Operations Research
Vol. 53, No. 3, May–June 2005, pp. 490–500
ISSN 0030-364X | EISSN 1526-5463 | 05 | 5303 | 0490

DOI 10.1287/opre.1040.0169

Harvest Scheduling Subject to Maximum Area Restrictions: Exploring Exact Approaches

Marcos Goycoolea
School of Industrial and Systems Engineering, Georgia Institute of Technology, 755 Ferst Drive, Atlanta, Georgia 30332-0205, mgoycool@isye.gatech.edu

Alan T. Murray
Department of Geography, Ohio State University, Columbus, Ohio 43210, murray.308@osu.edu

Francisco Barahona
IBM Watson Research Center, Yorktown Heights, New York 10598, barahon@us.ibm.com

Rafael Epstein, Andrés Weintraub
Departamento de Ingeniería Industrial, Universidad de Chile, República 701, Santiago, Chile
{repstein@dii.uchile.cl, aweintraub@dii.uchile.cl}

We consider a spatial problem arising in forest harvesting. For regulatory reasons, blocks harvested should not exceed a certain total area, typically 49 hectares. Traditionally, this problem, called the adjacency problem, has been approached by forming a priori blocks from basic cells of 5 to 25 hectares and solving the resulting mixed-integer program. Superior solutions can be obtained by including the construction of blocks in the decision process. The resulting problem is far more complex combinatorially. We present an exact algorithmic approach that has yielded good results in computational tests. This solution approach is based on determining a strong formulation of the linear programming problem through a clique representation of a projected problem.

Subject classifications: integer programming; cutting planes; environment.
Area of review: Environment, Energy, and Natural Resources.
History: Received March 2002; revisions received March 2003, December 2003; accepted February 2004.

1. Introduction

Optimization in forest planning has been widely and successfully used for several decades. Linear programming (LP) and integer programming (IP) models have been developed to deal with management policies, indicating which timber units to harvest each period to meet projected demands and satisfy silvicultural, environmental, sustainability, and other planning constraints. The spatial location of activities has long been included explicitly in decision models because it affects wildlife habitat, scenic beauty, and other environmental concerns.

Regulating spatial disturbance is now standard practice in public and private forest management (Jones et al. 1991, Barrett et al. 1998, American Forest and Paper Association 2001). A prominent approach is prohibiting the harvesting of contiguous areas larger than a specified value, similar to what was originally proposed by Thompson et al. (1973). Typically, this limit is 49 hectares (ha) but may vary depending on the region being studied. Such regulations are stipulated at the state and federal level in the United States (see Thompson et al. 1973, Jones et al. 1991, McDill et al. 2002). In particular, Boston and Bettinger (2001) detail that spatial restrictions are explicitly regulated in Oregon (49 ha) and California (17 ha) as well as internationally in Sweden (20 ha). Other states and countries limiting spatial impacts are detailed in Barrett and Gilless (2000). On a voluntary basis the Sustainable Forestry Initiative insists that participating private forests in the United States be managed so that "... average size of clear-cut harvest areas shall not exceed 120 acres ..." (American Forest and Paper Association 2001, p. 4).

A particular approach to this planning problem is referred to as the adjacency problem. If we assume that management units are less than 49 hectares and greater than 25 hectares, then if a unit is harvested in period t, none of its neighboring units can be harvested in that period or in a number of succeeding periods until regeneration occurs (called *green up requirements*). This condition can be formulated as an IP model with 0–1 variables and constraints for imposing adjacency restrictions. Solving this problem, however, is not trivial due to its NP-hard complexity. Proposed approaches for this problem include heuristics such as Tabu search (Murray and Church 1995), simulated annealing (Murray and Church 1995), Monte Carlo simulation (O'Hare et al. 1989, Nelson and Brodie 1990), and exact techniques such as dynamic programming (Hoganson and Borges 1998), column generation (Barahona et al. 1992), and formulation strengthening (Murray and Church 1996, 1997). These approaches have been relatively successful.

A new approach to this basic problem was proposed when it was noted that harvest planning units are typically composed of smaller management cells (Hokans 1983, Lockwood and Moore 1993, Murray 1999). These smaller cells constitute the basic units that can be harvested. In traditional adjacency models, referred to as the unit restriction model (URM) in Murray (1999), the management units are formed a priori by the forest planner. That is, using forest characteristics, often supported by geographical information systems (GIS), the planner forms cutting units by blocking basic cells (Barrett 1997). These basic cells typically range between 5 and 25 hectares, so a cutting unit will have from 3 to 8 basic cells. Thus, using basic cells one is left with a different harvesting problem, where area restrictions cannot be imposed using adjacency constraints as is done in the URM. This approach is referred to as the area restriction model (ARM) in Murray (1999).

The ARM is an important and difficult forest scheduling problem. Thus far, approaches proposed to solve this problem have mostly been heuristic (Hokans 1983, Lockwood and Moore 1993, Barrett et al. 1998, Clark et al. 2000, Richards and Gunn 2000, Boston and Bettinger 2002). There also have been two lines of exact solution approaches. The first of these was proposed by McDill et al. (2002), and further pursued by Crowe et al. (2003) for medium to large size, multiperiod problems. The second line of exact solution approaches is introduced in Martins et al. (1999, 2000), though in a different context. We further pursue this approach.

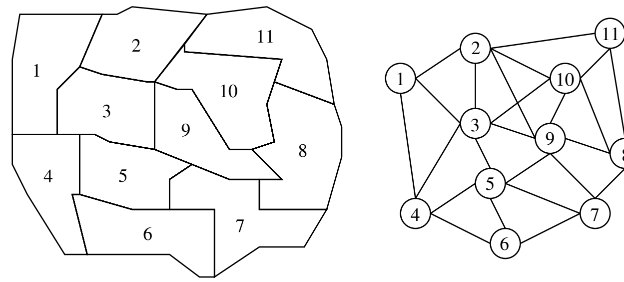
This paper develops an exact algorithmic approach for the area restriction problem. This paper makes two contributions: (1) it introduces a tight (with regard to the LP-relaxation), compact, and easy to formulate model for effectively solving medium to large size problems; and (2) it introduces a new methodology, which we call constraint projection, for generating valid inequalities. In §2, area restrictions are described. This is followed in §3 by several model formulations for imposing area restrictions. In §4, model formulation enhancements are studied and a new model formulation is proposed. In §5, application results are presented and discussed, and in §6 conclusions are given.

