Harvest Scheduling with Area-Based Adjacency Constraints

Marc E. McDill, Stephanie A. Rebain, and Janis Braze

ABSTRACT. Adjacency constraints in harvest scheduling models prevent the harvest of adjacent management units within a given time period. Two mixed integer linear programming (MILP) harvest scheduling formulations are presented that include adjacency constraints, yet allow the simultaneous harvest of groups of contiguous management units whose combined areas are less than some predefined limit. These models are termed Area Restriction Models, or ARMs, following Murray (1999). The first approach, the Path Algorithm, generates a set of constraints that prevent concurrent harvesting of groups of contiguous stands only when the combined area of a group exceeds the harvest area restriction. The second approach defines the set of Generalized Management Units (GMUs) that consist of groups of contiguous management units whose combined areas do not exceed the maximum harvest area limit. This formulation of the model can recognize direct cost savings—such as sale administration costs or harvest costs—or higher stumpage prices that may be realized by jointly managing stands. Example problems are formulated and solved using both ARM approaches and compared with models that restrict concurrent harvests on all adjacent units, regardless of area [termed Unit Restriction Models, or URMs, again following Murray (1999)]. The ARM formulations usually result in larger models and take longer to solve, but allow for higher objective function values than otherwise similar URM formulations. While the proposed ARM approaches should be applicable to more general problems, the examples are constructed so that the largest number of contiguous stands that can be harvested jointly is three. Strategies for reducing the size of the ARM formulations are discussed and tested. For. Sci. 48(4):631–642.

Key Words: Forest management, area-based forest planning, spatial optimization, integer programming, area restriction models, unit restriction models.

Adjacency constraints in harvest scheduling models prevent adjacent management units from being scheduled for harvest within a given time period. Many articles have been written about formulating and solving forest management planning problems with adjacency constraints (e.g., Boston and Bettinger 1999, Hoganson and Borges 1998, Borges et al. 1999, Lockwood and Moore 1992, McDill and Braze 2000, Meneghin et al. 1988, Murray 1999, Murray and Church 1995a, 1996a, 1996b, Nelson and Brodie 1990, Snyder and ReVelle 1996a, 1996b, 1997, Torres-Rojo and Brodie 1990, Weintraub et al. 1994, Yoshimoto and Brodie 1994). Adjacency constraints are of practical importance largely because of legal and voluntary restrictions on the maximum size of harvest openings. For example, the National Forest Management Act (U.S. Congress 1976) [Sec 6(g)(3)(F)(iv)] and Washington and Oregon's state forest practices acts place legal limits on clearcut sizes; and the Sustainable Forestry Initiative™ of the American Forest and Paper Association (AF&PA 2000) places voluntary limits on clearcut sizes.

When adjacency constraints are included in a model in order to comply with a maximum harvest area policy, it

Marc E. McDill is Assistant Professor of Forest Management, School of Forest Resources, The Pennsylvania State University, 214B Ferguson Building, University Park, PA 16802-4301–Phone: (814) 865-1602; Fax: (814) 865-3725; E-mail: mem14@psu.edu. Stephanie Rebain is Biometrician, USDA Forest Service, Ft. Collins, CO—E-mail: sarebain@fs.fed.us. Janis Braze is Forestry Systems Analyst, Geographic Dynamics Corp., Edmonton, Alberta—E-mail: mailbox@gdc-online.com.

Acknowledgments: The authors wish to thank the associate editor and three anonymous reviewers for helpful comments on an earlier version of the manuscript. Thanks also to the Penn State School of Forest Resources and the Pennsylvania Bureau of Forestry for providing financial support for this research.

Manuscript received February 1, 2000, accepted February 8, 2001.

should be possible to harvest adjacent stands together as long as their combined areas are less than the maximum harvest area. Concurrent harvests should be prohibited only on combinations of contiguous stands whose aggregate areas exceed the maximum harvest area. Constraining the model unnecessarily by prohibiting the concurrent harvest of any pair of adjacent stands will generally lead to suboptimal solutions. In spite of this, nearly all of the forest management optimization formulations presented in the literature that include adjacency constraints have prohibited harvests on all adjacent stands. Sometimes problems are preconditioned so that there are no adjacent stands whose combined areas would be less than the maximum allowable harvest area. For example, Snyder and ReVelle (1996b, p. 1,082) state "It is assumed that each planning unit is large enough such that harvesting in neighboring areas would violate maximum desired harvest opening size." Of course, this assumption often will not hold in practice. Other formulations simply disallow concurrent harvesting of any adjacent stands, without reference to a particular maximum harvest area (e.g., Borges et al. 1999, McDill and Braze 2000). In these cases, however, it often happens that individual stands in the problem—that must be harvested all at once—are larger than the combined areas of groups of adjacent stands that the adjacency constraints prevent from being harvested concurrently. An ideal approach would allow neighboring stands to be harvested together as long as their collective area is less than the maximum harvest area. Whether or not neighboring stands should be harvested together should not be predetermined, however. As long as the combined area of a group of contiguous stands does not exceed the maximum harvest area, the decision whether or not to harvest the group in the same period should be determined by the model solution.

Lockwood and Moore (1992) present a model that allows concurrent harvesting of contiguous groups of stands based on their combined areas. Their model penalizes harvest blocks that are too small as well as too large. Their penalty approach encourages the merging of small stands to create larger, more efficient harvest blocks. Furthermore, the approach allows violations of the maximum harvest area restriction when the costs of complying exceed the penalty for being too large. Lockwood and Moore's model is large and nonlinear, however, and would be very difficult to solve with an exact solution method. Lockwood and Moore used simulated annealing to find near-optimal solutions to their model.

Murray (1999) discusses the problem of imposing area-dependent adjacency constraints. He terms models "where harvesting any two adjacent units would violate the maximum area limitation" (p. 47) Unit Restriction Models (URMs). He proposes the label "Area Restriction Models" (ARMs) for models that impose maximum harvest area restrictions, but allow adjacent units to be harvested together as long as their combined area does not exceed the maximum harvest area. This article uses Murray's terminology, except that any model that precludes concurrent harvesting of all adjacent management units—regardless of their area—is considered a URM. Murray (1999) concludes that:

> To date, no attempts to solve the ARM by exact solution approaches have been published. Because of the non-linear characteristics of the ARM, it is unlikely that exact methods for identifying optimal solutions will ever be developed. This is due to the inherent difficulty of defining contiguous treated areas in advance. (p. 49)

This article presents two ways to formulate harvest scheduling problems that preclude harvests that would exceed a maximum allowable harvest area while allowing the concurrent harvest of contiguous groups of stands whose combined areas are smaller than this predefined limit—i.e., ARMs. Both of the proposed approaches produce mixed-integer linear programming (MILP) formulations. They do not require nonlinear models and are therefore not fundamentally more difficult to solve than other MILP harvest scheduling problems with adjacency constraints. The first method, which we call the Path Algorithm, produces a set of adjacency constraints that imposes only the adjacency restrictions that are needed to preclude harvest openings that would exceed a maximum harvest area limit. The second MILP ARM formulation adds variables that represent combinations of contiguous management units whose aggregate areas do not exceed the maximum allowable harvest area. We call these combinations of management units Generalized Management Units (GMUs). Both approaches will generally produce larger problem formulations than otherwise equivalent URMs. The Path Algorithm tends to require more constraints than an equivalent URM with Type 1 (Meneghin et al. 1988) adjacency constraints. The GMU formulation requires more variables and more constraints. The GMU formulation has the advantage, however, of being able to recognize direct cost savings (e.g., timber sale administration or harvesting costs) and/or higher stumpage prices that may result from combining groups of management units into a single, larger timber sale.

