Forest Harvesting and Minimum Cuts: A New Approach to Handling Spatial Constraints

Dorit S. Hochbaum *
Department of Industrial Engineering and Operations Research and
Walter A. Haas School of Business,
University of California, Berkeley
email: dorit@hochbaum.berkeley.edu

Anu Pathria †
Department of Industrial Engineering and Operations Research,
University of California, Berkeley
email: pathria@cimsim.berkeley.edu

December 10, 1996

*Research supported in part by ONR contract N00014-91-J-1241.

†Author supported in part by an *NSERC '67* scholarship provided by the Natural Sciences and Engineering Research Council of Canada, and in part by ONR contract N00014-91-J-1241.

Abstract. In this paper we discuss two forest harvesting optimization problems that arise in forest management. We consider one-stage decision problems in which the forest to be managed is partitioned into cells and the decision is to be made as to which cells to harvest. There are competing concerns to consider in assessing the value of a particular harvesting decision: the first has to do with the value of the timber harvested, and the second has to do with assessing the relative values of the resulting spatial layout from a wildlife habitat point of view. The two problems that we consider are similar, but assess the relative merits of resulting spatial layouts somewhat differently and therefore have different objective functions to be optimized. The first problem we consider assigns benefits for harvesting cells and penalties for harvesting *adjacent* cells; the second problem considered assigns benefits for harvesting cells as well as benefits for creating borders that separate harvested and unharvested cells.

We show that both forest harvesting problems can be interpreted as instances of the *Generalized Independent Set* graph optimization problem, which we introduce in this paper. *Generalized Independent Set* is a generalization of the well-known *Independent Set* graph optimization problem (a set of nodes in a graph is said to be *independent* if no two nodes in the set are adjacent). When the underlying graph is bipartite, we derive an efficient solution method for *Generalized Independent Set* via network flow methods. If the cell structure is grid-like in the forest harvesting problems, then an instance of either forestry problem can be reduced to an instance of *Generalized Independent Set* on a bipartite graph. For such instances we developed efficient algorithms for determining an optimal harvesting strategy. The running time is $O(n^2 \log n)$, where n is the number of cells in the land area to be spatially managed.

Additional Keywords: forest harvesting, spatial allocation, minimum cut, independent set

Introduction

Consider the land management decision problem of determining which parts of an old growth forest to harvest, and which parts to maintain. Multi-stage harvest planning strategies have been studied in the past, with the value of timber harvested being the primary concern. We wish to also incorporate the effect of particular spatial allocations of old growth versus cutover area to the wildlife in the area as a consideration. The introduction of the non-timber concern of wildlife habitat has been raised by Hof and Joyce (1992), in a single-stage setting. The idea is to weigh the relative benefits and costs of any harvesting strategy in terms of both the timber harvested and what the resulting forest spatial layout means to the wildlife in the area.

As stated in Hof and Joyce (1992), *"wildlife habitat requirements include factors related to food, shelter, and hiding needs"*. We will consider two single-stage problems in this area, and develop strategies for making optimal land allocation decisions for each. We imagine that the land to be managed is old growth forest. The forest is partitioned into contiguous cells, and a decision needs to be made as to which cells to harvest and which not to harvest. The two forest harvesting problems that we consider, and that are described later in this section, measure the non-timber costs/benefits of different spatial layouts somewhat differently.

We will represent both forest harvesting models as graph theoretic optimization problems. In turn, by reducing each of these graph theoretic optimization problems to a newly introduced combinatorial optimization problem, referred to as *Generalized Independent Set*, we reduce the problem of determining optimal harvesting strategies to solving instances of *Generalized Independent Set.* The formal definition for *Generalized Independent Set* can be found later in the paper. Under the assumption that the land to be managed is partitioned into a grid-like cell structure (that is, the forest is imagined to be divided up into rectangular regions, or cells, by vertical and horizontal boundaries cutting all the way through the land area, as in Figure 1), we describe an efficient algorithm to solve both forest harvesting problems.

What do we mean by an "efficient algorithm"? In this paper, we will assume a very basic familiarity with the notation used in complexity analysis. Complexity analysis is typically used to describe the running time of an algorithm; an algorithm is a step-by-step procedure for solving a problem. We wish to determine, as a function of some parameters measuring the size of the input, the number of elementary mathematical operations required by the algorithm to solve an instance of the problem being considered. Typically, the elementary operations include addition, subtraction, division, and multiplication, as well as comparison. Thus, complexity analysis is used to measure the efficiency of an algorithm in solving a problem.