2. Area Restrictions

Consider a forest partitioned into basic cells for which area, timber volume, and net benefit from harvesting are known. The partitioned region may be represented by means of a graph $G(V, E)$, where the set V of nodes corresponds to the basic cells (or units), and $(u, v) \in E$ if and only if cells u and v are adjacent.

We consider two ways of defining adjacency. Two cells are weakly adjacent if they share a finite set of points. Alternatively, two cells are strongly adjacent if they share

Figure 1. Graph representation of a forest.



a common border or an infinite number of points. Given this definition, if two cells are strongly adjacent, then they are also weakly adjacent. In the literature (see Daust and Nelson 1993, among others) it is recognized that alternative definitions of adjacency exist. However, the above distinction is unique and has implications for problem structure, as will be discussed.

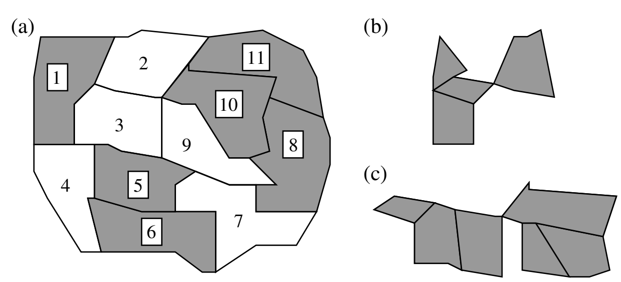
In Figure 1, we illustrate how a forest region is represented using a graph. The set of nodes corresponds to cells and arcs link weakly adjacent cells. In this example, arcs $(2, 9)$ and $(3, 10)$ are weakly adjacent pairs of cells. Note that an arc would not join these nodes in a strong adjacency graph.

Weak adjacency is commonly relied upon in forest management. However, the concept of strong adjacency can lead to significantly different spatial patterns of solutions in some cases. Further, adjacency type will influence our developed solution approach.

Using basic cells, we must characterize all potential groupings of cells that could be harvested. A *feasible cluster* is a set of strongly contiguous cells whose total area does not exceed the established maximum area restriction, 49 hectares in this case. Feasible clusters represent valid cutting zones or blocks. Two feasible clusters are *noncompatible* if they share a common cell or are weakly adjacent.

Consider the forest partition shown in Figure 2a, where basic cells range in size from 10 to 13 hectares. For this example, feasible clusters may be formed only by up to three or four cells if there is a 49 hectares maximum. As a result, each shaded region corresponds to a feasible cluster, and all are mutually compatible.

Figure 2. (a) Feasible solution. (b) An infeasible cluster. (c) A noncompatible pair of clusters.



In general, the problem consists of determining a harvest schedule with the greatest economic return such that no more than 49 contiguous ha of forest are harvested in any time period. In terms of our notation, this is equivalent to harvesting for each time period only compatible sets of feasible clusters. By describing the problem in this way, we introduce an important distinction between strong and weak adjacency in that the former definition is used to *construct* feasible clusters and the latter to *separate* clusters. Traditional ARM approaches do not make this distinction and generally use weak adjacency for both purposes. This distinction is important in that it may result in variations in the spatial structure of solutions. In Figure 2b we present a cluster which is weakly, but not strongly, connected; and in Figure 2c we present a pair of clusters that are individually feasible but noncompatible.

The choice of weak versus strong adjacency can impact the associated IP in many ways. Strong adjacency results in fewer feasible clusters and fewer noncompatibilities in contrast to weak adjacency. Even so, the approach developed in this paper is not dependent on these definitions. In fact, they may be adapted so as to address different concerns. For example, two cells may be alternatively defined as being weakly adjacent if they are within a certain distance. In the same way, two cells may be defined as strongly adjacent if the common border exceeds a certain length. As will be discussed later, our proposed approach may even be used without this distinction. As such, our developed approach may be used to solve the classic ARM. The new modeling approach developed in this paper will henceforth be referred to as the extended area restriction model (EARM) and is based on the notion of being able to distinguish between weak and strong adjacency, unlike the ARM.

3. Modeling Approach

To solve the area-based harvesting scheduling problem, we propose using a set-packing type IP formulation, similar to that used by Martins et al. (2000). We begin by presenting a simple formulation that defines one constraint for each pair of noncompatible clusters. Afterward, we show several ways to strengthen and extend this formulation.

3.1. Pairwise Adjacency Formulation

Consider a planning problem with T periods. The time horizon is sufficiently short that harvested units will not have enough time to mature for multiple harvests.

Let $\Lambda = \{S_1, S_2, \ldots, S_n\}$ be the set of all feasible clusters in $G(V, E)$. Further, let $c_{v,t}$ and a_v represent the net harvest benefit in period t and area associated with each cell v in V.

For each cluster $S \in \Lambda$, we define its benefit and area as the sum of the benefits and areas of the cells that it is composed of

$$c_{S,t} = \sum_{v \in S} c_{v,t}, \quad a_S = \sum_{v \in S} a_v.$$

If there are fixed costs for harvesting a cluster, in addition to the costs associated with harvesting each cell, these can be incorporated in the above definition by deducting the costs from the net benefit.

We may now formulate our problem in a straightforward way as an IP using the following variables:

$$x_{S,t} = \begin{cases} 1 & \text{if cluster } S \text{ is harvested in time period } t, \\ 0 & \text{if not.} \end{cases}$$

Extended Area Restriction Model-1 (EARM-1)

$$\text{Maximize} \sum_{(S,t)} c_{S,t} x_{S,t},$$

subject to

$$\sum_{(S,t) \mid v \in S} x_{S,t} \leqslant 1 \quad \text{for each cell } v, \tag{1}$$

$$x_{S,t} \in \{0,1\} \quad \text{for each cluster } S \in \Lambda, \text{ for each time period } t, \tag{2}$$

$$x_{S,t} + x_{S',t} \leqslant 1 \quad \text{for each time period } t, \text{ for each pair } S, S' \text{ of noncompatible clusters.} \tag{3}$$

The objective maximizes the net benefit over time horizon T. Constraints (1) state that each basic cell may be harvested at most once during the time horizon. Constraints (2) indicate the integer definition of the variables. Constraints (3) state that for any pair of noncompatible clusters, only one may be harvested in the same period. A small modification will allow this to be extended to more than one period to account for green up requirements. Green up is the time lapse needed for a harvested area to grow to a minimal height, allowing neighboring cells to be harvested. For instance, if g is the green up time requirement, Constraints (3) may be replaced for each t by a constraint of the form

$$\sum_{l \in t \ldots t+g-1} (x_{S,l} + x_{S',l}) \leqslant 1 \quad \text{for } t = 1, \ldots, T-g. \tag{4}$$

This generalization may be applied to all the adjacency restrictions presented in this paper.

