program called MakeLand [26]. MakeLand randomly created ten 300 and ten 500-unit forests and assigned age-classes to each of the units in such a way so that the overall age-class distribution of each forest would be slightly over-mature resembling a typical Pennsylvania hardwood forest. The age-class allocations were repeated three times for each forest resulting in 60 problems.

The real forests included the 5,224-unit NBCL5 (New Brunswick, Canada), the 1,323-unit Eldorado (California, U.S.) and the 1,019-unit Shulkell (Nova Scotia, Canada). The datasets associated with these forests were obtained from the University of New Brunswick's Forest Management Optimization Site [14]. Four of the five remaining real forests were from Pennsylvania, U.S.: the 32-unit Kittaning4, the 71-unit FivePoints, the 89-unit PhyllisLeeper and the 90-unit BearTown. Lastly, the 186-unit Pack Forest was from Washington, U.S. The hypothetical forests, plus Pack, Eldorado and Shulkell had only one forest type and site class. Kittaning4, FivePoints, PhyllisLeeper and BearTown had five forest types and three site classes, while NBCL5 had six forest types and only one site class. The size of the largest management units in each forest was smaller than the most restrictive harvest opening size limit that was applied to that particular forest. To achieve this, the units larger than 16.1 ha in Shulkell and larger than 20.23 ha in NBCL5 were excluded. Units in NBCL5 without yield information in the FMOS database were also dropped.

2.3.2. Planning parameter settings:

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 Eldorado, 40 and 60 ha on Shulkell and 21, 30 and 40 ha on NBCL5. These restrictions are comparable to those used in earlier studies such as [18]. Other critical planning parameters included the length of the planning horizon, which was set to 60 years for the hypothetical problems, and to 50, 45, 40 or 25 years for the real problems. The length of the planning periods was 10 years for each problem except for Eldorado, Shulkell and Pack where it was only 5 years. We assumed that the forests regenerated in the same forest type and site class within one planning period after harvest. The minimum rotation age was different for each forest but was constant across forest types and site classes with the exception of NBCL5 where it varied based on forest type. The minimum average age of the forests at the end of the planning horizon was set to be 50 years for Pack and 40 years for Eldorado, Shulkell, Kittaning4, FivePoints, PhyllisLeeper, BearTown and for the hypothetical forests. In NBCL5, the ending age was set to be at least one half of the minimum rotation age: 10, 15, 20, 25, 30 and 50 years for the six forest types, respectively. The lower and upper bounds on the harvest volumes in a particular period except the first were varied for each forest between 80 and 95%, and between 110 and 130%, respectively, of that of the previous period.

2.3.3. Implementation:

We formulated and solved the Path, Maximal Clique-based Cluster Packing and Bucket models, and ran the proposed cutting plane algorithms, using the Java programming language and IBM-ILOG CPLEX v. 12.1 [19] Concert Technology (4-thread, 64-bit, released in 2009). A Power Edge 2950 server was used for both model formulations and optimization. The server had four Intel Xeon 5160 central processing units at 3.00Gz frequency and 16GB of random access memory (RAM). The operating system was MS Windows Server 2003 R2, Standard x64 Edition with Service Pack 2 (MS Windows 2003). There were a few larger model instances that we formulated on a more powerful machine, Power Edge R510 that had two Intel Xeon X5670 central processing units at 3.00Gz frequency, with 32GB of RAM and MS Windows Server 2008 R2, Standard x64 Edition. These "larger" instances were the Bucket models for Shulkell at 24.3 ha A_{max}, Eldorado at 72.8 ha and NBCL5 at 21, 30 and 40 ha, and the Path and the Maximal Clique-based Cluster Packing models for Pack at 48.6 ha A_{max}. As it will be seen in the Results section, the fact that for a few problems the formulation times were measured using the faster machine, had no impact on our conclusions because these formulation times were still longer than those obtained with the alternative models where the slower Power Edge 2950 was used.

Algorithm I [18], or, equivalently, the evaluation of $\Gamma(M, A_{max})$, was implemented in Java. Hash tables and linked lists were used to store the intermediate results of enumeration and to check for redundancies in an efficient manner. Algorithm I was not only used to formulate the Path and the Maximal Clique-based Cluster Packing models but it was also used to solve the ARM Separation Problem, $\Gamma(M_{k,p}, A_{max})$, in the proposed cutting plane algorithms. The maximal clique enumeration and the Floyd-Warshall Algorithm (Eq. 26), that were needed for the Maximal Clique-based Cluster Packing and the Bucket models, were also implemented in Java. Formulation times did not include the times to calculate the revenue and volume coefficients.

As for mixed integer optimization, the solver CPLEX was instructed to terminate after 6 hours of run time or after a 0.05% optimality gap was reached, whichever happened first. Except for the 1GB working memory limit, all other CPLEX parameters were left at their default settings. The ARM Separation Problem was imbedded in both versions of the cutting plane algorithm using CPLEX's *lazy constraint callback*, and *cut callback* routines. Callbacks are user-defined sub-routines that are implemented by CPLEX during optimization whenever certain conditions hold. CPLEX invokes lazy constraint callbacks whenever a new solution, fractional or not, is found with an objective function value that is better than the incumbent's. In cut callbacks, the user defines the conditions under which the sub-routine is invoked. We used lazy constraint callbacks for the first (Algorithm 1) and cut callbacks for the second cutting planes strategy (Algorithm 2). CPLEX was instructed to invoke the cut callbacks whenever an integral solution was found with an objective function value better than the incumbent's. All cuts were added as globally valid. We experimented with adding the cuts locally, but found that many cuts had to be re-created repeatedly during optimization leading to inefficiencies.