The next two sections describe the logic behind the two approaches to formulating MILP ARMs. Following that, a general model and results from various formulations of two example problems are presented. The concluding section discusses the significance of the proposed model formulations.

The Path Algorithm

An intuitive way to allow qualifying adjacent management units (i.e., adjacent[1] management units whose combined area is less than the maximum harvest area) to be harvested concurrently is to begin with a complete set of Pairwise adjacency constraints (i.e., a set of adjacency constraints that would preclude the concurrent harvest of any adjacent management units) and simply delete the constraints corresponding to pairs of adjacent management units whose combined areas do not exceed the maximum harvest area restriction. Of course, the problem is not that simple, as illustrated by the following example.

Consider a forest with three stands—A, B, and C—as depicted in Figure 1. The following sets of Pairwise constraints would prevent the concurrent harvest of any adjacent stands in the forest:

$$X_{At} + X_{Bt} \le 1$$
$$X_{Bt} + X_{Ct} \le 1 \tag{1}$$

where X_{Ut} is a binary variable where a value of 1 indicates that management unit U should be harvested in period t.

Now, if the combined area of units A and B is less than the maximum harvest area and if the combined area of units B and C is greater than the maximum harvest area, then simply deleting the first of these Pairwise adjacency constraint sets would result in a correct ARM. However, if the combined area of units B and C was also less than the maximum harvest area and if the second set of Pairwise constraints was also deleted, it would then be possible to harvest all three units together. Therefore, if the combined area of units A, B, and C exceed the maximum harvest area, a new constraint of the following form is needed:

$$X_{At} + X_{Bt} + X_{Ct} \le 2 \tag{2}$$

For reasons that will be clear shortly, we refer to constraints of this type as path constraints. The left-hand-side coefficients of these path constraints are always 1, and the right-hand-side coefficient is always one less than the number of variables included in the constraint. Note that, except in the case of path constraints involving only two stands—which are equivalent to Pairwise constraints—these constraints are not Type 1 constraints (Meneghin et al. 1988). First, in the above example, units A and C are not adjacent to each other, and, second, the right-hand-side coefficient is 2, not 1. This second difference is particularly important because it means that these constraints are not clique constraints [i.e., constraints where all of the nonzero coefficients on both sides of the inequality are one (Murray and Church 1996b)]. Clique constraints tend to increase the efficiency of the branch and bound algorithm that is often used to solve MILP problems.

The proposed Path Algorithm is as follows[2]:

1. Start with any pair of two adjacent polygons (stands). If the combined area of the two polygons exceeds the maximum harvest area, then write a Pairwise adjacency constraint for those two polygons. This pair of polygons forms the initial cluster of processed polygons (i.e., the set of polygons for which path constraints have been identified).
2. Select any polygon that is adjacent to the current cluster of processed polygons and add it to the cluster.
3. Define a network, based on the current cluster, with a node corresponding to each polygon and an arc connecting each pair of nodes corresponding to a pair of adjacent polygons. Identify each possible path originating at the node corresponding to the new polygon in the cluster.[3] Terminate a path when the accumulated area of the polygons along the path just exceeds the maximum harvest area or when there are no more polygons in the current cluster that are adjacent to the path but not already part of the path. Note that paths can follow multiple branches, but paths cannot return to a node already on the path. When a path whose area exceeds the maximum harvest area terminates, determine whether the path is redundant. A newly identified path is redundant if the set of stands in a previously identified path is a subset of the stands in the new path. If the new path is not redundant, add a set of path constraints (one for each period, for example, if the exclusion period equals one planning period) corresponding to the set of polygons defined by the path.
4. Stop when the cluster contains all of the management units in the forest. Otherwise, return to step 2.

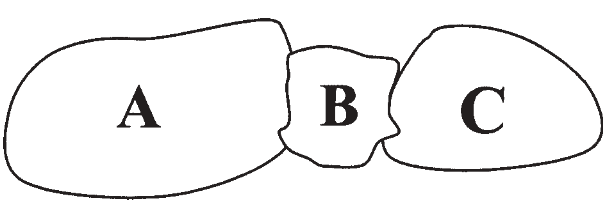

Figure 1. Example forest with three management units.

Implementing this procedure is relatively straightforward as long as the maximum number of management units in a path does not exceed 3 or 4. We have written a program implementing the algorithm for path numbers up to 4. We expect that the algorithm would work with path numbers of 5 or more, but implementation would be considerably more complex. The maximum path size can be limited by not defining stands whose areas are less than or equal to, say, one fourth of the maximum harvest area. In this case, the maximum path length would be four. This is, of course, an overly restrictive requirement. Even if a forest includes stands whose areas are less than one fourth of the maximum harvest area, if these stands are not clustered together there will not be any groups of four contiguous stands with a combined area less than the maximum harvest area. Finally, even if clusters of four or more contiguous stands with a combined area less than the maximum harvest area do exist, one can simply ignore the possibility of harvesting more than three contiguous stands concurrently. This would still represent an improvement on the URM approach, which does not allow concurrent harvests of any adjacent stands.

The Path Algorithm can be illustrated by expanding the example in Figure 1. The forest in Figure 2 contains the three original stands in Figure 1, plus three additional stands, labeled D, E, and F. Figure 2 also shows the areas of the stands. Assume

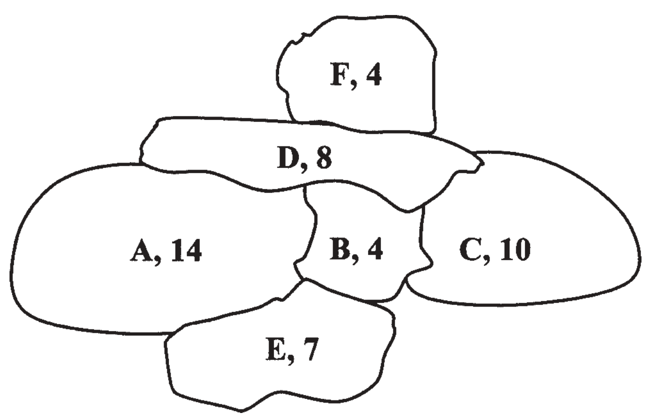

Figure 2. Example forest with six management units (numbers indicate management unit areas).

that the maximum harvest area is 20. Since we have already discussed stands A, B, and C, we continue the example with these stands as our current cluster. Constraint (2) above imposes all of the adjacency restrictions that are needed for this cluster.

Stand D is added next to the cluster. The first path generated from stand D is DA. This path is not redundant, since the only path that has been created so far is ABC, and it is not a subset of DA. Thus, a pairwise path constraint is added for these stands:

$$X_{At} + X_{Dt} \leq 1 \tag{3}$$

The next path that is generated is DBA. Since the stands on path DA are a subset of the stands on this path, it is redundant.[4] The next path is DBC. Since neither ABC nor AD are subsets of DBC, this path is not redundant. The following path constraint is therefore added:

$$X_{Bt} + X_{Ct} + X_{Dt} \leq 2 \tag{4}$$

The next path is a forked path consisting of two subpaths, DB and DC.[5] We denote this type of path with a hyphen between the subpaths, DB-DC in this case. This path is redundant, since it passes through the same set of stands as DBC. The final path that can be generated starting from stand D is DCB. This path also contains the same set of stands as DBC, so this path is redundant. This completes the set of ARM constraints for the current cluster.