This model, while intuitively simple, has drawbacks. The number of Constraints (3) equals the number of pairwise noncompatible clusters, which grows very fast with the number of basic cells. In addition, this is a very weak formulation because the continuous LP relaxation leads to mostly fractional solutions (Nemhauser and Sigismondi 1992). As a result, branch and bound is not particularly successful when using this formulation for the EARM (or ARM).

3.2. Arc Adjacency Formulation

As discussed in the previous section, EARM-1 has inherent drawbacks that make it very difficult to solve by use of branch-and-bound algorithms. This suggests the need to find a more compact and tighter formulation. Martins et al. (2000) present an alternative for a variation of the classic ARM that can be easily adapted to solve the EARM.

Consider two noncompatible clusters S_i and S_j in Λ. Because they are noncompatible, they must either be (weakly) adjacent or overlapping (share common cells). Thus, there must exist some weak adjacency arc (u, v) in G such that $u \in S_i$ and $v \in S_j$. Thus, noncompatibility may be prohibited by defining one constraint for each weak adjacency arc in E (i.e., imposing that in any solution only one feasible cluster may intersect each weakly adjacent arc).

Define $\Lambda(u, v)$ as the set of all clusters S in Λ such that $u \in S$, $v \in S$, or both u and $v \in S$. The EARM may be formulated in an alternative way by focusing on arcs in constraint set (3).

Extended Area Restriction Model-2 (EARM-2)

Maximize $\sum_{(S,t)} c_{S,t} x_{S,t}$,

subject to (1), (2), and

$$\sum_{S \in \Lambda(u,v)} x_{S,t} \leqslant 1 \quad \text{for each time period } t, \quad \text{for each weak adjacency arc } (u,v) \text{ in } G. \tag{3a}$$

Constraints (3a) prohibit noncompatible clusters from being harvested in the same time period. Note that in EARM-2 all feasible clusters appear in Constraints (3a).

Suppose we have three mutually adjacent cells, any two of which could be harvested feasibly but which are of a size so that all three cannot be harvested at the same time (assume a one-time-period model for simplicity).

Then, the three units form six feasible clusters ($\{1\}$, $\{2\}$, $\{3\}$, $\{1,2\}$, $\{2,3\}$, $\{1,3\}$). Because there are three adjacency arcs, in the EARM-2 model we have

$$\Delta(1,2) = (\{1\}, \{2\}, \{1,2\}, \{2,3\}, \{1,3\}),$$

$$\Delta(1,3) = (\{1\}, \{3\}, \{1,2\}, \{2,3\}, \{1,3\}),$$

$$\Delta(2,3) = (\{2\}, \{3\}, \{1,2\}, \{2,3\}, \{1,3\}).$$

Note that in the first constraint, from $\Delta(1,2)$ we have that $X_{(1)}$ and $X_{(2)}$ cannot be both 1, even though they are compatible. However, the cluster $\{1,2\}$ with its variable $X_{(1,2)}$ can be set to 1, which allows cells 1 and 2 to be harvested.

This model formulation has the advantage that the number of constraints is relatively small, particularly in comparison to EARM-1. The reason for this is that EARM-2 relies on the number of nodes and weak adjacency arcs in G. It also has the added advantage that it dominates EARM-1 in that all Constraints (3) are dominated by Constraints (3a). However, this formulation, based on the work of Martins et al. (2000), may be further strengthened.

3.3. A Strengthened Formulation

Define graph $G(\Lambda, \Gamma)$ in which each node in Λ corresponds to a feasible cluster, and in which $(S, T) \in \Gamma$ if and only if S and T are noncompatible clusters in Λ. We henceforth call this graph the *compatibility graph*.

The EARM may be interpreted as an instance of node packing (or the independent set problem) in graph $G(\Lambda, \Gamma)$ and hence suggests that the use of many well-known facet defining constraints (Nemhauser and Sigismondi 1992) might be helpful. One approach is to use maximal cliques in graph $G(\Lambda, \Gamma)$. Cliques are sets of nodes in which all nodes are connected by arcs to each other. A clique is maximal if it is not contained in any other clique. This suggests yet another problem formulation.

Extended Area Restriction Model-3 (EARM-3)

Maximize $\sum_{(S,t)} c_{S,t} x_{S,t}$,

subject to (1), (2) and

$$\sum_{S \in K} x_{S,t} \leqslant 1 \quad \text{for each time period } t, \quad \text{for each maximal clique } K \text{ in } G(\Lambda, \Gamma). \tag{3b}$$

The maximal clique Constraints (3b) are known to be very strong for node-packing problems (Fulkerson 1971, Padberg 1973). Furthermore, it is easy to see that these constraints dominate Constraints (3a) in that each set $\Lambda(u, v)$ in (3a) corresponds to a set of mutually noncompatible clusters that define a clique that may or may not be maximal. Therefore, this formulation dominates EARM-2 as Constraints (3a) define cliques that are not necessarily maximal in $G(\Lambda, \Gamma)$.

From node-packing research we know that this formulation may in turn be strengthened by odd cycle constraints (Padberg 1973), web-antiweb constraints (Barahona and Mahjoub 1994), K4 reduction constraints (Giles and Trotter 1979, Barahona and Mahjoub 1989), and others typically used for node packing.

Although we have substantially enhanced the formulation of the EARM, there remain several problems with attempting to solve EARM-3. First, due to the large number of nodes and arcs in graph $G(\Lambda, \Gamma)$, the number of cliques (and other similar constraints) is extremely large, leading to unmanageable problems. Also, cutting plane approaches are difficult to implement in that solving the separation for this problem would be very difficult considering its size. We could reduce the size of the problem because it is not necessary to add all constraints to ensure a strengthened formulation. However, a problem would arise in determining which constraints to use. So, while EARM-3 represents a very tight formulation of this problem in terms of cliques, it is not realistic to think that we can actually structure it in practice.