3. Results

Tables 1 and 2 contain details about the computational performance of the proposed cutting plane method. These details include formulation and solution times, optimality gaps and the number and percent of path constraints that were used as cuts by the cutting plane method during optimization. The column "Formulation time" lists only the Core (Eq. 2-6 and 8) formulation times. Computing times associated with the detection algorithm were included in the "Solution times" column. Tables 3 and 4 compare the formulation and solution times that were achieved by the cutting plane method with those of the original Path, the Maximal Clique-based Cluster Packing and the Bucket models. The cluster enumeration times, which are part of both the Path and Maximal Clique-based Cluster Packing formulation times, were listed separately in the second column to show how significant this effort can be computationally. All results listed in the above tables with respect to the cutting plane approach are from the first strategy (Algorithm 1) where the ARM Separation Problem was invoked every time a potentially optimal solution, fractional or not, was found by the branch-and-bound algorithm. While on average the second strategy (Algorithm 2) led to 13% better performance, it ran out of the 6-hour time limit in six, while it ran out of memory in three out of the sixty hypothetical problems. We never experienced any time or memory limit issues with the first strategy.

Tables 1 and 3 display results about the eight real, while Tables 2 and 4 display results about the sixty hypothetical forest planning problems. Since there were too many hypothetical problems to display individually, only aggregate results are listed in Tables 2 and 4. The aggregate results include medians and first and third quartiles, which are cutoff values for the lowest 25 and 75% of the test population in terms of performance metrics. The optimality gaps were not aggregated.

One key observation that can be made based on Tables 1 and 2 is that the number of path constraints that were used as cuts by the cutting plane method was, in the vast majority of cases, only a tiny fraction of the entire set of path constraints that are needed for the original Path Model. The only exceptions were Kittaning4, FivePoints, PhyllisLeeper and the BearTown instances with 40 and 50ha max opening sizes, where, not surprisingly, the cutting plane method was not able to outperform the original Path Model (Table 3). Another important observation is that the proportion of path constraints always decreased as the maximum clear-cut size increased. This foreboded our primary result, shown in Tables 3 and 4, that the proposed cutting plane algorithm outperformed the other methods, sometimes by dramatic margins, as the maximum clear-cut size increased. It is clear that formulation times for the Path and the Maximal Clique-based Cluster Packing models were very sensitive to increasing maximum clear-cut sizes, while those of the Bucket model were very sensitive to the increasing number of units. Considering Shulkell at 24.3 ha max opening size, or Eldorado at 72.8ha, while the Path or the Maximal Clique-based Cluster Packing models required 2-3 days to complete the formulation and the optimization process, the proposed cutting plane method found a solution within the predefined 0.05% optimality gap in 3 minutes for the former and in 71 minutes for the latter instance.

The advantage of the cutting plane method was even more dramatic for Pack Forest at the 40.5 or 48.6 ha max opening sizes. Of the 924,133 path constraints that are needed for the full Path Model at the 48.6 ha opening size, which is the legal limit in Washington, only 30 (0.00325%) had to be used by the cutting planes method to find a feasible solution within 0.16% of optimality (Table 1). At 40.5 ha, only 54 of the 170,232 constraints (0.0317%) were necessary. The huge difference in the number of constraints that had to be used led to massive savings in formulation time. While formulating the core Path Model, which is all what is needed for the cutting planes method, took only 1.66 seconds regardless of opening size, the full Path and Maximal Clique-based Cluster Packing Models each took more than 60 days for the State's 48.6 ha limit and they took almost 2 days for the 40.5 ha limit to formulate (Table 3). Formulation times were very reasonable at 2-3 minutes for the Bucket Model since Pack Forest had only 186 units. However, in terms of optimality gaps (none of the models solved to the target 0.05% gap within 6 hours), the Bucket Model performed far worse than the proposed cutting planes method and the other benchmarks (Table 3). At 0.34 and 0.16%, respectively, the cutting planes method achieved the best gaps for the 48.6 and the 40.5 ha opening size instances at Pack Forest. Of note is that the Maximal Clique-based Cluster Model did not produce any feasible solutions at 48.6 ha.

As for the relative performance of the Bucket Model vs. the cutting plane method, the latter did always better in real problems. The advantage of cutting planes was particularly dramatic in cases where the number of units in the forest was large as in NBCL5 (Table 3).

To further contrast how the four methods performed with respect to the hypothetical problems, Fig. 2 maps out trend lines for solution and formulation plus solution times on natural logarithmic scales as functions of increasing clear-cut size. The lines were derived from median formulation and solution times. The top graphs show that solution times for the Path and the Maximal Clique-based Cluster Packing models increased more than polynomially with the maximum clear-cut size. On the other hand, solution times for the Bucket method increased only slightly, and they peaked in the middle of the clear-cut size range for the cutting plane method. If only the solution times are considered, the cutting plane method did not perform better than the other models. Moreover, the medians obtained with the Bucket model were always below those of the cutting plane method. This is not surprising since solution times include formulation times for the cutting plane method. The two cannot be separated due to the integrated nature of the algorithm. If formulation plus solution times are compared, the result is very different (bottom graphs). Total times for the Path and the Maximal Clique-based Cluster Packing models increased at a rate faster than solution times as a function of increasing maximum harvest opening size. At the extreme, in the 500-unit, 60ha category, the median total time was one magnitude higher than the median solution time for these two models. In 5 out of 6 categories (bottom graphs), median total times for the Bucket model were greater than that of the cutting plane method. Except for the 300-unit, 40 ha category, where the median for the Bucket model was only marginally smaller at 553.46s than that of the cutting plane at 555.44s, all the other problems formulated and solved faster with the new method.