We give a brief informal presentation of complexity analysis here. More detailed and formal development of this material is available elsewhere (Baase 1988, Garey and Johnson 1979, Papadimitiou and Steiglitz 1982). We wish to measure the asymptotic growth in running time of the algorithm as a function of the input size. Our measure will be in terms of a *worst-*

case analysis in that some instances of the problem may be solvable more quickly than others; however, the complexity (or, running time) is a guarantee for *all* instances of a given size. We introduce the "big *Oh*" notation: an algorithm is said to have running time $O(f(n))$ if there exist constants c and n_0 so that the number of elementary operations required to solve a problem instance of size n is no more that $cf(n)$ for all inputs of size $n \geq n_0$. Notice that when using the "big *Oh*" notation to describe a function, we do not care about lower order terms and we do not care about constant factors that may occur in the function; rather, we are simply interested in the rate of growth of the function.

Now, an algorithm is said to be *good* or *efficient* if the running time is bounded by a polynomial in the size of the input;[1] such an algorithm is often referred to as a *polynomial time algorithm.* Consider Table 1. Notice that the exponential function, 2^n, which is not bounded by a polynomial, grows much more rapidly than the others. We will develop efficient algorithms in this paper for the two forest harvesting decision problems that are considered when a grid-like cell structure is assumed.

We describe the two forest harvesting problems next.

Problem 1: Avoiding Adjacent Cutover Regions. In this problem, in addition to timber concerns, the issue of preserving suitable habitat for old-growth dependents is addressed. Old growth forest provides both protection and food for certain animals, such as deer. Hence, we wish to take avoid large exposed regions of cutover area in any land management decision. So, associated with each cell is a benefit H_i attained from harvesting the cell i; but, a cost of C_e is assessed if both cells sharing a border e are harvested. For example, looking at the cell-structure of Figure 1, if cell 7 is harvested then a cost will be incurred for harvesting any of cells 4, 8, or 10. The objective is to determine a harvesting strategy maximizing the difference between the benefits and the costs.

A version of this problem, in which adjacent cells were simply disallowed from both being harvested, was considered by Barahona, Weintraub and Epstein (1992).

Problem 2: Harvesting with Edge Effects. In this problem, in addition to timber concerns, we wish to consider the so-called "edge effect". Certain species, known as "edge dependents", have habitat requirements involving both cutover area and old growth. As such animals often cannot range very far, we wish to maintain a suitable mixture of both cutover area and old growth in close proximity to each other; that is, there is benefit associate with the amount of border between cutover area and old growth. For example, looking at the

[1] A polynomial in n is a function of the form $\sum_{i=0}^{k} c_i n^i$, where k is a non-negative integer and the c_i's are arbitrary constants. There is a well known class of problems, the so-called *NP-hard* problems, that are thought to be immune to having efficient algorithms being developed for them (the *Traveling Salesman Problem* is perhaps the most celebrated of problems in this class). See Garey and Johnson (1979) for a more detailed description of the class of NP-hard problems.

cell-structure of Figure 1, there will be an "edge effect" benefit associated with border between cells 2 and 5 only if exactly one of those two cells is harvested and one is not. So, associated with each cell is a benefit H_i from harvesting cell i, a benefit U_i for maintaining old growth in cell i, and a benefit B_e for harvesting exactly one of the two cells sharing the border e (B_e is designed to account for the "edge effect"). The objective is to determine a harvesting strategy maximizing total benefit.

This problem was introduced, in a somewhat simplified version, by Hof and Joyce (1992), There, the three types of benefits described above were all constant. The number of possible harvesting decisions for this (as well as for the first harvesting problem), is 2^n, where n is the number of cells. As we mentioned earlier, such exponential functions grow very rapidly as n grows; or, as stated in Hof and Joyce (1992), *"the number of spatial possibilities approaches google numbers."* The solution method they employed for determining an optimal spatial configuration for this problem, (and demonstrated on a 3×3 grid), was to model the problem as a non-linear integer program and to solve the formulation with the integrality restrictions relaxed. The difficulties associated with such an approach are two-fold: first, because of the non-linearity, the solution returned may be a local, rather than global, optima; second, the solution returned may not even be integer valued, and hence infeasible for the problem. A follow-up paper of Hof and Joyce (1993), offered a mixed linear integer programming formulation for the problem, but still no polynomial time solution algorithm.