Stand E is added next to the cluster. The first path generated from stand E is EA. This path is not redundant, so a pairwise path constraint is added for these management units:

$$X_{At} + X_{Et} \leq 1 \tag{5}$$

The next path that is generated is EBA. The management units on this path are a superset of the management units on path EA, so this path is redundant. The next path is EBC. This path is not redundant, so the following path constraint is added:

$$X_{Bt} + X_{Ct} + X_{Et} \leq 2 \tag{6}$$

The path EBDA is a superset of the path DA, and the path EBDC is a superset of the paths BCD and EBC, so they are redundant and no path constraints are written for those paths. This completes the set of ARM constraints for the current cluster.

Management unit F is added next to the cluster. The first path generated is FDA. This path is redundant because it is a superset of the path DA. The next path that is generated is FDBA. This path is also a superset of the path DA. The next path, FDBC, is redundant, as it is a superset of path BCD. The next path is FDBE. This path is not redundant, so the following path constraint is added:

$$X_{Bt} + X_{Dt} + X_{Et} + X_{Ft} \leq 3 \tag{7}$$

The next path is another forked path, FDB-DC. This path is redundant, as it, too, is a superset of path BCD. However, the next path, FDC, is not redundant, so a final constraint is written for this path:

$$X_{Ct} + X_{Dt} + X_{Ft} \leq 2 \tag{8}$$

Because all of the stands in the forest have now been added to the cluster, this completes the set of path constraints for the forest shown in Figure 2.

This simple example illustrates how the Path Algorithm can be used to construct a set of linear ARM constraints for the forest depicted in Figure 2. For a one-period model, seven constraints would be required. Clearly, enumerating all possible paths from each management unit through the current cluster can be complex. Identifying redundancies can also be nontrivial. Nevertheless, the algorithm can be used to formulate an MILP ARM. One disadvantage of the approach, however, is that it cannot be used to model situations where significant direct cost savings may be realized by harvesting adjacent management units jointly. The approach described in the next section allows this.

The Generalized Management Units Approach

If one knew in advance that two adjacent management units should be harvested together, the two units could simply be combined before formulating the problem. However, whether or not a set of adjacent management units should be combined can also be viewed as a decision that can be made by the model. These decisions can be modeled by adding variables corresponding to allowable combinations of management units (i.e., combinations of contiguous management units whose combined areas do not exceed the maximum allowable harvest area).

In the example forest depicted in Figure 2, there are eight allowable combinations of management units—AB, BC, BD, BDE, BDF, BE, CD, and DF—in addition to the original set of six management units. These combinations can be thought of as alternative, or generalized, management units (GMUs) into which the forest can also be divided, and variables can be defined for these GMUs just as for the original set of individual management units.[6] For example, $X_{BDF,t}$ would be a binary variable where a value of 1 indicates that management units B, D, and F should be harvested concurrently in period t.

Under the approach described here, the logical constraints dictating that a unit can only be assigned to one prescription—such as those that would be found in a multiperiod, Model I URM (Johnson and Scheurman 1977)—would need to be expanded to require that a unit must be managed as one and only one of the GMUs to which it belongs. For example, in a multiperiod, Model I URM the logical constraint for management unit B in Figure 2 would be:

$$\sum_{t=0}^{T} X_{Bt} \leq 1 \tag{9}$$

With the addition of variables corresponding to GMUs, the logical constraint for management unit B would become:

$$\sum_{u \in G_B} \sum_{t=0}^{T} X_{Ut} \leq 1 \tag{10}$$

where G_B is the set of GMUs containing management unit B, including the original unit itself. In the example in Figure 2, this set would be {B, AB, BC, BD, BDE, BDF, and BE}.

Adjacency constraints for GMUs can be written using any standard adjacency constraint formulation, such as Pairwise, Type 1, New Ordinary Adjacency Matrix (NOAM), etc. (Murray and Church 1995, 1996b, McDill and Braze 2000). For example, the GMU AB in Figure 2 is adjacent to management units D, C, E, CD and DF. Note that it is not necessary to treat AB as adjacent to BE, since the logical constraint for stand B [Equation (10)] already makes it impossible to assign prescriptions to both of these GMUs. With the GMUs, there are 38 pairs of adjacent management units in Figure 2. These are listed in Figure 3. A Pairwise adjacency constraint would be required for each of these pairs for each period of the model. The number of adjacency constraints can be reduced to 28 times the number of periods if Type 1 ND constraints are used.[7] The groups of management units corresponding to Type 1 ND constraints for the example forest in Figure 2 are listed in Figure 4.

The GMU approach will generally produce an ARM formulation of a given problem that is considerably larger than a URM formulation of the same problem. In the example in Figure 2, the number of management units (and, hence, variables) more than doubled, from 6 to 14, and the number of Pairwise adjacency constraints needed per period increased from 8 to 38. The ARM could become so large that the resultant model would take prohibitively long to solve. This is a significant drawback of the approach. Nevertheless, several factors moderate these concerns. First, the logical constraints (10) in the GMU ARM are clique constraints. That is, these constraints include long lists of variables from which only one can take a nonzero value. This structure tends to reduce the difficulty of solving these models. Furthermore, with the GMU approach, Type 1 adjacency constraints can be used. McDill and Braze (2000) have shown that this adjacency constraint type is generally more efficient than the most promising alternatives, Pairwise and NOAM constraints. Finally, the optimal solution to the ARM will always be at least as good as the optimal solution to the URM. Thus a suboptimal solution to the ARM—such as one obtained with a heuristic procedure or by terminating the branch-and-bound algorithm prematurely might take less time to obtain and yet have a better objective function value than the optimal solution to the URM. McDill and Braze (2001) have shown that it often takes substantially less time to find slightly suboptimal solutions than to find optimal solutions.

While it is apparently inefficient, relative to the Path formulation of the ARM, the GMU approach is important simply because it provides an alternative way to formulate ARM problems. This formulation may prove to be useful for applications that have not been considered previously—for example, in modeling forest patch sizes or the area of interior forest habitat—or for formulating the ARM for heuristic solution approaches. An important advantage of the GMU approach is that the direct cost savings from jointly managing groups of management units can be modeled easily because the joint management of these units is represented by specific variables. The objective function coefficients corresponding to management alternatives for a GMU do not have to equal the sum of the objective function coefficients for the equivalent management alternatives on the individual management units included in the GMU. Thus, these coefficients can include direct cost savings such as reduced timber sale administration or logging costs that can be realized by joint management of the units. In addition, the GMU approach could potentially be useful for recognizing the cost savings that can be realized through the joint management of any group of stands, whether or not they are adjacent.

A, B	A, D	A, E	A, BC	A, BD
A, BDE	A, BDF	A, BE	A, CD	A, DF
B, C	B, D	B, E	B, CD	B, DF
C, D	C, AB	C, BD	C, BDE	C, BDF
C, BE	C, DF	D, F	D, AB	D, BC
D, BE	E, AB	E, BC	E, BD	E, BDF
F, BD	F, BDE	F, CD	AB, CD	AB, DF
BC, DF	BE, CD	BE, DF		

Figure 3. Adjacency list for 38 pairwise constraints for the example forest in Figure 2.