4. Discussion and conclusions

Overall, our tests showed that the proposed cutting plane method can be very efficient in formulating and solving spatially-explicit harvest scheduling problems with area restrictions if the maximum clear-cut size is large relative the average size of the units or if the number of units is high, or both. The greater the number of units, or the larger the maximum harvest opening size, the cutting plane method is more likely to outperform the other models. While total times for the Path, the Maximal Clique-based Cluster Packing and the Bucket models exhibited a robust increasing trend as a function of maximum harvest opening size or as a function of the number of units, the total time for the cutting plane method either decreased or if it increased, it increased at a sub-linear rate. With the exception of Eldorado and the four smallest problems (Kittaning4, FivePoints, PhyllisLeeper and BearTown), where the relative number of path constraints used as cuts was high, total times with the cutting plane method were always the lowest at the largest clear-cut size. This trend was expected since a relaxation of harvest opening size should lead to fewer violations in potential solutions. Fewer violations require fewer path constraints to be generated by the cutting plane method, which in turn could result in shorter computing times.

The management planning implications of these time savings is best illustrated by Pack Forest, an actual forestry organization. The biggest part of the opportunity costs of acquiring FSC certification at Pack Forest comes from foregone timber revenues. These revenues can only be estimated by finding a harvest schedule that maximizes profit subject to regulations that are present in the absence of FSC certification. Since state regulations include a 48.6 ha limit on maximum clear-cut size, whatever model is used to find a profit maximizing harvest schedule needs to take this restriction into account. Of the three models that were available from the forest planning literature, only two, the Path and the Bucket Models were able to find feasible harvest schedules within 6 hours of solution time (Table 3). The failure of the Maximal Clique-based Cluster Packing Model to find feasible solutions was not completely unexpected as [18] have already predicted that ARM instances where the feasible clusters average at or above 10 units in cardinality "seem largely intractable" (p161). The feasible clusters in the 48.6 ha instance of Pack Forest average 10.27. Of the two models that did find feasible solutions within 6 hours, the Path Model required more than 60 days to formulate. The cutting plane method, which requires only 1.66 seconds of formulation time, can be used to run hundreds of sensitivity analyses during this timeframe to investigate the impact of such important but uncertain factors as wood prices or timber yields. While the Bucket Model formulated the 48.6 ha instance in only 176 seconds, it found a solution only within 1.01% of optimality. Although the 1.01% does not seem much, it amounts to about one year's worth of salaries for one of the Pack Forest workers. Again, the computational advantage of the cutting plane method over the Bucket Model is even more pronounced in larger problems such as NBCL5, Shulkell and El Dorado.

An interesting observation in NBCL5 and the hypothetical problems, is that there is a peak in total times for the cutting plane approach in the middle of the maximum clear-cut size range (Table 1 and 2). We speculate this might be due to tradeoffs between the difficulty of solving the ARM Separation Problem and the cost of adding cuts to active nodes in the branch-and-bound process. As A_{max} increases relative to the average size of the units, the ARM Separation Problem becomes slightly harder to solve because a larger number of units is needed to construct minimally infeasible clusters. As the opening size further increases, however, this added computational difficulty is eventually offset by the lack of units that can possibly form large enough clusters.

It might be possible to further improve the efficiency of the proposed cutting plane approach.

We tested only two strategies for imbedding the ARM Separation Problem in the branch-and-bound algorithm. The two differed only in terms of when the Separation Problem was invoked. While in the first strategy, it was invoked whenever a solution, fractional or not, was found with an objective value that exceeded that of the incumbent's, in the second strategy, it was invoked only when the solution candidate was all-integral. Many other possible strategies exist not only in terms of when the Separation Problem is invoked but also in terms of how exactly the Separation Problem is defined. We defined the set of input variables, $H_{k,p}$, for function $\Gamma(M_{k,p}, A_{max})$, which served as a tool to solve the Separation Problem, based on whether their optimal values in LP_k were equal to one. An equally valid strategy would be to define $H_{k,p}$ based on whether the input variables took strictly positive values in LP_k. If they did, they would be included in $H_{k,p}$, otherwise they would not. While this strategy would make the Separation Problem harder to solve because of the larger input size, it might allow the identification of stronger cuts early in the optimization process.

As a last note, it is important to emphasize that the value of new ARM methodologies is not restricted to timber management. Apart from potential applications in the radio transmission problem [17], the ARM can be used to design prescribed burns or other types of "treatment" units to minimize the risk of catastrophic wildfires and pest infestations in agricultural, grass and forest lands. The contiguous size of the burn or treatment units is typically restricted not only because of operational, safety and economic reasons, but also because the ultimate goal is often to create a landscape structure that is more resilient to future fires and pests [8]. This can be best achieved with a fire or treatment mosaic of burned (treated) and unburned (untreated) areas [8]. Spatial optimization of land-use patterns in managed landscapes [29] is yet another area where the ARM can come handy to promote the spatial diversity of landscape elements such as farms of various crops, grasslands and forests.

Being a generalization of the Stable Set Problem (SSP), the ARM can also be used in other graph theoretical applications of the SSP where some dependence (connectivity) among the nodes is allowed. One example is portfolio optimization where financial instruments on the stock market can be represented as nodes and correlations among instruments with respect to price movements or liquidity can be represented as edges [5, 6]. Finding large independent portfolios in a *market graph* is problematic due to the globalization of the stock market [5]. Thus, the ability to select a pair of instruments that are correlated could be beneficial as long as the covariance (risk) of the pair does not exceed the tolerance of the investor. This is where ARM methodologies, the proposed cutting plane algorithm in particular, can be useful since the market graph is large and risk tolerance varies among investors.

5. Acknowledgment

We thank Dr. Daniel Junglas at IBM Deutschland Research and Development GmbH for his help in software development. Thanks also to the USDA National Institute of Food and Agriculture (NIFA) for financial support under grant numbers WNZ-1327,1328 and 1398.

References

[1] Ascheuer, N., M. Fischetti, M. Grötschel. 2001. Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. *Math. Prog.* **90** 475-506.