4. Constraint Projection

So far the most manageable formulation presented for the EARM is EARM-2. However, this formulation needs further strengthening if it is to be successful in solving medium-sized problem instances. In this section, we present a method for generating strong valid inequalities that are stronger than those in EARM-2 and move toward the tight formulation of EARM-3.

4.1. Cluster Packing

For the sake of simplicity, assume that only one time period exists in our planning problem. This reduces EARM-1 to the following cluster-packing problem.

Cluster Packing Problem (CPP)

$$\text{Maximize } \sum_S c_S x_S,$$

subject to

$$x_S + x_{S'} \leqslant 1 \quad \text{for each pair } S,\ S' \text{ of noncompatible clusters},$$

$$x_S \in \{0, 1\} \quad \text{for each cluster } S \in \Lambda.$$

The objective function and constraints of CPP are identical to those of EARM-1, except that we are not considering multiple time periods. Using the CPP, we can derive projected constraints that will ultimately be applicable for the EARM.

4.2. Projected Constraints

Constraint projection is a method for generating strong inequalities that are valid for CPP. This methodology for generating constraints can in turn be complemented with additional constraints derived for classic node-packing problems, such as clique or odd cycle inequalities, derived from the node-packing problem in $G(\Lambda, \Gamma)$.

Let $\mathrm{CP}(G)$ be the convex hull of all vectors $x \in \{0, 1\}^{|\Lambda|}$ that satisfy

$$x_S + x_T \leqslant 1 \quad \text{for all } (S, T) \text{ in } \Gamma.$$

These 0–1 vectors correspond to all feasible solutions of cluster packing over $G(V, E)$ (in $G(\Lambda, \Gamma)$). Let $\mathrm{NP}(G)$ be the convex hull of all vectors $y \in \{0, 1\}^{|V|}$ that satisfy

$$y_u + y_v \leqslant 1 \quad \text{for all } (u, v) \text{ in } E.$$

These 0–1 vectors correspond to all feasible solutions of node packing in $G(V, E)$.

For a polyhedron P, the inequality $ax \leqslant \alpha$ is *valid* if it is satisfied by every element of P. To use LP techniques, we need valid inequalities. The inequalities that define facets are the "strongest" valid inequalities that one can use (see Nemhauser and Wolsey 1988 for a more complete discussion on this subject).

A valid inequality for $\mathrm{NP}(G)$ can be transformed into a valid inequality for $\mathrm{CP}(G)$ as follows. For $I \subseteq V$, define $\Lambda(I) = \{S \in \Lambda\colon S \cap I \neq \varnothing\}$.

Constraint Projection Lemma. *If*

$$\sum_{i \in I} a_i y_i \leqslant \alpha \tag{5}$$

is valid for $\mathrm{NP}(G)$, *then*

$$\sum_{S \in \Lambda(I)} b_S x_S \leqslant \alpha \tag{6}$$

is valid for $\mathrm{CP}(G)$, *where*

$$b_S = \underset{y_i}{\text{maximum}} \sum_{i \in S \cap I} a_i y_i, \tag{7}$$

subject to

$$y_i + y_j \leqslant 1 \ \textit{for each pair of adjacent cells } i, j \textit{ in } S \cap I, \tag{8}$$

$$y_i \in \{0, 1\} \quad \textit{for each } i \in S \cap I. \tag{9}$$

Proof. Consider an instance of CPP and a valid inequality (5). For each feasible cluster S, define y^S as any vector that solves (7)–(9).

Let x be a 0–1 vector in $\mathrm{CP}(G)$. Define $w \in \{0, 1\}^V$ such that

$$w_i = \begin{cases} y_i^S & \text{if } i \in S \cap I \text{ and } x_S = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Note that w is well defined. In fact, if S_1 and S_2 are such that $x_{S_1} = x_{S_2} = 1$, then they are compatible clusters, and as such they are disjoint.

Note also that $w \in \mathrm{NP}(G)$. Because of (8), no two adjacent cells i, j in a cluster will be such that $y_i = y_j = 1$, and because x is in $\mathrm{CP}(G)$, we know that if S_1 and S_2 are such that $x_{S1} = x_{S2} = 1$, then for $i \in S_1$ and $j \in S_2$, they cannot be adjacent due to the compatibility of S_1 and S_2. Thus, for x we have that

$$\sum_{S \in \Lambda(I)} b_S x_S = \sum_{i \in I} a_i w_i \leqslant \alpha,$$

with which we conclude that Constraint (6) is valid.

4.3. Some Important Projected Constraints

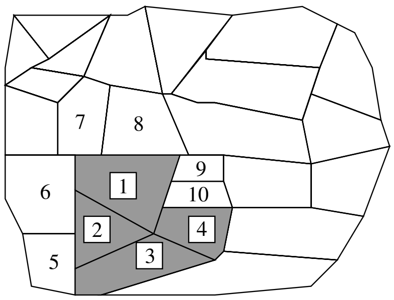
4.3.1. Projected Clique Constraints. Consider the highlighted cells $K = \{1, 2, 3, 4\}$ in graph $G(V, E)$ corresponding to the forest partition shown in Figure 3. These cells are all mutually adjacent to each other and thus form a clique.

If we were solving the node-packing problem in graph $G(V, E)$, we could add the valid clique constraint

$$y_1 + y_2 + y_3 + y_4 \leqslant 1.$$

This can be projected into a valid constraint for CPP. Consider the set of all clusters that intersect cells $\{1, 2, 3, 4\}$. For the purpose of illustration, we will suppose that there are only five such clusters (the set $\{S, T, U, W, Z\}$), though

Figure 3. A clique in a forest region.



generally there may be hundreds of these clusters even for a relatively small problem application. These clusters are depicted in Figure 4.

These clusters are all mutually noncompatible with each other. That is, it is impossible to choose two clusters in set $\{S, T, U, W, Z\}$ that are not adjacent or overlapping. Thus, they form a clique in $G(\Lambda, \Gamma)$ and define the following valid projected constraint:

$$X_S + X_T + X_U + X_W + X_Z \leqslant 1.$$

Note that by the constraint projection lemma, the coefficients of all variables in a projected clique constraint will always be one. This follows from the fact that for any cluster C intersecting clique K, the maximum size node packing in $C \cap K$ is of value one.