Model Formulation

The general structure of our ARM MILP based on the Path Algorithm is as follows:

$$Max \quad Z = \sum_{m=1}^{M} \sum_{t=1}^{T} c_{mt} \cdot A_m \cdot X_{mt} \qquad (11)$$

Subject to

$$\sum_{t=0}^{T} X_{mt} \le 1 \qquad \text{for } m = 1,2,...,M \qquad (12)$$

$$\sum_{m=1}^{M} v_{mt} \cdot A_m \cdot X_{mt} - H_t = 0 \qquad \text{for } t = 1,2,...,T \qquad (13)$$

$$B_{lt} H_t - H_{t+1} \le 0 \qquad \text{for } t = 1,2,...,T-1 \qquad (14)$$

$$-b_{h,t} H_t + H_{t+1} \le 0 \qquad \text{for } t = 1,2,...,T-1 \qquad (15)$$

$$\sum_{U \in P_i} X_{Ut} \le n_{P_i} - 1 \qquad \forall \ P_i \text{ and } t = 1,2,...,T \qquad (16)$$

$$\sum_{m=1}^{M} \sum_{t=1}^{T} (Age_{mt}^{T} - \overline{Age}^{T}) A_m \cdot X_{mt} \ge 0 \qquad (17)$$

$$X_{mt} \in \{0,1\} \qquad \text{for } m = 1,2,...,M \text{ and } t = 0,1,2,...T \qquad (18)$$

A, B, D	A, E, BC	A, BD	A, BDE	A, BDF
A, BE, CD	A, DF	B, C, DF	B, E	B, CD
C, D, AB	C, BD	C, BDE	C, BDF	C, BE
D, F	D, BC	D, BE	E, AB	E, BD
E, BDF	F, BD	F, BDE	F, CD	AB, CD
AB, DF	BC, DF	BE, DF		

Figure 4. A list of groups of GMUs that form 28 Type 1 ND constraints for the example forest in Figure 2.

where

X_{mt} = a binary variable whose value is 1 if management unit m is to be harvested in period t for t = 1, 2, ... T; 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),

M = the number of management units in the forest,

T = the number of periods in the planning horizon,

c_{mt} = the net discounted revenue per hectare if management unit m is 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,

H_t = the total volume of sawtimber in m^3 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 nondeclining 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%),

P_i = the set of indexes corresponding to the management units in path i,

n_{Pi} = the number of management units in path i,

Age^T_{mt} = the age of management unit m at the end of the planning horizon, if it is harvested in period t, and

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

Equation (11) specifies the objective function of the problem, namely to maximize the discounted net revenue from the forest during the planning horizon. The first set of constraints (12) are logical constraints. They require a management unit to be assigned to at most one prescription, including a do-nothing prescription. The second set of constraints (13) are harvest accounting constraints. They sum the harvest volume for each period and assign the resulting value to the harvest variables (H_t). Constraint sets (14) and (15) are flow constraints. Constraint set (16) represents the adjacency constraints generated with the Path Algorithm. These constraints assume that the exclusion period equals the length of a planning period. The structure is easy to generalize to alternative exclusion periods which are integer multiples of a planning period (see, for example, Snyder and ReVelle 1997). Constraint (17) is an ending age constraint. It requires the average age of the forest at the end of the planning horizon to be at least $\overline{Age}^T$ years, preventing the model from overharvesting the forest. Constraint (18) identifies the management unit treatment alternative variables as binary.

Constraint sets (12) and (16) must be replaced in order to formulate the ARM using the GMU approach. As discussed above, constraint set (12) is replaced by the following:

$$\sum_{u \in G_m} \sum_{t=0}^{T} X_{ut} \le 1 \qquad \text{for} \quad m = 1, 2, ..., M_o \qquad (19)$$

where

M_O = the set of original management units (note that in this case M represents the complete set of management units, including GMUs), and

G_m = the set of management units in M that include original management unit m.

Constraint set (16) is replaced with the following:

$$\sum_{m \in C_i} X_{mt} \le 1 \qquad \text{for all } C_j \text{ and } t = 1, 2, ..., T \qquad (20)$$

where C_j = the set of indexes corresponding to the j th pair of adjacent management units (for Pairwise adjacency constraints) or the jth nondominated set of mutually adjacent management units (for Type 1 ND adjacency constraints).

Example Problems

To demonstrate the proposed methods of formulating ARMs, we created two hypothetical example forests with 50 and 80 management units, respectively. Both example forests were created with MAKELAND, a program which generates hypothetical random forest maps (Braze 1999, McDill and Braze 2000). The map for the 50 unit example problem is shown in Figures 5, 6, and 7. In the figures, the pair of numbers inside each management unit indicates the management unit's ID number and its initial age class. For example, the management unit in the upper left-hand corner of the 50 unit forest is number 48, and it is in age class 1 (ages 0 to 20). In both hypothetical forests, the average area of a management unit is 20 ha. The largest management units are 32.6 ha in the 50 unit forest and 39.1 ha in the 80 unit forest; the smallest management units are 10.5 ha and 10.7 ha, respectively. All management units are assumed to be stands of the same forest type and site quality; management units differ only in their ages.

MAKELAND assigns a forest age class to the polygons in the maps according to a target age-class distribution. Actual age-class distributions tend to vary slightly from the target because of the discrete nature of the age-class assignments. The resulting age-class distributions of the two forests are shown in Table 1. The optimal economic rotation for stands in both forests is 80 years.[8] Thus, the forests could be described as overmature, with approximately 29% of their areas past the optimal rotation. If strict Faustmann rotations were followed on each management unit, more than half of each forest would be harvested over the next 20 yr.

Several problems were created for each of these two forests to demonstrate and compare the proposed ARM formulations with each other and with URM models. Solu-

Table 1. The age-class distributions of the example forests.

Age class (yr)	50 unit forest		80 unit forest	
	Area (ha)	Percent	Area (ha)	Percent
0 to 20	114.9	11.5	152.4	9.5
21 to 40	137.9	13.8	250.7	15.7
41 to 60	226.7	22.7	331.6	20.7
61 to 80	228.9	22.9	399.6	25.0
>80	291.6	29.1	465.7	29.1
Total	1,000.0	100.0	1,600.0	100.0

tions were obtained for each example problem with CPLEX® V6.5 using a 500 MHz Pentium III computer with 384 Mb of RAM. All problems were solved to a 0.0001 percent gap[9] to minimize the likelihood of "false" solution differences between problems due simply to stopping the branch and bound algorithm too soon. In addition, the time it takes CPLEX® to solve a given problem differs somewhat each time the problem is solved. This source of variation was essentially eliminated by solving each problem five times and averaging the solution times. In all cases, the five solutions obtained for a given problem were identical.

URM problems were formulated first for both forests, using both Type 1 ND and Pairwise constraints. Spatially unrestricted models—with no adjacency constraints—were also formulated for each forest. The number of constraints, number of variables, objective function values, solution times, average harvest opening size, and coefficient of variation (CV) of harvest opening size are reported in Table 2 for the URM and the unrestricted models. As one should expect, for a given forest the solutions for the URM models were the same regardless of the adjacency constraint formulation. Consistent with results reported in McDill and Braze (2000), the Pairwise formulations took longer to solve than the Type 1 ND formulations. Surprisingly, solution times for both forests were greater for the spatially unconstrained problems than for the URM models. In the 80 stand case, the difference was quite large. Figure 5 shows the URM solution for the 50 stand forest.