[2] Athey, S., J. Levin, E. Seira. 2011. Comparing open and Sealed Bid Auctions: Evidence from Timber Auctions. *Q. J. Econ.* **126(1)** 207-257.

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

[4] Barrett, T. M., J.K. Gilles. 2000. Even-aged Restrictions with Sub-graph Adjacency. *Ann. Oper. Res.* **95** 159-175.

[5] Boginski, V., S. Butenko, P.M. Pardalos. Mining market data: a network approach. *Comput. Oper. Res.* **33** 3171-3184.

[6] Boginski, V., S. Butenko, P.M. Pardalos. 2003. On structural properties of the market graph. In A. Nagurney, editor. *Innovation in Financial and Economic Networks*, Edward Elgar Publishers, London pp 29-45.

[7] Boston, K., P. Bettinger. 2006. An economic and landscape evaluation of the green-up rules for California, Oregon, and Washington (USA). *Forest Policy and Economics.* **8(3)** 251-266.

[8] Bunting, S.C., B.M. Kilgore, C.L. Bushey. 1987. Guidelines for Prescribed Burning - Sagebrush-Grass Rangelands in the Northern Great Basin. *USDA Forest Service General Technical Report INT-231. Intermountain Research Station. Ogden, UT.* 38 pp.

[9] Constantino M., I. Martins, J.G. Borges. 2008. A new mixed-integer programming model for harvest scheduling subject to maximum area restrictions. *Oper. Res.* **56** 542-551.

[10] Cordeau, J.-F. 2006. A Branch-and-Cut Algorithm for the Dial-a-Ride Problem. *Oper. Res.* **54** 573-586.

[11] Crowe, K., J. Nelson, M. Boyland. 2003. Solving the area-restricted harvest-schedu ling model using the branch and bound algorithm. *Can. Jour. For. Res.* **33** 1804-1814.

[12] Dantzig, G.B., D.R. Fulkerson, S.M. Johnson. 1954. Solution of a large-scale traveling-salesman problem. *Oper. Res.* **2** 393-410.

[13] Floyd, R. W. 1962. Algorithm 97: Shortest path. *Commun. ACM.* **5 (6)** 345.

[14] FMOS - Forest Management Optimization Site. 2011. http://ifmlab.for.unb.ca/fmos/datasets/. Retreived: 12/26/2011.

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

[16] Forest Stewardship Council - U.S. Forest Management Standard (v1.0). 2010. `http://www.fscus.org/images/documents/standards/FSC-US%20Forest%20Management%20Standard%20v1.0.pdf`. Retrieved: 12/26/2011.

[17] Goycoolea, M., A.T. Murray. F. Barahona, A. Weintraub. R. Epstein. 2005. Harvest Scheduling Subject to Maximum Area Restrictions: Exploring Exact Approaches. *Oper. Res.* **53** 490-500.

[18] Goycoolea M., A. Murray, J.P. Vielma, A. Weintraub. 2009. Evaluating approaches for solving the area restriction model in harvest scheduling. *For. Sci.* **55 (2)** 149-165.

[19] IBM ILOG CPLEX. 2009. CPLEX 12.1 Reference Manual.

[20] Kirby, M.W.,P. Wong, P.A. Hager, M.E. Huddleston. 1980. A guide to the integrated resources planning model. *USDA Forest Service, Berkeley, CA.* 211 pp.

[21] Land, A.H., A.G. Doig. 1960. An automatic method of solving discrete programming problems. *Econometrica* **28**(3) 497520.

[22] Laporte, G. 2010. A concise guide to the Traveling Salesman Problem. *J. of the Oper. Res. Society.* **61** 35-40.

[23] Martins, I., M. Constantino, J. G. Borges. 1999. Forest management models with spatial structure constraints. Working paper 2/1999. Centro de Investigação Operacional, Faculdade de Ciências, Lisbon, Portugal.

[24] Martins I., M. Constantino, J.G. Borges. 2005. A column generation approach for solving a non-temporal forest harvest model with spatial structure constraints. *Eur. J. of Oper. Res.* **161** 478-498.

[25] McDermott, C., B.W. Cashore, P. Kanowski. 2010. Global Environmental Forest Policies: An International Comparison. London: Earthscan, 2010.

[26] McDill, M. E., J. Braze. 2000. Comparing Adjacency Constraint Formulations for Randomly Generated Forest Planning Problems with Four Age-Class Distributions. *For. Sci.* **46 (3)** 423-436.

[27] McDill, M.E., S.A. Rebain, J. Braze. 2002. Harvest Scheduling with Area-Based Adjacency Constraints. *For. Sci.* **48** 631-642.

[28] Meneghin, B., M. Kirby, J.G. Jones. 1980. "An algorithm for writing adjacency constraints efficiently in linear programming models", in: The 1988 Symposium on Systems Analysis in Forest Resources, *USDA Forest Service Rocky Mountain Forest and Range Experiment Station General Technical Report* RM-161, 46-53

[29] Meyer, B.C., J.M. Lescot, R. Laplana. 2009. Comparison of Two Spatial Optimization Techniques: A Framework to Solve Multiobjective Land Use Distribution Problems. *Environ. Manage.* **43** 264-281.

[30] Murray, A. T. 1999. Spatial restrictions in harvest scheduling. *For. Sci.* **45 (1)** 45-52.

[31] Murray, A.T., R. Church. 1995. Measuring the efficacy of adjacency constraint structure in forest planning models. *Can. Jour. For. Res.* **25** 1416-1424.

[32] Murray, A.T., R. Church. 1996. Analyzing Cliques for Imposing Adjacency Restrictions in Forest Models. *For. Sci.* **42(2)** 166-175.

[33] Murray, A., M. Goycoolea, A. Weintraub. 2004. Incorporating average and maximum area restrictions in harvest scheduling models. *Can. Jour. For. Res.* **34** 456-464.