The paper is laid out as follows. We first show how to describe the cell structure of the land that is to be managed via a graphical representation. The two forest harvesting problems are then formally defined as graph optimization problems. We next introduce a new graph optimization problem called the *Generalized Independent Set* problem, which is a generalization of the well-known *Independent Set* problem (e.g. Garey and Johnson 1979). We provide an efficient polynomial time algorithm that solves instances of *Generalized Independent Set* occurring on graphs that are *bipartite*. We then show that an instance of either forest harvesting problem can be reduced to an instance of *Generalized Independent Set*. Furthermore, when a grid-like cell structure is assumed, then the corresponding instance of *Generalized Independent Set* is on a bipartite graph; hence, an optimal management decision strategy can be efficiently found for such forest harvesting problems.

Formal Description of the Forest Harvesting Problems

In this section, we show how to describe the cell structure of the land on which the management decision is to be made in the forest harvesting problems as a graph. Thus, we begin by briefly introducing some graph theoretic notation. A graph $G = (V, E)$ is made up of a set of vertices

V and a set of edges E. An edge $e \in E$ designates a link between two vertices; more formally, an edge $e = \{v_i, v_j\}$ is given as an unordered pair of vertices where $v_i \in V$ and $v_j \in V$. We will only consider *simple* graphs in which there are no self-loops (i.e. $v_i \neq v_j$) and no multi-edges (i.e. at most a single edge connecting two vertices). For a given subset of vertices $S \subseteq V$, $(S, \bar{S})$ refers to the set edges with one endpoint in S and the other in $\bar{S} = V - S$. A graph may be weighted in that weights are assigned to vertices and/or edges.

Now, imagine the cell structure of the land to be managed. The cells are assumed to be contiguous and partition the forest area. We construct a graph to correspond to a particular cell structure. Each vertex in the graph corresponds to a cell, and there is an edge between two vertices if and only if the two corresponding cells are adjacent (note that two cells are said to be adjacent if they share a common border; thus, cells that meet at a point are not considered adjacent in this definition). See Figure 2 for the graph describing the given grid-like cell structure. Any weights associated with a cell are now associated with its corresponding vertex in the graph, and any weight associated with the border between two adjacent cells is now associated with the edge corresponding to this border. We can now formulate the two forest harvesting problems as graph optimization problems using the graphical description of the cell structure just given.

Consider first the harvesting problem in which there is the wildlife concern of accounting for the habitat of old growth dependents. Recall that there is a benefit H_v associated with harvesting cell v. However, there is a cost $C_{\{i,j\}}$ associated with harvesting both cells i and j, for cells i and j sharing a common border. The corresponding graph optimization problem is defined below. (See Figure 3(a) for the graph corresponding to the 4×3 grid-like cell structure with weights labeled.)

Problem Name: *Forest Harvesting: Cutover Adjacency*

Instance: *Given a graph $G = (V, E)$, a weight H_v associated with each vertex $v \in V$, and a cost C_e associated with each edge $e \in E$.*

Optimization Problem: *Select a subset of the vertices $S \subseteq V$ that maximizes the difference between the weight of the vertices in S and the penalty of those edges that have both endpoints in S; that is, the objective is to maximize the quantity, $S \subseteq V$ that maximizes the quantity,*

$$\sum_{v \in S} H_v \; - \sum_{e=\{i,j\} \in E : i,j \in S} C_e \; .$$

Consider now the forest harvesting problem in which there are two non-timber concerns: that of maintaining old growth forest, and that of providing a benefit to animals via areas where there is a mix of old growth forest and harvested land. Recall that there is a benefit

H_v associated with harvesting cell v. However, there is also a benefit U_v associated with *not* harvesting cell v; in addition, there is a benefit $B_{\{i,j\}}$ associated with harvesting exactly one of cells i or j, for cells i and j sharing a common border. The corresponding graph optimization problem is defined below. (See Figure 3(b) for the graph corresponding to the 4×3 grid-like cell structure with weights labeled.)

Problem Name: *Forest Harvesting: Edge Effects*

Instance: *Given a graph $G = (V, E)$, two weights H_v and U_v associated with each vertex $v \in V$, and a benefit B_e associated with each edge $e \in E$.*

Optimization Problem: *Select a subset of the vertices $S \subseteq V$ that maximizes overall benefit; that is, the objective is to maximize the quantity,*

$$\sum_{v \in S} H_v + \sum_{v \notin S} U_v + \sum_{e \in (S, \bar{S})} B_e \ .$$

We now turn to introducing a new graphical combinatorial optimization problem, *Generalized Independent Set*. As we have mentioned, both of our forest harvesting graph optimization problems can be reduced to instances of *Generalized Independent Set*; of particular relevance, when a grid-like cell structure is assumed, the corresponding instance of *Generalized Independent Set* will be solvable by an efficient algorithm that we develop in the next section.