ARM formulations were created using both the GMU approach and the Path Algorithm. Three key parameters were considered that can affect the efficiency of the ARM formulations. The first of these parameters is the maximum harvest area. Clearly, the number of combinations of contiguous stands that can be harvested concurrently—and, hence, model size—will be larger for larger maximum harvest areas (or, conversely, for smaller management units). To assess the impact of this parameter on the model size and solution time, problems were created for both forests using maximum harvest areas of 40 ha and 48.6 ha. The smaller area was chosen because it is larger than the largest stand in either

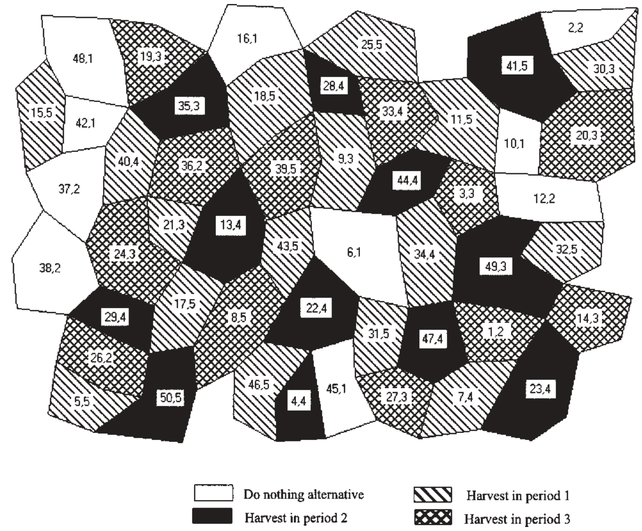

Figure 5. Harvest schedule map for the 50 stand example—URM solution.

forest. This area is also about four times the size of the smallest stands in the maps. The larger area, 48.6 ha, is the maximum harvest area specified by the AFPA's Sustainable Forestry Initiative (AF&PA 2000).

The second parameter that can affect the efficiency of the ARM formulation is the maximum age difference among a group of contiguous stands in order to consider harvesting them concurrently. The size and complexity of the ARM formulation for a forest tends to increase as the number of stands that can be harvested concurrently increases. Therefore, if a group of contiguous stands are unlikely to be harvested concurrently—i.e., if they are sufficiently different, especially in terms of their maturity—it seems reasonable to ignore the possibility of harvesting them jointly. Thus, for example, while the combined area of stands 42 and 15 in Figure 5 is only 28.1 ha, we probably do not need to consider harvesting these stands concurrently since their ages differ by four age-classes—approximately 80 yr. To compare the potential losses in objective function values with the potential time savings from limiting the range of age differences within groups of contiguous stands allowed to be harvested concurrently, we varied the maximum age-class difference from zero to four. (Four age classes is the largest possible difference.)

The third parameter we considered in our example problems is the maximum number of stands that can be harvested concurrently. As noted earlier, the Path Algorithm becomes increasingly complex as the number of stands that can be included in a path constraint is increased. Similarly, the number of GMUs also tends to increase, which increases the

Table 2. Problem formulation and solution information for URM (Type 1 ND and Pairwise) and spatially unrestricted (no adjacency constraints) 50 and 80 stand problems.

	50 stand			80 stand		
	Type 1 ND	Pairwise	None	Type 1 ND	Pairwise	None
Number of variables	184	184	184	295	295	295
No. adjacency constraints	153	241	0	252	397	0
Average solution time (sec)	0.7	1.0	1.6	1.1	1.9	13.5
Objective function ($)	2,781,670	2,781,670	2,994,838	4,679,628	4,679,628	4,848,105
Average harvest area (ha)	20.1	20.1	41.7	20.4	20.4	35.6
Coef. of var. of harvest area	0.284	0.284	1.089	0.341	0.341	0.630

size of the GMU formulation of the problem. Allowing a maximum path length of three stands is equivalent to allowing a maximum of two stands in a GMU. Similarly, allowing a maximum path length of four stands is equivalent to allowing a maximum of three stands in a GMU. Path lengths greater than four stands complicate the Path Algorithm considerably, and, while these complexities are not insurmountable, path lengths greater than four were not considered here.

The results for the 50 stand forest problems and the 80 stand forest problems are presented in Tables 3 and 4, respectively. There were no groups of three contiguous stands with a combined area less than 40 ha in the 50 stand forest, so only three cases were considered for that forest: (1) a 40 ha maximum harvest area, with a maximum of two adjacent stands harvested concurrently, (2) a 48.6 ha maximum harvest area, with a maximum of two adjacent stands harvested concurrently, and (3) a 48.6 ha maximum harvest area, with a maximum of three contiguous stands harvested concurrently. The same three cases are considered in Table 4 for the 80 stand forest; also, there were a few groups of three contiguous stands whose combined areas were less than 40 ha in the 80 stand forest, so that case is also included in Table 4. For each case, the maximum age class difference was varied between zero and four age classes.

Figure 6 illustrates the solution to the ARM model for the 50 stand forest with a 48.6 ha maximum harvest area and a maximum age difference between stands allowed to be harvested concurrently of at least two age classes. (Allowing a larger maximum age difference resulted in the same solution.) In all cases, the solution to the GMU formulation was the same as the solution to the Path formulation for a given problem. To test for the possibility that the Path Algorithm might yield different solutions depending on the order that polygons are added to the cluster, three problems were reformulated ten times using different initial polygons each time.[10] In all cases, the resulting solutions were the same.

The results in Tables 3 and 4 indicate how model size is

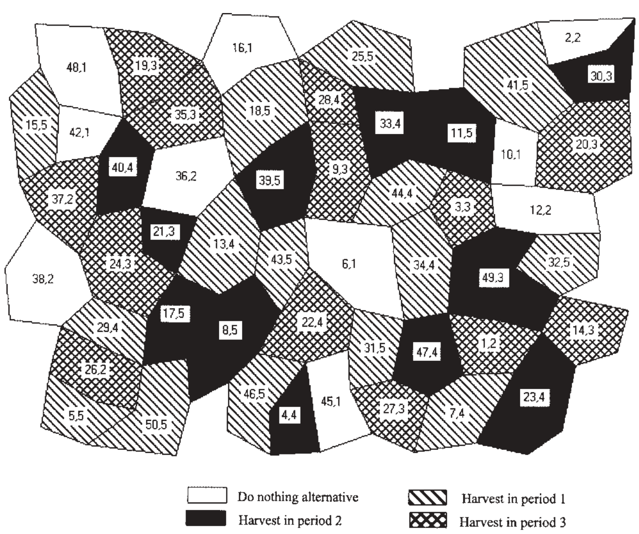

Figure 6. Harvest schedule map for the 50 stand example—ARM solution with a maximum harvest area of 48.6 ha, an age difference restriction of two age classes or larger, a maximum of two adjacent stands harvested concurrently, and no fixed cost.

affected by the type of formulation (Path or GMU) and the three parameters considered: the maximum harvest area, the maximum number of contiguous stands allowed to be harvested concurrently, and the maximum age difference. For the Path Algorithm, the number of variables is not affected by the three formulation parameters, but the number of constraints tends to increase with each of the three parameters. The number of path constraints was always larger than the number of Type 1 constraints for a URM formulation of a given problem. For more restrictive settings of the three formulation parameters, the number of path constraints was less than the number of Pairwise constraints for the URM formulation. As the three parameters were increased, however, the number of constraints in the Path formulation became larger than the number of Pairwise constraints. For the GMU formulations, both the number of variables and the number of constraints increase with all three parameters. Furthermore, both the number of variables and the number of constraints is substantially larger with GMU models than with the Path Algorithm. Thus, model size is more of a concern with the GMU approach than with the Path Algorithm.