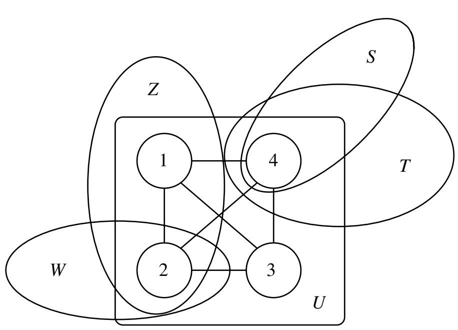
Although the clique $\{1, 2, 3, 4\}$ in $G(V, E)$ may be maximal, this is not necessarily the case for the projected clique. For instance, we have that the cluster defined by nodes $\{5, 6, 7, 8, 9, 10\}$ does not intersect clique $\{1, 2, 3, 4\}$, but is noncompatible with $S, T, U, W,$ and Z (as well as with any other cluster that intersects $\{1, 2, 3, 4\}$). If we define cluster $\{5, 6, 7, 8, 9, 10\}$ as R, the above constraint can be strengthened as

$$X_R + X_S + X_T + X_U + X_W + X_Z \leqslant 1.$$

In general, to obtain facets from a projected clique constraint associated with clique K in $G(V, E)$, it is a matter of following a simple procedure:

Step 1. Define set $\text{exterior}(\Lambda, K) = \{S \in \Lambda \backslash \Lambda(K)$: S is adjacent to all nodes in $K\}$.

Figure 4. Graph representation of a clique and its intersecting clusters.



Step 2. Choose $S_1 \in \text{exterior}(\Lambda, K)$; define $n = 1$; $\Lambda_1 = \Lambda(K)$.

Step 3. Let $\text{exterior}(\Lambda, K) = \text{exterior}(\Lambda, K) \backslash \{S_n\}$.

Step 4. Define $\Lambda_{n+1}(K) = \Lambda_n(K) \cup \{S_n\}$.

Step 5. Choose S_{n+1} in $\text{exterior}(\Lambda, K)$ such that S_{n+1} is noncompatible with all clusters in $\Lambda_{n+1}(K)$.

Step 6. If no such cluster exists, STOP. Else, $n = n + 1$. Goto 3.

This simple algorithm is easy and fast to implement and always identifies a maximal clique. Given that clusters may be selected in many different ways in Steps 2 and 4, this algorithm defines a tree of possible iterations for which each terminal branch defines a maximal clique dominating $\Lambda(K)$. We found that in the examples we have solved, projected cliques from $\Lambda(K)$ were only sometimes maximal. However, when not maximal, they generally became so after only two or three iterations of the lifting algorithm. In our case, lifting these projected cliques did not prove much of an advantage in strengthening the formulation. This, however, might not be the case for all graphs $G(V, E)$.

This clique family of constraints is also derived by Martins et al. (2000) in a similar, but more complex adjacency problem. In addition to adjacency, the authors impose constraints for preserving blocks of mature (old growth) stands.

4.4. Constraint Projection for the EARM

As we have argued, formulation EARM-2 is not strong enough for solving medium-size instances of the EARM in an acceptable time, and formulation EARM-3 is unrealistic in that it has too many constraints. From our previous discussion, a very tight formulation can be structured for moderate-size problems as a compromise of the two.

To apply the constraint projection method (§4.2) to the EARM, it is important to note that it is possible to interpret EARM as series of cluster-packing problems—one for each period. Thus, to use constraint projection one must decide which constraints to project and which period of time to project them in.

How one should formulate the problem depends on the structure of the weak adjacency graph $G(V, E)$, because that will give a clue to which constraints are worth projecting. We found in most of our test cases that formulating the EARM by projecting all the clique inequalities for each time period proved best because the number of clique inequalities was manageable. By doing this, we obtain the following problem formulation.

Extended Area Restriction Model (EARM-4)

$$\text{Maximize} \sum_{S, t} c_{S, t} x_{S, t},$$

subject to (1), (2), and

$$\sum_{(S, t):\, S \in \Lambda(K)} x_{S, t} \leqslant 1 \quad \text{for each maximal clique } K \text{ in } G(V, E), \text{ for each } t. \tag{3c}$$

Table 1. LP relaxation results for single period applications.

Instance	EARM-2::LP Obj. value	EARM-2::LP Sol. time	EARM-4::LP Obj. value	EARM-4::LP Sol. time
Butter Creek	10,362.73	7.89	10,114.15	1.03
El Dorado	1,702,266.69	5.03	1,692,205.00	1.19

Recall that the objective maximizes the net benefit over the time horizon T. Constraint (1) states that each basic cell may be harvested at most once during the time horizon. Constraint (2) imposes integrality. Constraint (3c) states that from any maximal clique composed of noncompatible clusters, only one cluster can be harvested in each given period. These inequalities guarantee that no pair of compatible clusters will be simultaneously harvested because for every pair of noncompatible clusters there exists some weak arc in E that joins them, which in turn is dominated by some clique K in $G(V, E)$ intersecting both clusters.

Figure 1 will be used to show the differences between the various models discussed thus far. Assume that all cells are 10 hectares in size and that a 49 hectares maximum is imposed. Using EARM-1, cells $(3, 9, 5, 7)$ make up one of the many feasible clusters. We would need to define all feasible clusters and then write one constraint for each pair of noncompatible clusters. As an example, consider the following for one time period:

$$X_{(3,9,5,7)} + X_{(10,9,8,11)} \leqslant 1.$$

Using EARM-2, we would consider all arcs (u, v) in G. For example, for arc $(2, 9)$, all clusters that contain cells 2, 9, or both would appear in Constraint (8):

$$X_{(3,9,5,7)} + X_{(10,9,8,11)} + X_{(2,3,9,10)} + X_{(1,2,10)} + \cdots \leqslant 1.$$

In contrast, only maximal cliques in $G(\Lambda, \Gamma)$ would be considered in EARM-3. This case cannot be represented using only Figure 1; we would also need to represent each feasible cluster with a node in the compatibility graph. These nodes are joined by arcs if the clusters are noncompatible. For example, let cluster $\{3, 9, 5, 7\}$ be node a, cluster $\{1, 3, 4, 5\}$ be node b, cluster $\{2, 10, 11\}$ be node c, and cluster $\{1, 3, 9\}$ be node d. Because all these clusters are noncompatible, the nodes are all connected and thus form a clique. However, there are other clusters that can be added, so this clique is not maximal.

The associated graphic representation for EARM-4 for this case has already been detailed in Figure 4.