[34] Murray, A., A. Weintraub. 2002. Scale and Unit Specification Influences in Harvest Scheduling with Maximum Area Restrictions. *For. Sci.* **48(4)** 779-789.

[35] Nemhauser, G., L. Trotter. 1974. Properties of vertex packing and independence system polyhedra. *Math.l Program.* **6** 48-61.

[36] Petroski, J. 2006. Treatment Unit Delineations for Spatial Planning on Pennsylvania State Forest Land. A Thesis in Forest Resources and Operations Research. The Pennsylvania State University College of Agricultural Sciences, 2006.

[37] Roy, B. 1959. Transitivité et connexité. *C. R. Acad. Sci. Paris* 249: 216218.

[38] Stone, J.S., D.B. Rideout. 1997. Does Sealed Bidding Promote Competition in the Auctioning of Public Timber? *J. For. Econ.* **3** 133-142.

[39] Tóth, S.F., M.E. McDill, N. Könnyű, and S. George. In Press. Testing the Use of Lazy Constraints in Solving Area-Based Adjacency Formulations in Harvest Scheduling Models. *For. Sci.*

[40] 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 and Natural Resources Sciences*.**4(1)** 16-38.

[41] Van Roy, T.J., L.A. Wolsey. 1987. Solving mixed integer programming problems using automatic reformulation. *Oper. Res.* **35(1)** 45-57.

[42] Warshall, S. 1962. A theorem on Boolean matrices. *Journal of the ACM* **9(1)** 1112.

[43] Washington State Department of Natural Resources. 2011. Chapter 222-30-025 WAC - Timber Harvesting. p30-26. URL: `http://www.dnr.wa.gov/Publications/fp_rules_ch222-30wac.pdf`. Retrieved: 12/26/2011.

[44] Weintraub, A., F. Barahona, R. Epstein. 1994. A column generation algorithm for solving general forest planning problems with adjacency constraints. *For. Sci.* **40(1)** 142-161.

Algorithm 1 The ARM Branch-And-Cut Algorithm with $\Gamma(M_{k,p}, A_{max})$ invoked at each solution candidate. LP_k represents linear programming sub-problem or node k, where $k = 0$ represents the root node. Set D denotes the set of active, or dangling nodes, and Z^* denotes the set of incumbent solutions. Objective function value $\underline{z}$ is the current best primal bound and z_k represents the objective value corresponding to node k.

Initialize: $\underline{z} \leftarrow -\infty, \quad Z^* \leftarrow \{\emptyset\}, \quad D \leftarrow \{LP_0\}$
while $D \neq \{\emptyset\}$ **do**
  currentNode $\leftarrow$ select element from D {strategy depends on the solver}
  currentSolution $\leftarrow$ solveLP(currentNode)
  $z_k \leftarrow$ objectiveValue(currentNode)
  if currentSolution is feasible **and** $z_k > \underline{z}$ **then**
    $\Lambda_k^+ \leftarrow \bigcup_p \Gamma(M_{k,p}, A_{max})$ {evaluate the ARM Separation Problem $\forall p \in P$}
    if $\Lambda_k^+ \neq \{\emptyset\}$ **then**
      for all $l \in D$ **and** $C \in \Lambda_k^+$ **do**
        append LP_l with constraints $\sum_{m \in C} x_{m,t} \leq |C| - 1$
      end for
    else if currentSolution is integral **then**
      $Z^* \leftarrow$ currentSolution
      $\underline{z} \leftarrow z_k$
      for all $j \in D$ **do**
        $z_j \leftarrow$ objectiveValue(LP_j)
        if $z_j \leq z$ **then**
          $D \leftarrow D \setminus \{LP_j\}$
        end if
      end for
    else
      $D \leftarrow D \cup \{LP_{|D|+1}\} \cup \{LP_{|D|+2}\}$ {branch and create two nodes from currentNode}
    end if
  else
    $D \leftarrow D \setminus$ {currentNode}
  end if
end while
if $Z^* = \{\emptyset\}$ **then**
  problem is infeasible
else
  Z^* is optimal
end if

Algorithm 2 The ARM Branch-And-Cut Algorithm with $\Gamma({}_{k,p}, A_{max})$ invoked at only integral solutions. LP_k represents linear programming sub-problem or node k, where $k = 0$ represents the root node. Set D denotes the set of active, or dangling nodes, and Z^* denotes the set of incumbent solutions. Objective function value $\underline{z}$ is the current best primal bound and z_k represents the objective value corresponding to node k.

Initialize: $\underline{z} \leftarrow -\infty$, $Z^* \leftarrow \{\emptyset\}$, $D \leftarrow \{LP_0\}$
while $D \neq \{\emptyset\}$ **do**
  currentNode $\leftarrow$ select element from D {strategy depends on the solver}
  currentSolution $\leftarrow$ solveLP(currentNode)
  $z_k \leftarrow$ objectiveValue(currentNode)
  if currentSolution is feasible **and** $z_k > \underline{z}$ **then**
    if currentSolution is integral **then**
      $\Lambda_k^+ \leftarrow \bigcup_p \Gamma(M_{k,p}, A_{max})$ {evaluate the ARM Separation Problem $\forall p \in P$}
      if $\Lambda_k^+ \neq \{\emptyset\}$ **then**
        for all $l \in D$ **and** $C \in \Lambda_k^+$ **do**
          append LP_l with constraints $\sum_{m \in C} x_{m,t} \leq |C| - 1$
        end for
      else
        $Z^* \leftarrow$ currentSolution
        $\underline{z} \leftarrow z_k$
        for all $j \in D$ **do**
          $z_j \leftarrow$ objectiveValue(LP_j)
          if $z_j \leq z$ **then**
            $D \leftarrow D \setminus \{LP_j\}$
          end if
        end for
      end if
    else
      $D \leftarrow D \cup \{LP_{|D|+1}\} \cup \{LP_{|D|+2}\}$ {branch and create two nodes from currentNode}
    end if
  else
    $D \leftarrow D \setminus$ {currentNode}
  end if
end while
if $Z^* = \{\emptyset\}$ **then**
  problem is infeasible
else
  Z^* is optimal
end if