The differences in model size are generally reflected in solution times. Solution times tend to grow with the number of constraints and variables. Solution times for the GMU models were consistently longer than for the Path models, and solution times for the Path models were consistently longer than solution times for either URM formulation (Type 1 or Pairwise). Interestingly, solution times often decreased as the maximum age difference was increased, even though this lead to a larger model. The most significant example of this is the GMU formulation for the 80 stand problem with a 48.6 ha maximum harvest area and a maximum of two adjacent stands harvested concurrently. Most likely this is simply a reflection of the unpredictability of solution times for these types of problems, but it also underscores the fact that reducing the number of variables and/or constraints in these problems does not always result in reduced solution times.

While the ARM models are more difficult to formulate and take longer to solve, this cost is compensated by better objective function values. With a 40 ha maximum harvest area, the best ARM solutions were 3.2 and 1.2% better than the corresponding URM solutions for the 50 stand and 80 stand forests, respectively. With a 48.6 ha maximum harvest area, the improvement was 4.8 and 2.7%, respectively. These gains are not especially large, but they are significant when viewed as a percentage reduction in the cost of the adjacency constraints. For example, in the case of the 50 stand forest, the cost of the URM adjacency constraints—as measured by the difference between the URM objective function value and the spatially unconstrained objective function value (Table 2)—is \$213,168, or \$213/ha. With a 40 ha maximum harvest area, using an ARM model reduces this cost to \$122,826—a 42.4% reduction. Using an ARM formulation reduces the cost of adjacency for the 50 stand forest even further, by 62.8%, when the maximum harvest area is 48.6 ha. Similar figures for the 80 stand forest are 32.2% with a 40 ha maximum harvest area and 74.7% with a 48.6 ha maximum.

The results clearly show that allowing the concurrent harvest of contiguous stands with large age differences tends to increase the model size with little or no corresponding improvement in the solution. In our examples, there was no benefit from considering the concurrent harvest of contiguous stands with an age difference greater than two age classes—about half of a rotation. Furthermore, the gains from considering age differences greater than one age class (one-fourth of a rotation) were very small. The largest improvement in the objective function obtained by allowing age differences of up to two age classes versus allowing a maximum age difference of one age class was only 0.06% (in the case of the 80 stand forest with a maximum harvest area of 40 ha).

The improvement in the objective function that can be achieved by allowing up to three contiguous stands to be harvested concurrently clearly depends on the parameters of the problem, but for the examples presented here there was little gain from considering the concurrent harvest of more than two adjacent stands. As mentioned earlier, there were no groups of three contiguous stands with a combined area of less than 40 ha in the 50 stand forest. Even in the 80 stand forest, however, where such groups did exist, none were scheduled for harvest in the best solution to the problem. When the maximum harvest area was 48.6 ha, allowing groups of three stands to be harvested together yielded a better solution for the 50 stand forest only in the case that required contiguous groups of stands to be in the same age class in order to be harvested together. The best overall solution was obtained for the 80 stand forest with a 48.6 ha maximum harvest area when three contiguous stands were allowed to be harvested together. However, this solution was less than 0.015% better than the best solution obtained with a maximum of two adjacent stands harvested concurrently.

One of the potential drawbacks of the ARM approach is that it could conceivably decrease the variability of stand sizes, and, hence, the spatial diversity of the forest (Hunter 1990, Chap. 6). The danger is that small stands will be eliminated as they are harvested together with adjacent stands. Furthermore, very large stands will not be deliberately created due to the maximum harvest area restriction. (Nature may create a few, of course.) It seems inevitable that these two effects would squeeze the stand size distribution from both ends, reducing the diversity of stand sizes. We did not necessarily observe this effect in our examples, however. If future stand size is determined by harvest opening sizes, ARM models will tend to create larger stands over time than URM models. (See Tables 3 and 4, average harvest area.) However, in nearly all cases, the coefficient of variation of the size of the harvest areas was greater for the ARM models than for the corresponding URM models. On the other hand, the coefficient of variation of harvest opening size did increase with larger maximum harvest areas, and the variability of the harvest opening size was much larger for the spatially unrestricted models.

Accounting for Fixed Timber Sale Administration Costs

As discussed earlier, one advantage of the GMU formulation is that it can recognize the direct cost savings that can be often be realized by combining smaller timber sales. To demonstrate this, we developed two more models that included a fixed administrative cost of $1,500 associated with a timber sale.[11] Groups of contiguous stands which are harvested concurrently incur this fixed cost only once. These models are based on the 50 stand example forest, and, other than the fixed timber sale cost, they are the same as the Path and GMU models presented in Table 3 with a 48.6 ha maximum harvest area, a maximum age difference of two age classes, and a maximum of two adjacent stands harvested concurrently. The solution to this problem without fixed costs is depicted in Figure 6.

Table 3. Problem formulation and solution information for 50 stand ARM problems.

Max age diff.	No. variables		No. adj. constr.		Solution time (sec)		Objective function value ($)	Number of GMU		Average harvest area (ha)	C.V.* of harvest area	% of adj. cost saved
	Path	GMU	Path	GMU	Path	GMU		Created	Used †			
40 ha maximum harvest area/maximum of two adjacent stands harvested concurrently												
0	184	206	224	263	3.0	3.6	2,839,895	8	5	23.0	0.344	27.3
1	184	259	225	524	5.2	28.7	2,872,012	28	7	23.9	0.348	42.4
2	184	276	240	618	7.2	29.0	2,872,012	36	7	23.9	0.348	42.4
3	184	284	245	668	8.2	16.2	2,872,012	41	7	23.9	0.348	42.4
4	184	287	249	694	9.5	41.2	2,872,012	44	7	23.9	0.348	42.4
48.6 ha maximum harvest area/maximum of two adjacent stands harvested concurrently												
0	184	235	220	396	5.0	10.8	2,876,037	18	7	23.7	0.452	44.3
1	184	322	300	982	13.0	111.4	2,915,466	50	8	25.1	0.438	62.8
2	184	360	378	1,377	13.8	203.7	2,915,466	68	8	25.1	0.438	62.8
3	184	374	397	1,591	19.2	139.0	2,915,466	77	8	25.1	0.438	62.8
4	184	382	416	1,787	16.9	243.6	2,915,466	85	8	25.1	0.438	62.8
48.6 ha maximum harvest area/maximum of three contiguous stands harvested concurrently												
0	184	241	217	438	3.9	7.4	2,881,220	20	5	23.6	0.446	46.7
1	184	328	296	1,030	13.8	90.1	2,915,466	52	8	25.1	0.438	62.8
2	184	377	371	1,610	15.8	140.9	2,915,466	75	8	25.1	0.438	62.8
3	184	400	386	1,998	13.3	794.5	2,915,466	91	8	25.1	0.438	62.8
4	184	419	404	2,485	21.7	760.9	2,915,466	110	8	25.1	0.438	62.8

* Coefficient of variation.
† Used by the optimal solution.

Table 4. Problem formulation and solution information for 80 stand problems.