5. Forest Scheduling Application

5.1. Implementation

A maximum area restriction of 49 hectares was imposed in all modeling applications. Two forest planning problems are first examined. These instances correspond to the Butter Creek Forest and a region in the El Dorado National Forest, both in northern California. Butter Creek is a 351-unit problem (each unit averaging 10 hectares in size) defined for one period, and El Dorado is a 1,351-unit problem (each unit averaging 15.5 hectares in size) with seven time periods (note that we examined runs for the El Dorado application with fewer periods as well). To complement the analysis we also carried out a study using randomly generated problem applications. The randomly generated instances were constructed as square regions subdivided into equally-sized cells of 9.7 hectares or 16.2 hectares. The variation in sizes for the random-generated problem instances enables resulting graph structure to be examined. Each cell was randomly assigned a timber volume between 300 and 600 cubic meters per hectares. A 5% growth rate was assumed between periods. Each cubic meter of timber was priced at \$10 US for the first period, and subsequently a 7% interest rate per period was applied.

CPLEX version 7.1 was used to solve the reported planning applications. The test problems were solved on a Pentium III 700 mhz processor PC with 1 GB of RAM. Solution times are reported in CPU seconds.

For multiple time period problem instances, volume conservation constraints were introduced to assure minimum yields and regular production. Let $v_{S,t}$ correspond to the volume of timber obtained if cluster S is harvested in time period t. The following constraints on producing an approximate even flow of timber in each period impose that for each period the volume of timber harvested is within 15% of that harvested in the previous period:

$$0.85 \sum_{S\in\Lambda} v_{S,t-1} x_{S,t-1} \leqslant \sum_{S\in\Lambda} v_{S,t} x_{S,t} \leqslant 1.15 \sum_{S\in\Lambda} v_{S,t-1} x_{S,t-1} \quad \forall t \geqslant 2. \qquad (10)$$

5.2. Results

5.2.1. EARM. In Tables 1–6, results are presented for applications involving both random and real problems. Tables 1 and 2 contrast the single time period EARM-2 and

Table 2. Single period application results.

Instance	EARM-2 Obj. value	EARM-2 Sol. time	EARM-4 Obj. value	EARM-4 Sol. time
Butter Creek	10,072.50 (0.98% gap)	14,400.00	10,104.19	1.44
El Dorado	1,692,055.00	230.59	1,692,100	4.57

Table 3. Multiple period instances of the El Dorado problem.

Instance	Time periods	EARM-4::LP Obj. value	EARM-4::LP Sol. time	EARM-4 Obj. value	EARM-4 Sol. time
El Dorado	3	3,154,887.40	332.77	3,082,505 (2.26% gap)	14,400.00
El Dorado	5	4,041,891.39	201.93	4,006,110 (0.08% gap)	14,400.00
El Dorado	7	4,920,589.16	394.32	4,837,950 (1.67% gap)	28,800.00

Table 4. LP relaxation results for the single period random applications with 9.7 hectares per cell.

Instance	EARM-2::LP Obj. value	EARM-2::LP Sol. time	EARM-4::LP Obj. value	EARM-4::LP Sol. time
8×8 grid	38,906,477.28	4.41	34,242,116.40	0.47
12×12 grid	82,256,166.96	32.45	69,741,865.44	2.63
16×16 grid	142,743,792.00	329.33	118,331,663.76	14.92
23×23 grid	291,096,000.00	1,060.25	238,113,522.72	116.40

Table 5. Single period results for the random applications with 9.7 hectares per cell.

Instance	EARM-2 Obj. value	EARM-2 Sol. time	EARM-4 Obj. value	EARM-4 Sol. time
8×8 grid	34,242,116.40 (1.19% gap)	14,400.00	34,242,116.40	0.47
12×12 grid	66,830,259.36 (16.39% gap)	14,400.00	69,094,914.24	35.88
16×16 grid	108,207,570.24 (24.03% gap)	14,400.00	117,306,168.72	711.97
23×23 grid	202,427,829.36 (30.26% gap)	14,400.00	236,486,352.00 (0.17% gap)	14,400.00

EARM-4 formulations of the Butter Creek and El Dorado applications. In both cases, the EARM-4 formulation produces a tighter bound and a faster solution time in the LP relaxation. The difference in the time it takes to obtain an optimal integer solution is substantial. For Butter Creek, EARM-2 needed four hours to obtain a 0.98% gap, while the proposed EARM-4 obtained an optimal solution in 1.44 seconds and is illustrated in Figure 5. For El Dorado, the time to reach an optimal solution is reduced from 230.59 to 4.57 seconds. Table 3 shows results for the El Dorado forest application with three, five, and seven periods. Once multiple periods are introduced, solving the problems becomes far more difficult. For EARM-4, a stopping rule of four hours solution time was used if the gap was below 3%. Otherwise, solution time was extended to eight hours. Given the poor performance of the EARM-2 formulation, we did not attempt to solve any of the multiple period applications.

The results show that EARM-4 can be solved for a relatively large forest, 1,351 spatial units (cells) and up to seven periods, although this approached the capabilities of the utilized computing equipment. To further confirm the advantage of the proposed approach, applications were carried out using the described randomly generated problems.

Table 4 shows LP relaxation results for single-period instances of the 8×8, 12×12, 16×16, and 23×23 grids. The EARM-4 produces significantly tighter bounds and shorter solution times. Table 5 contrasts the time taken to obtain integer solutions. In four hours of computing time

Figure 5. Optimal EARM-4 solution for the Butter Creek application.

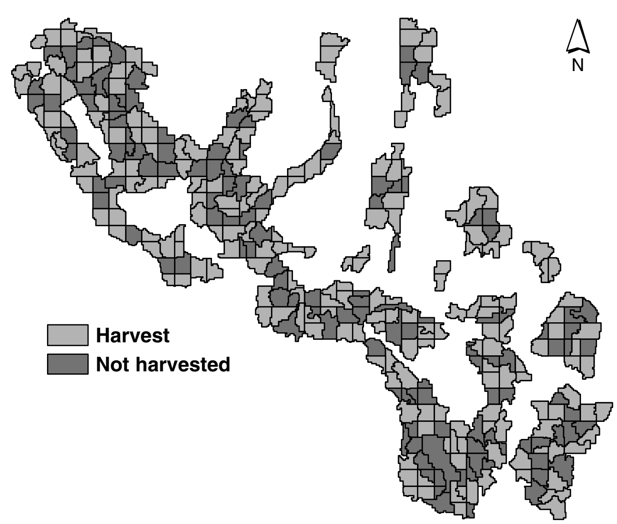


Table 6a. Multiple period results for random applications with 9.7 hectares per cell.