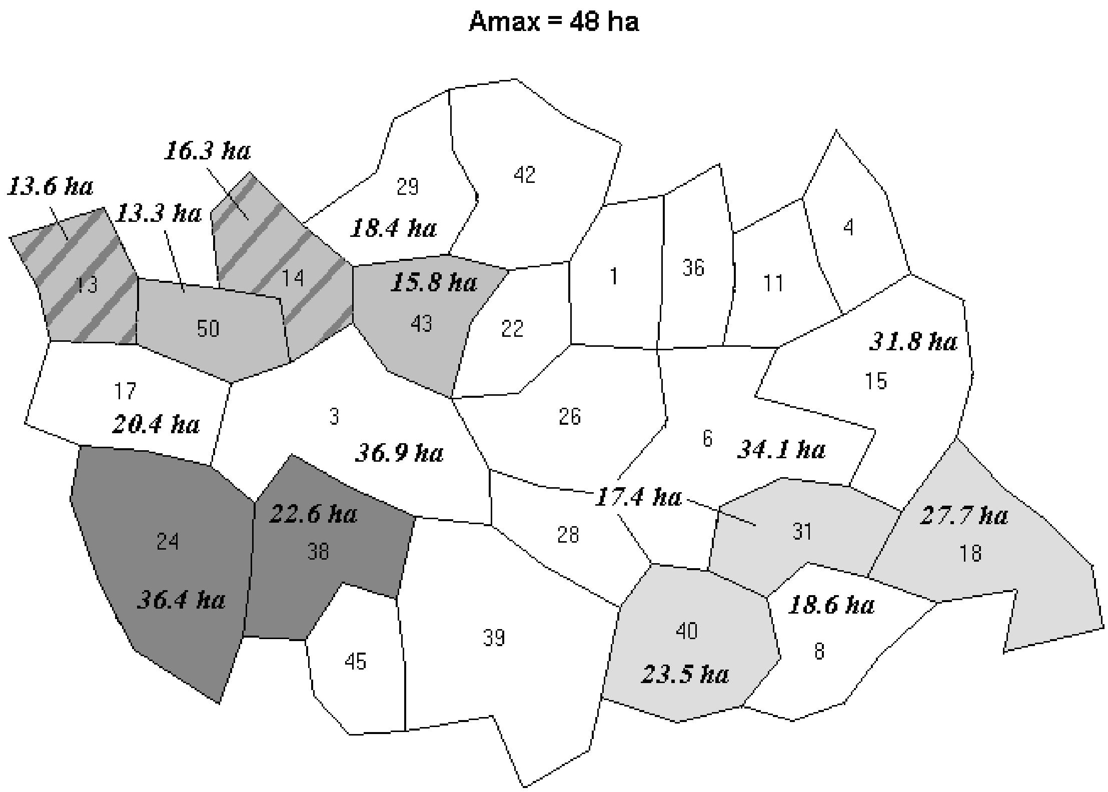

Figure 1: Spatially explicit harvest scheduling models optimize the binary harvesting decisions for each forest stand (depicted as polygons on a map) over a set of planning periods. The figure also shows examples of feasible and minimally infeasible clusters and clear-cuts (or buckets) whose complete sets are required for alternative formulations the ARM. Sets {13,14,43,50}, {24,38} and {18,31,40} are examples of minimally infeasible clusters, used in the Path Formulation [27], whereas {13},{14},{18},{24},{31},{38},{40},{43},{50},{13,50},{14,43},{14,50},{13,14,50},{14,43,50},{18,31} or {31,40} are feasible clusters used in the Maximal Clique-based Cluster Packing Model [17]. Finally, potential clear-cuts used in [9] include all the feasible clusters plus spatially disjoint assignments such as {13,14} or {43,50}.