Max age diff.	No. variables Path	No. variables GMU	No. adj. constr. Path	No. adj. constr. GMU	Solution time (sec) Path	Solution time (sec) GMU	Objective function value ($)	Number of GMU Created	Number of GMU Used†	Average harvest area (ha)	C.V.* of harvest area	% of adj. cost saved
40 ha maximum harvest area/maximum of two adjacent stands harvested concurrently												
0	295	334	382	461	8.4	14.9	4,699,035	14	5	22.6	0.346	11.5
1	295	412	383	784	18.0	46.0	4,730,829	45	10	23.9	0.365	30.4
2	295	458	436	1,046	52.3	343.2	4,733,804	66	9	23.7	0.367	32.2
3	295	466	447	1,124	67.5	254.7	4,733,804	71	9	23.7	0.367	32.2
4	295	469	453	1,151	43.1	236.9	4,733,804	74	9	23.7	0.367	32.2
40 ha maximum harvest area/maximum of three contiguous stands harvested concurrently												
0	295	334	382	461	10.6	14.9	4,699,035	14	5	22.6	0.346	11.5
1	295	418	380	836	20.3	90.8	4,730,829	49	10	23.9	0.364	30.4
2	295	478	429	1,174	43.6	443.3	4,733,804	76	9	23.7	0.367	32.2
3	295	486	440	1,252	44.5	289.8	4,733,804	81	9	23.7	0.367	32.2
4	295	491	446	1,295	42.2	253.9	4,733,804	86	9	23.7	0.367	32.2
48.6 ha maximum harvest area/maximum of two adjacent stands harvested concurrently												
0	295	361	370	573	48.7	100.8	4,778,588	23	8	23.1	0.452	58.7
1	295	490	447	1,323	106.3	1,063.1	4,804,717	72	16	26.9	0.433	74.2
2	295	586	631	2,317	140.1	10,956.3	4,804,717	113	16	26.9	0.433	74.2
3	295	601	681	2,707	75.9	9,029.5	4,804,717	123	16	26.9	0.433	74.2
4	295	609	710	2,904	99.6	3,971.9	4,804,717	131	16	26.9	0.433	74.2
48.6 ha maximum harvest area/maximum of three contiguous stands harvested concurrently												
0	295	373	364	651	69.9	115.5	4,778,588	27	8	23.1	0.452	58.7
1	295	526	432	1,655	116.9	761.9	4,804,717	88	16	26.9	0.433	74.2
2	295	711	592	3,907	154.7	12,792.3	4,805,425	167	18	28.6	0.433	74.7
3	295	745	646	5,160	125.4	52,435.2	4,805,425	192	18	28.6	0.433	74.7
4	295	762	676	5,649	74.5	58,076.2	4,805,425	209	18	28.6	0.433	74.7

* Coefficient of variation.
† Used by the optimal solution.

The Path formulation cannot recognize the direct cost savings due to combining the administrative costs of concurrent contiguous timber sales, and, other than the value of the objective function, the solution to the Path formulation was the same with the fixed cost as without it (see Figure 6). This Path model solution selected eight pairs of adjacent stands to be harvested in the same period. The solution to the GMU model that recognized the cost savings from combining adjacent timber sales is shown in Figure 7. When these cost savings were recognized, 11 pairs of adjacent stands were scheduled to be harvested concurrently. After adjusting the Path objective function value for the fixed cost savings that would have been realized on the 8 pairs of adjacent stands scheduled to be harvested together, the GMU objective function value was only 0.04% better than the Path objective function value. The present value of the expected cost savings was $1,140 over the 60 yr planning horizon. While the improvement in the objective function was small in this case, it is likely that more significant savings would be possible in many applied situations.

Summary and Conclusions

We have demonstrated two ways to formulate ARM models as MILPs. The first approach, the Path Algorithm, generates a set of constraints that prevent concurrent harvesting of groups of contiguous stands only when their combined area exceeds the harvest area restriction. Implementing the algorithm is straightforward as long as the Path constraints do not involve more than four management units. Larger groups could be accommodated, but the algorithm would become increasingly complex. Path formulations typically require more constraints than an equivalent URM formulation with Type 1 adjacency constraints, and in our examples Path formulations took five or more times longer to solve than the corresponding URM formulations. On the other hand, the Path approach was able to reduce the cost of imposing adjacency constraints by between 30 and 75% over the URM solutions in our example problems. Theoretically, of course, the savings could be anything between 0 and 100%.

Generalized management units (GMUs), consisting of groups of individual management units whose collective areas do not exceed the maximum harvest area limit, are used

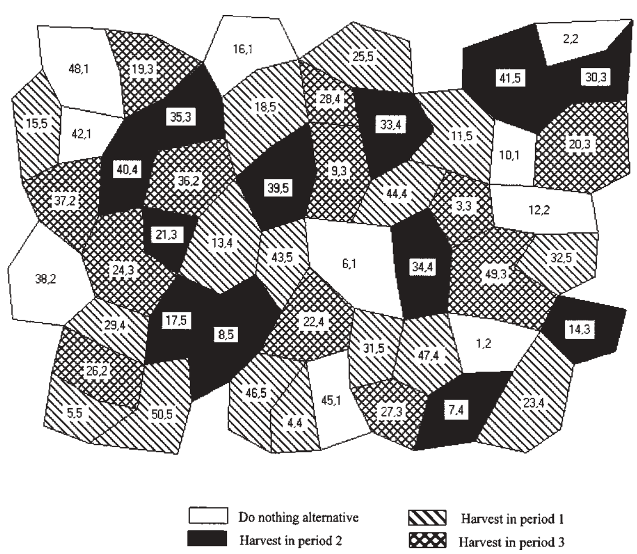

Figure 7. Harvest schedule map for the 50 stand example—GMU solution with a fixed cost of $1,500 per timber sale and a maximum harvest area of 48.6 ha, a maximum age difference of two age classes for jointly managed stands, and a maximum of two adjacent stands harvested concurrently.

in the second proposed MILP ARM formulation. GMU models require more variables and constraints than equivalent models formulated with the Path Algorithm, and solution times tend to be considerably longer. One advantage of the GMU approach, however, is that it allows the model to recognize certain cost savings that can be realized by jointly managing adjacent units, such as sharing fixed sale administration or harvesting costs. Such cost savings can be significant in forestry. It is important to note that such cost savings can also be achieved through the joint management of stands which are not necessarily adjacent. This aspect of the approach will have utility to managers whether or not they are concerned with maximum harvest area restrictions. As long as two or more management units are within close enough proximity so that cost savings can be achieved through their joint management, it may be useful to model these groups collectively as GMUs. Such groups could be larger than the maximum harvest area, as long as the component stands of the GMU are not adjacent. GMU models have the potential, therefore, to more accurately account for management costs in harvest scheduling, thus providing better on-the-ground solutions than those provided by other models. Since "pretty good" suboptimal solutions can often be obtained with significantly less effort than optimal solutions (McDill and Braze 2001), an important question is whether it is better to have a suboptimal solution to a model which more closely reflects real-world costs and constraints or whether it is better to have an optimal solution to a model which approximates the real world problem less well. The answer to this question will, of course, vary from one situation to another.

The GMU approach is also useful simply because it provides an alternative way to formulate ARMs. This formulation may prove useful for setting up ARM formulations for heuristic solution algorithms or for modeling spatial conditions such as interior space or forest patch size distributions. In addition, having two ways to formulate complex problems provides a useful check on the procedures used to formulate the models. In this case, the fact that both ARM formulations of the example problems resulted in the same solutions provides confidence that both formulations have been correctly specified.

Both of the ARM formulations presented here will become increasingly complex and difficult to formulate and solve if the original management units are significantly smaller than the maximum harvest area. In our examples, three was the largest number of contiguous management units that could be harvested concurrently. In principle, the methods presented here can be applied to problems with stand sizes that are smaller relative to the maximum harvest area—i.e., with groups of four or more contiguous stands that can be harvested concurrently. However, problem formulation and solution will become increasingly complex. In any case, our examples suggest that even when there are many groups of four or more stands that can be harvested concurrently, there may be diminishing gains from considering these possibilities. In our examples, the gains from considering the possibility of harvesting groups of three contiguous stands rather than only allowing pairs of adjacent stands to be harvested concurrently were very small. Furthermore, we have also shown how the size and complexity of the models presented here can be reduced by placing some *a priori* limits on which adjacent stands will be considered for joint management. For example, our results suggest that the benefit of considering the concurrent harvest of adjacent stands whose ages differ by half of a rotation or more will generally be negligible.