Instance	Time periods	EARM-4::LP Sol. time	EARM-4::LP Obj. value	EARM-4 Sol. time	EARM-4 Obj. value
8×8 grid	3	11.08	66,140,370.04	14,400 1st 1%: 159.13	66,130,291.90 (0.02% gap)
8×8 grid	5	14.61	76,025,038.08	14,400 1st 1%: 3,979.87	75,577,172.42 (0.59% gap)
8×8 grid	7	760.46	88,340,830.55	28,800	84,004,225.36 (4.91% gap)
12×12 grid	3	150.33	143,370,968.87	14,400 1st 1%: 2,268.10	143,299,974.97 (0.05% gap)
12×12 grid	5	77.37	164,797,737.92	14,400 1st 1%: 10,606.71	164,593,468.66 (0.12% gap)
12×12 grid	7	428.51	191,494,399.85	14,400	186,723,087.41 (2.49% gap)
16×16 grid	3	997.46	2.51E + 08	28,800 1st 1%: 22,377.81	2.50E + 08 (0.43% gap)
16×16 grid	5	166.52	2.89E + 08	14,400 1st 1%: 9,907.44	2.88E + 08 (0.22% gap)
16×16 grid	7	18,007.58	335,640,000	28,800	**
23×23 grid	3	2,989.6	5.18E + 08	28,800	502,328,293.3 (3.09% gap)
23×23 grid	5	601.26	595,776,000.00	14,400 1st 1%: 12,849.75	594,942,346.32 (0.14% gap)
23×23 grid	7	*	*	*	*

Note. The time at which the first solution at 1% of the optimum is obtained is indicated by 1st 1% when applicable.

* Problem too large, not enough RAM memory to formulate.

** Problem too large, no feasible solutions found.

large optimality gaps remain for the EARM-2 approach. The proposed EARM-4 formulation readily solves the 8×8, 12×12, and 16×16 problem instances. For the 23×23 instance, with 529 cells, a small gap of 0.17% remains after four hours.

The application results illustrate how much more constrained EARM-4 is than EARM-2 and how much faster it may be solved by use of branch and bound. EARM-4 is a tighter formulation capable of solving larger and more complex problem instances than EARM-2.

Table 6b. Multiple period results for random applications with 16.2 hectares per cell.

Instance	Time periods	EARM-4::LP Sol. time	EARM-4::LP Obj. value	EARM-4 Sol. time	EARM-4 Obj. value
8×8 grid	3	0.69	110,233,950.07	14,400 1st 1%: 114.61	110,226,933.78 (0.01% gap)
8×8 grid	5	0.71	126,708,396.80	14,400 1st 1%: 1,190.29	126,595,316.97 (0.09% gap)
8×8 grid	7	2.96	147,234,717.59	14,400	142,124,737.78 (3.47% gap)
12×12 grid	3	7.12	238,951,614.78	14,400 1st 1%: 1,734.55	238,890,814.41 (0.03% gap)
12×12 grid	5	3.14	274,662,896.54	14,400 1st 1%: 865.60	274,417,236.16 (0.09% gap)
12×12 grid	7	2.95	319,157,333.08	14,400	315,069,200.21 (1.28% gap)
16×16 grid	3	28.82	4.19E + 08	14,400	412,335,756.24 (1.55% gap)
16×16 grid	5	9.05	4.81E + 08	14,400 1st 1%: 399.40	480,212,290.6 (0.25% gap)
16×16 grid	7	6.32	5.59E + 08	14,400 1st 1%: 2,424.71	554,982,507.2 (0.49% gap)
23×23 grid	3	192.63	8.64E + 08	28,800	806,503,666.03 (6.64% gap)
23×23 grid	5	20.09	2.48E + 07	14,400 1st 1%: 3,542.51	24,670,778.05 (0.61% gap)
23×23 grid	7	857.65	2.88E + 07	14,400	28,249,058.36 (2.07% gap)

Multiple time period extensions are reported in Tables 6a and 6b. In addition to the solution time, the time at which the first feasible solution within 1% of the optimum is found is also provided. One observation that can be made from these results is that the random problem instances required greater computing effort than the actual forest planning applications. The regularity of the grid undoubtedly contributes to the existence of more alternatives in the feasible grouping of harvested cells. A second observation is that there are two cases in which a feasible solution was not able to be identified in Table 6a (16×16 grid with seven time periods and 23×23 grid with seven time periods). A final observation is that feasible solutions were found for the larger celled problems reported in Table 6b. Thus, in comparison with Table 6a the increased cell sizes reduce the number of potential alternative solutions (feasible clusters) and can now be solved, at least for feasible solutions with reasonable optimality gaps.

6. Conclusions

In this paper, we have presented a new way to model harvest scheduling subject to maximum area constraints by introducing the extended area restriction model (EARM). This new way to describe the problem incorporates the notions of strong and weak adjacency to provide solutions with better spatial structure than the traditional area restriction model (ARM). We have presented a way to formulate the EARM that is also applicable to the classic ARM. We discuss several ways to strengthen this formulation and show that there is an ideal formulation for this problem that is unrealistic because its size is unmanageable.

By further studying the formulations presented, it is seen that underlying the EARM is a very difficult combinatorial problem we call the cluster-packing problem (CPP). We see that this problem is strongly related to the classic node-packing problem (NPP) and show how to take advantage of this relationship by proving one can project strong valid inequalities from the NPP to the CPP. By use of this projection theorem, we are able to use these new projected valid inequalities in the EARM and obtain a very strong formulation that is similar to the ideal formulation.

Application results confirm that this formulation is very tight in that the optimum linear relaxation solutions are very close to the optimum integer solutions for the problems tested. This tightness enables branch and bound to solve medium-sized EARM instances reasonably fast to within 1% of the optimal solution. These findings suggest that the EARM may be utilized for solving medium-sized, multiperiod problem instances with volume constraints.

The basic problem consists of one in which we start with a large number of cells and proceed to form them into contiguous blocks in such a way that there is some separation between these blocks. Other problems in forestry that might have this type of characterization so that this algorithmic approach (with modifications) might be useful include (1) need to preserve blocks of mature trees of minimal size, as in Martins et al. (1999, 2000); (2) cases where there is a need to have corridors of mature trees between clearings resulting from harvesting, where wildlife feeds and the role of the corridors is to provide animals with protection to move from one clearing to another (Sessions and Sessions 1991); and (3) cases where the perimeter between clearings and blocks of mature trees play a role in supporting certain animal species (Martins et al. 1999). Looking at another application area, radio transmission studies might consider a problem of forming blocks of cells (where each cell might be a city block) associated with service zones and separation of blocks/zones is needed to avoid interference.