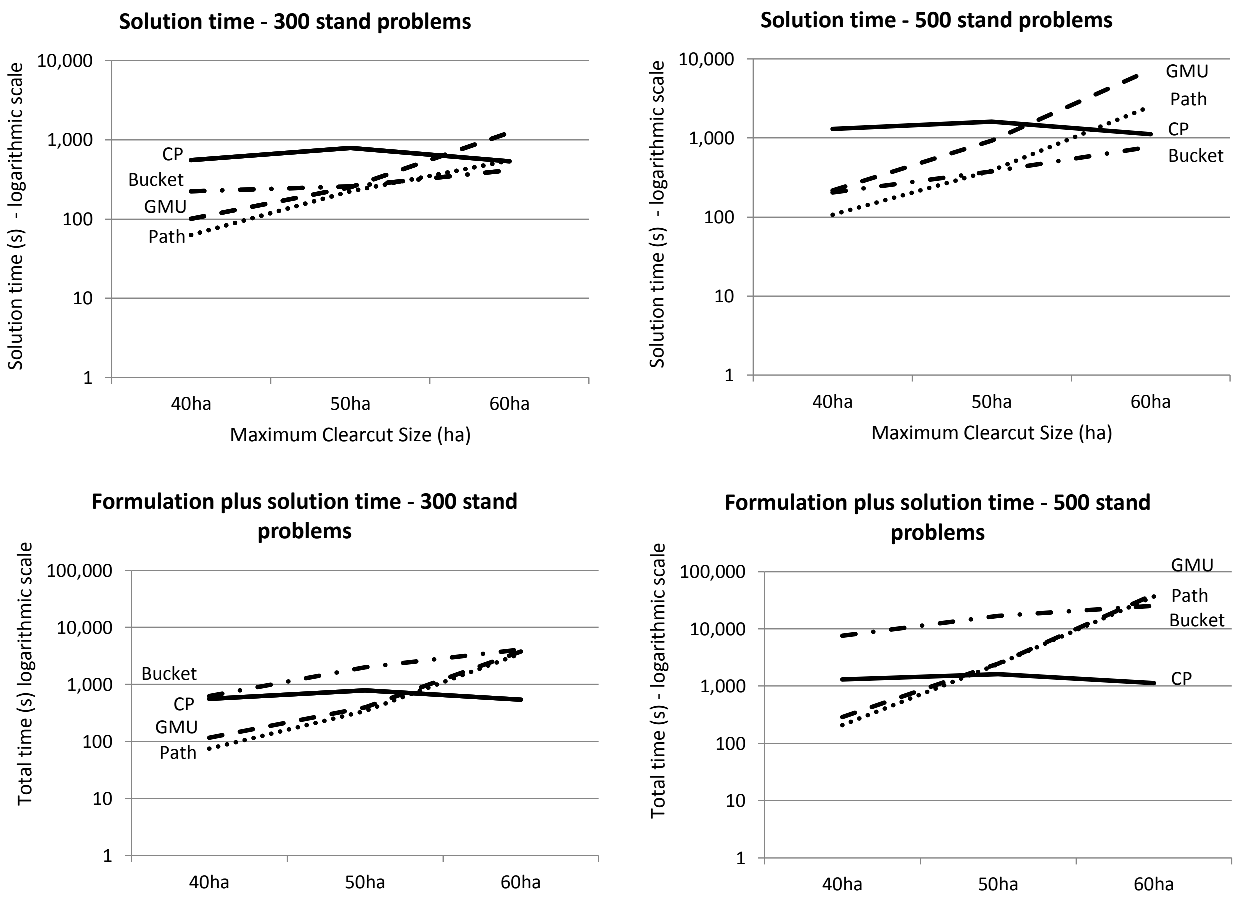

Figure 2: Comparing solution time and total (formulation plus solution) time of the four ARM approaches based on the hypothetical problems. Data points represent the trend in the median values of Table 4.

Table 1: Computational performance of the ARM Cutting Planes - real forests

Problem ID/ Max clear-cut size	Formulation time (s)	Solution time (s)	Optimality gap (%)	Number of cuts used	Percent of path constraints used
Pack, |N|=186					
24.3ha		-	0.29	145	1.84
32.4ha	1.66	-	0.32	79	0.23
40.5ha		-	0.16	54	0.03
48.6ha		-	0.34	30	0.00
Shulkell, |N|=1,019					
16.1ha	2.03	183.24	0.03	56	0.11
24.3ha		179.68	0.04	28	0.01
Eldorado, |N|=1,363					
48.6ha		4,628.30	0.03	1,471	2.72
60.7ha	2.44	3,265.15	0.05	1,228	0.88
72.8ha		4,298.62	0.05	1,866	0.36
NBCL5, |N|=5,224					
21ha		443.13	0.05	1,343	4.67
30ha	4.17	1,081.20	0.04	1,000	1.38
40ha		159.36	0.02	965	0.38
Kittaning4, |N|=32					
40ha		28.37	0.05	64	43.54
50ha	1.36	13.91	0.05	31	18.79
60ha		22.48	0.05	28	14.58
80ha		58.19	0.05	15	4.89
FivePoints, |N|=71					
40ha		36.38	0.05	127	28.22
50ha	1.23	7.42	0.05	93	14.24
60ha		19.94	0.05	88	9.35
80ha		11.88	0.04	52	2.82
PhyllisLeeper, |N|=89					
40ha		-	0.09	483	83.71
50ha	1.17	-	0.09	631	66.35
60ha		-	0.06	653	57.99
80ha		-	0.07	594	24.54
BearTown, |N|=90					
40ha		-	0.11	401	82.34
50ha	1.45	-	0.07	452	68.38
60ha		-	0.18	582	65.91
80ha		-	0.10	724	39.56

Table 2: Computational performance of the ARM Cutting Planes - hypothetical forests

Problem ID	Formulation time (s)	Solution time (s)	Number of cuts used	Percent of path constraints used
THIRD QUARTILE (cutoff for the lowest 75% of the data)				
300-stand problems				
40ha		1,307.01	399.25	3.30
50ha	1.11	2,030.89	359.50	0.84
60ha		1,683.81	244.25	0.22
500-stand problems				
40ha		2,489.13	484.25	1.80
50ha	1.20	2,998.58	390.25	0.43
60ha		2,288.71	270.75	0.13
MEDIAN (center - 50% - of the distribution)				
300-stand problems				
40ha		554.33	347.00	2.41
50ha	1.11	789.04	251.50	0.67
60ha		537.79	167.50	0.16
500-stand problems				
40ha		1,302.44	410.00	1.15
50ha	1.17	1,613.98	303.00	0.30
60ha		1,121.93	213.00	0.08
FIRST QUARTILE (cutoff for the lowest 25% of the data)				
300-stand problems				
40ha		386.92	289.00	1.90
50ha	1.09	508.74	202.25	0.50
60ha		304.68	134.50	0.11
500-stand problems				
40ha		792.64	346.00	0.64
50ha	1.17	940.18	229.75	0.17
60ha		626.90	127.00	0.05