The methods presented here clearly have their limitations for solving problems with many combinations of four or more adjacent stands that can be harvested simultaneously without violating the maximum harvest area restriction. For those cases where combinations of four or more stands that can and should be harvested concurrently are rare, the methods proposed here provide a practical way to comply with harvest area restrictions. Even for the more difficult cases where such combinations are common, however, these methods will be useful because they can provide exact solutions (or at least upper or lower bounds in cases where problems take a very long time to solve to optimality) against which the solutions provided by alternative algorithms can be compared.

A possible problem with ARMs is that they tend to reduce the frequency of smaller harvest areas, moving the average size of harvest areas in the direction of the maximum harvest area. This could reduce the spatial diversity of the forest. Somewhat unexpectedly, however, in our examples the coefficients of variation of the harvest areas were larger for ARMs than for URMs. Nevertheless, combining stands into larger management areas may not always be desirable. One possible solution to this problem, if it is a problem, might be to add constraints requiring minimum numbers of harvests in different ranges of sizes. It is more obvious how this can be accomplished with a GMU formulation than with a Path formulation.

A promising aspect of ARMs is that they are a step towards models which are not bound by *a priori* management boundaries. The models presented here allow management units to be recombined by the model into more efficient configurations which may be difficult to recognize without models such as these. Unfortunately, however, spatially explicit forest management problems are difficult to solve even when the management units are predefined in sizes and shapes that correspond roughly to the areas that will be treated on the ground. As forest management researchers move further in the direction of building treatment areas up from smaller and smaller management units, the resulting problems undoubtedly will be considerably more difficult to solve. As in any modeling exercise, we must be careful to ensure that the gains from such increased complexity outweigh the costs.

Endnotes

[1] There are many ways to define adjacency. Adjacency is defined in this paper as two management units sharing a common boundary arc. The methods described in this article would not be affected if an alternative definition of adjacency were used.

[2] To simplify the discussion, we assume here that the forest consists of a set of contiguous management units. The algorithm can easily be generalized to relax this assumption.

[3] The algorithm for identifying paths can be streamlined so that some redundant paths are not generated. For example, if the arcs in the

adjacency network are ordered, once all paths originating with a given arc have been identified, that arc does not need to be included in paths originating from arcs lower on the list. Even though a unique path may be created by starting with a lower arc and branching on the original arc, that path will always be redundant. (Proof of this is available from the authors.)

[4] Note that if constraint (3) holds, then the constraint $x_{At} + x_{Bt} + x_{Dt} \leq 2$ must also hold.

[5] At this point, we could have generated a forked path DB-DA, but we do not need to. This is an example of how the efficiency of the algorithm can be increased by not returning to an arc—DA in this case—that is higher on the arc list—than DB in this case—and has already been fully expanded within the current cluster. It is only necessary to consider branches off of DB (such as DC) that are lower on the arc list.

[6] GMUs can be identified by calculating the combined areas of all pairs of adjacent stands, identified from the adjacency list. GMUs are then created for each adjacent pair whose combined area is less than the maximum harvest area. Next, records are added to the adjacency list describing the adjacency relationships of the new GMUs. Combined areas are calculated for these new adjacency records. Again, GMUs are created for any corresponding groups of stands whose combined areas are less than the maximum harvest area, and the adjacency list is again updated. This process is continued until none of the newly created adjacency records identify additional groups of contiguous stands with a combined area less than the maximum harvest area.

[7] ND stands for "nondominated" (Murray and Church 1996b, McDill and Braze 2000).

[8] The yield and economic data used in this example are the same as those described in McDill and Braze (2000), which loosely represent oak stands in Pennsylvania.

[9] The "percent gap" is the percent difference between the current best integer solution and the value of the relaxed LP objective function for the best unexplored node in the branch and bound tree (see McDill and Braze 2001). The true optimal objective function value must be within these bounds. The default gap in CPLEX® is 0.01%.

[10] The three problems were (1) the 50 stand forest problem with a 48.6 ha maximum harvest area, a maximum age difference of two age classes, and a maximum of two adjacent stands harvested concurrently, (2) the same problem with up to three contiguous stands harvested concurrently, and (3) the same problem as (1) for the 80 stand forest.

[11] Nelson et al. (1991) also discuss fixed costs associated with management activities.

Literature Cited

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

Borges, J.G, H.M. Hoganson, and D.W. Rose. 1999. Combining a decomposition strategy with dynamic programming to solve spatially constrained forest management scheduling problems. For. Sci. 45(2):201–212.

Boston, K., and P. Bettinger. 1999. An analysis of Monte Carlo integer programming, Simulated Annealing, and Tabu Search heuristics for solving spatial harvest scheduling problems. For. Sci. 45(2):292–301.

Braze, J. 1999. A comparison of alternative adjacency constraint formulations in forest planning models. M.Sc. Thesis, School of For. Resour., The Pennsylvania State Univ. 101 p.

Hoganson, H.M., and J.G. Borges. 1998. Using dynamic programming and overlapping subproblems to address adjacency in large harvest scheduling problems. For. Sci. 44(4):526–538.

Hunter, M.L., Jr. 1990. Wildlife, forests, and forestry: Managing forests for biodiversity. Prentice Hall, Englewood Cliffs, NJ. 370 p.

Johnson, K.H., and H.L. Scheurman. 1977. Techniques for prescribing optimal timber harvest and investment under different objectives—discussion and synthesis. For. Sci. Monogr. 18.

Lockwood, C., and T. Moore. 1992. Harvest scheduling with spatial constraints: a simulated annealing approach. Can. J. For. Res. 23:468-478.

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., and J. Braze. 2001. Using the branch and bound algorithm to solve forest planning problems with adjacency constraints. For. Sci. 47(3):403–418.

Meneghin, B.J., M.W. Kirby, and J.G. Jones. 1988. An algorithm for writing adjacency constraints efficiently in linear programming models. P. 46–53 *in* The 1988 symposium on systems analysis in forest resources, Kent, B., and L. Davis (eds.). USDA For. Serv. Rocky Mt. For. Range Exp. Sta. Gen. Tech. Rep. RM-161.

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. 1996a. Constructing and selecting adjacency constraints. INFOR 34:232–248.

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

Nelson, J.D., and J.D. Brodie. 1990. Comparison of a random search algorithm and mixed integer programming for solving area-based forest plans. Can. J. For. Res. 20:934–942.

Nelson, J.D., J.D. Brodie, and J. Sessions. 1991. Integrating short-term, area-based logging plans with long-term harvest schedules. For. Sci. 37(1):101–122.

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

Snyder, S., and C. ReVelle. 1997. Dynamic selection of harvests with adjacency restrictions: the SHARe model. For. Sci. 43(2):213–222.

Torres-Rojo, J.M., and J.D. Brodie. 1990. Adjacency constraints in harvest scheduling: an aggregation heuristic. Can. J. For. Res. 20:978–986.

U.S. Congress. 1976. National Forest Management Act. (88 Stat. 476; 16 U.S.C. 1601-1610), October 22.

Weintraub, A., F. Barahona, and R. Epstein. 1994. A column generation algorithm for solving general forest planning problems with adjacency constraints. For. Sci. 40(1):142–161.

Yoshimoto, A., and J.D. Brodie. 1994. Comparative analysis of algorithms to generate adjacency constraints. Can. J. For. Res. 24:1277–1288.