Projected clique constraints were implemented in this paper to solve a forest scheduling problem. For this problem or others described in this section, additional valid inequalities, such as odd cycles, properly implemented, might be of algorithmic value. Further research is needed to determine efficient algorithms to implemented additional valid inequalities. As the number of periods in the problems increases, our approach encountered difficulties. Another line of future research lies in solving problems with significantly larger number of cells and more planning periods.

Acknowledgments

Partial funding for this research was received from FONDECYT under grant numbers 1000959 and 1020317. Research support of the second author was provided by the National Science Foundation (Geography and Regional Science Program and the Decision, Risk, and Management Science Program) under grant BCS-0114362. The authors thank Klaus Barber of the USDA Forest Service for making the utilized forests available for analysis. This paper has benefited also from the comments provided by three referees and the associate editor, for which the authors are grateful.

References

American Forest and Paper Association. 2001. Sustainable forestry initiative standard. http://www.afandpa.org/.

Barahona, F., A. R. Mahjoub. 1989. Composition of graphs and polyhedra III: Graphs with No W_4 Minor. No. 89565-OR, Inst. für Ökonomie und OR. Rheinische Friedrich Wilhelms Universität, Bonn, Germany.

Barahona, F., A. R. Mahjoub. 1994. Compositions of graphs and polyhedra II: Stable sets. *SIAM J. Discrete Math.* **7** 359–371.

Barahona, F., R. Epstein, A. Weintraub. 1992. Habitat dispersion in forest planning and the stable set problem. *Oper. Res.* **40** S14–S20.

Barrett, T. M. 1997. Voronoi tessellation methods to delineate harvest units for spatial forest planning. *Canadian J. Forest Res.* **27** 903–910.

Barrett, T. M., J. K. Gilless. 2000. Even-aged restrictions with sub-graph adjacency. *Ann. Oper. Res.* **95** 159–175.

Barrett, T. M., J. K. Gilles, L. Davis. 1998. Economic and fragmentation effects of clearcut restrictions. *Forest Sci.* **44** 569–577.

Boston, K., P. Bettinger. 2001. The economic impact of green-up constraints in the southeastern United States. *Forest Ecology and Management* **145** 191–202.

Boston, K., P. Bettinger. 2002. Combining tabu search and genetic algorithm heuristic techniques to solve spatial harvest scheduling problems. *Forest Sci.* **48** 35–58.

Clark, M. M., R. D. Meller, T. P. McDonald. 2000. A three-stage heuristic for harvest scheduling with access road network development. *Forest Sci.* **46** 204–218.

Crowe, K., J. Nelson, M. Boyland. 2003. Solving the area-restricted harvest-scheduling model using the branch and bound algorithm. *Canadian J. Forest Res.* **33** 1804–1814.

Daust, D. K., J. D. Nelson. 1993. Spatial reduction factors for strata-based harvest schedules. *Forest Sci.* **39** 152–165.

Fulkerson, D. 1971. Blocking and anti-blocking pairs of polyhedra. *Math. Programming* **1** 168–194.

Giles, R., L. Trotter Jr. 1979. On stable set polyhedra for $K_{1,3}$-free graphs. *J. Combin. Theory* **31** 313–326.

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

Hokans, R. H. 1983. Evaluating spatial feasibility of harvest schedules with simulated stand-selection decisions. *J. Forestry* **81** 601–603, 613.

Jones, J. G., B. Meneghin, M. Kirby. 1991. Formulating adjacency constraints in linear optimization models for scheduling projects in tactical planning. *Forest Sci.* **37** 1283–1297.

Lockwood, C., T. Moore. 1993. Harvest scheduling with spatial constraints: A simulated annealing approach. *Canadian J. Forest Res.* **23** 468–478.

Martins, I., M. Constantino, J. Borges. 1999. Forest management models with spatial structure constraints. Working paper 2/99, Centro de Investigação Operacional, Universidade de Lisboa, Lisbon, Portugal.

Martins, I., M. Constantino, J. Borges. 2000. A column generation approach for solving a non-temporal forest harvest model with spatial structure constraints. Technical report Centro de Investigação Operacional, Universidade de Lisboa, Lisbon, Portugal.

McDill, M. E., S. A. Rabin, J. Braze. 2002. Harvest scheduling with area-based adjacency constraints. *Forest Sci.* **48** 631–642.

Murray, A. T. 1999. Spatial restrictions in harvest scheduling. *Forest Sci.* **45** 1–8.

Murray, A. T., R. L. Church. 1995. Heuristic solution approaches to operational forest planning problems. *OR Spektrum* **17** 193–203.

Murray, A. T., R. L. Church. 1996. Analyzing cliques for imposing adjacency restrictions in forest models. *Forest Sci.* **42** 166–175.

Murray, A. T., R. L. Church. 1997. Facets for node packing. *Eur. J. Oper. Res.* **101** 598–608.

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

Nemhauser, G. L., G. Sigismondi. 1992. A strong cutting plane/branch-and-bound algorithm for node packing. *J. Oper. Res.* **43** 443–457.

Nemhauser, G. L., L. A. Wolsey. 1988. *Integer and Combinatorial Optimization*. John Wiley and Sons, New York.

O'Hare, A., B. H. Faaland, B. B. Bare. 1989. Spatially constrained timber harvest scheduling. *Canadian J. Forest Res.* **19** 715–724.

Padberg, M. W. 1973. On the facial structure of set packing polyhedra. *Math. Programming* **5** 199–215.

Richards, E. W., E. A. Gunn. 2000. A model and tabu search method to optimize stand harvest and road construction schedules. *Forest Sci.* **46** 188–203.

Sessions, J., J. B. Sessions. 1991. Tactical harvest planning. *Proc. 1991 Soc. American Foresters National Conventions*, August 4–7. SAF Publication 91-05, Bethesda, MD, 362–368.

Thompson, E. F., B. G. Halterman, T. S. Lyon, R. L. Miller. 1973. Integrating timber and wildlife management planning. *Forestry Chronicle* **47** 247–250.