Table 3: Formulation and solution times - real forests

Problem ID	Cluster enumeration	**Path** formulation	**Path** solution	**Cluster Packing** formulation	**Cluster Packing** solution	**Bucket** formulation	**Bucket** solution	**CP** total
		(time in seconds / % optimality gap at 6 hrs)						
Pack, |N|=186								
24.3ha	31.59	36.53	(0.21%)	35.00	(0.44%)	104.78	(1.83%)	(0.29%)
32.4ha	2,731	2,846	(0.20%)	2,751	(0.58%)	135.68	(0.95%)	(0.32%)
40.5ha	162,593	164,079	(0.44%)	162,726	(0.92%)	121.32	(0.86%)	(0.16%)
48.6ha	5,302,395[c]	5,302,397[c]	(0.55%)	5,303,282[c]	-	176.00	(1.01%)	(0.34%)
Shulkell, |N|=1,019								
16.1ha	546.85	548.87	42.837	586.52	246.45	68,265	795.09	185.27
24.3ha	164,059	164,062	515.09	164,202	16,143	24,555[c]	4,079	181.70
Eldorado, |N|=1,363								
48.6ha	1,497	1,500	20.16	1,516	32.23	56,253	(0.08%)	4,631
60.7ha	20,158	20,160	75.61	20,144	115.92	64,384	(0.14%)	3,267
72.8ha	234,216	234,219	3,518	233,879	530.95	39,149[c]	(0.56%)	4,301
NBCL5, |N|=5,224								
21ha	143.70	182.14	11.23	167.81	21.27	210,717[c]	532.14	447.30
30ha	2,626	2,931	22.63	2,825	86.63	789,070[c]	(0.07%)	1,085
40ha	73,558	83,743	79.78	74,510	12,748	949,889[c]	19,516	163.53
Kittaning4, |N|=32								
40ha	0.05	1.42	13.48	1.80	162.23	0.44	235	29.73
50ha	0.02	1.38	8.92	1.66	473.14	0.22	2,725	15.27
60ha	0.02	1.38	4.38	1.41	1,164.09	0.23	13,322	23.84
80ha	0.02	1.38	13.81	1.16	138.88	0.30	(0.27%)	59.55
FivePoints, |N|=71								
40ha	0.09	1.34	4.03	1.78	210.25	2.84	6.97	37.61
50ha	0.08	1.34	0.56	1.25	461.71	3.47	7,075	8.66
60ha	0.08	1.33	0.78	1.29	229.89	4.47	10,342	21.17
80ha	0.34	1.59	0.33	1.62	2,426.62	6.64	35	13.11
PhyllisLeeper, |N|=89								
40ha	0.05	1.23	(0.07%)	1.41	(0.16%)	2.49	(0.18%)	(0.09%)
50ha	0.05	1.22	(0.08%)	1.19	(0.16%)	3.67	(0.11%)	(0.09%)
60ha	0.06	1.23	19,553.34	1.51	(0.15%)	4.83	(0.21%)	(0.06%)
80ha	0.23	1.44	1,796.89	1.51	(0.13%)	9.79	(0.20%)	(0.07%)
BearTown, |N|=90								
40ha	0.03	1.48	(0.11%)	1.16	(0.18%)	3.08	(0.21%)	(0.15%)
50ha	0.03	1.48	(0.07%)	1.55	(0.24%)	3.39	(0.14%)	(0.12%)
60ha	0.02	1.48	(0.18%)	1.32	(0.14%)	4.11	(0.38%)	(0.14%)
80ha	0.08	1.56	(0.10%)	1.95	(0.24%)	6.19	(0.51%)	(0.06%)

(grey box): best total time/ optimality gap

-: no feasible solution within 6 hrs

[c]: formulated on Power Edge R510

Table 4: Formulation and solution times - hypothetical forests

Problem	Cluster	**Path**		**GMU**		**Bucket**		**CP**
ID	enumeration	formulation	solution	formulation	solution	formulation	solution	total
				(time in seconds)				
THIRD QUARTILE (cutoff for the lowest 75% of the data)								
300-stand problems								
40ha	9.40	10.50	98.02	11.57	146.32	364.14	397.64	1,308.15
50ha	213.77	214.86	276.10	221.90	403.19	1,877.06	1,079.39	2,032.00
60ha	5,634.02	5,635.13	963.78	5,656.08	2,579.94	3,715.71	776.25	1,684.96
500-stand problems								
40ha	348.25	349.43	223.72	358.21	302.66	10,773.57	241.37	2,490.31
50ha	8,094.98	8,096.14	833.70	8,131.70	2,099.76	20,484.90	680.96	2,999.75
60ha)	160,339.08	160,340.25	5,467.03	160,536.04	20,188.99	30,855.36	1,889.09	2,289.58
MEDIAN (center - 50% - of the distribution)								
300-stand problems								
40ha	7.40	8.51	62.92	9.51	100.91	330.11	223.35	555.44
50ha	146.41	147.52	224.99	151.54	251.88	1,726.40	257.46	790.16
60ha	2,656.64	2,657.73	550.03	2,671.16	1,236.04	3,472.05	411.23	538.91
500-stand problems								
40ha	87.48	88.66	107.32	93.00	215.83	7,343.02	203.69	1,303.64
50ha	1,799.79	1,800.96	387.94	1,814.19	930.21	15,978.36	376.56	1,615.20
60ha	34,389.50	34,390.69	2,524.52	34,456.30	7,300.03	25,071.03	776.45	1,123.10
FIRST QUARTILE (cutoff for the lowest 25% of the data)								
300-stand problems								
40ha	6.38	7.46	35.82	8.69	59.74	293.99	130.18	388.02
50ha	121.67	122.77	122.85	127.44	191.84	1,412.86	159.75	509.85
60ha	2,064.02	2,065.12	268.48	2,079.97	1,010.41	3,222.42	339.18	305.77
500-stand problems								
40ha	26.39	27.58	56.03	29.90	126.38	6,191.18	117.33	793.81
50ha	596.25	597.45	172.18	609.30	504.89	13,419.41	256.26	941.38
60ha	6,756.09	6,757.28	791.98	6,786.41	2,428.95	20,344.91	357.69	626.90

: best total time