Minimum Cuts and the Generalized Independent Set Problem

The Generalized Independent Set Problem

The *Independent Set* problem is a well-known NP-hard combinatorial optimization problem. It is a graph optimization problem in which, given a graph with vertex weights, the objective is to select a maximum weight set of vertices such that no two vertices selected are adjacent in the original graph. The problem remains NP-hard even when all vertices are assumed to have unit weight. A formal definition of *Independent Set* is given below.

Problem Name: *Independent Set*

Instance: *Given a graph $G = (V, E)$, and a weight W_v associated with each vertex $v \in V$.*

Optimization Problem: *Select a subset of the vertices $S \subseteq V$ of maximum weight, such that no edge $e \in E$ has both of its endpoints in S, where the weight of S is $\sum_{v \in S} W_v$.*

In the *Generalized Independent Set* problem, we relax the structural constraint that disallows for two adjacent vertices to both be selected in the set; rather, we allow for such selection, but at the cost of incurring a penalty. A formal definition of *Generalized Independent Set* is given below.

Problem Name: *Generalized Independent Set*

Instance: *Given a graph* $G = (V, E)$*, a weight* W_v *associated with each vertex* $v \in V$*, and a penalty* P_e *associated with each edge* $e \in E$.

Optimization Problem: *Select a subset of the vertices* $S \subseteq V$ *that maximizes the difference between the weight of the vertices in* S *and the penalty of those edges that have both endpoints in* S*; that is, the objective is to maximize the quantity,*

$$\sum_{v \in S} W_v \; - \sum_{e=\{i,j\} \in E : i,j \in S} P_e \; .$$

Note that in *Generalized Independent Set*, an edge absent from the graph is equivalent to an edge of cost 0. The standard problem, *Independent Set*, is a special case of the generalized problem, *Generalized Independent Set*, in which all edge penalties have infinite weight. Since *Independent Set* is NP-hard, it follows that *Generalized Independent Set* also is, for a polynomial time algorithm for the latter would clearly also be a polynomial time algorithm for the former. Since our eventual aim to reduce instances of the forest harvesting problems to instances of *Generalized Independent Set*, the fact that *Generalized Independent Set* is NP-hard, and hence *very* unlikely to be amenable to an efficient solution procedure, would seem to be problematic.

As we will show next, however, for certain instances of the *Generalized Independent Set* problem in which the underlying graph G is assumed to have some special structure, we can indeed find an optimal solution with an algorithm whose running time is polynomial in the size of the input. In particular, we will develop such an efficient algorithm to handle instances of *Generalized Independent Set* in which the underlying graph is *bipartite*. A bipartite graph is a graph whose vertices can be partitioned into two sets such that no edge in the graph has both endpoints in the same set. We will show later in the paper that both forest harvesting problems can be cast as *Generalized Independent Set* problems; furthermore, when a grid-like cellular structure is assumed, an optimal solution can be found via the aforementioned polynomial time algorithm. We turn now to developing this algorithm.

Minimum Cuts

In this subsection we reduce the *Generalized Independent Set* problem on bipartite graphs to looking for a *minimum s-t cut* in a network. The minimum s-t cut problem is a well-studied problem that arises in the theory of network flows; hence, we can employ one of the many known efficient algorithms to solve instances of this problem. The reduction of *Generalized Independent Set* to minimum s-t cut is an extension of the usual reduction (see e.g. Harary 1969) of *Independent Set* to minimum s-t cut.

Before describing this reduction, we introduce some notation regarding networks. A *network* $N_G = (N, A)$ is made up of a set of nodes (which we referred to as vertices in graph notation) N and a set of arcs A. Each arc is defined as an ordered pair of nodes (v_i, v_j), and is interpreted as a link directed *from* v_i *to* v_j. Given a subset of the nodes $S \subseteq N$, $(S, \bar{S})$ is the set of arcs directed from S to $\bar{S} = V - S$. This set of arcs is referred to as a *cut.* $(S, \bar{S})$ is an s-t cut if $s \in S$ and $t \in \bar{S}$. Often there is a capacity $c_{i,j}$ associated with the arc (i, j); the capacity of a cut $(S, \bar{S})$, denoted $\mathsf{Cap}(S, \bar{S})$, is equal to the sum of capacities of the arcs in the cut. For a thorough description of minimum cuts, and the more general topic of *network flows*, the reader is referred to a book by Ahuja, Magnanti and Orlin (1993).

We move now to the reduction of *Generalized Independent Set* on bipartite graphs to minimum s-t cut. Suppose we are given an instance of *Generalized Independent Set* on a bipartite graph $G = (V, E)$ in which V_1 and V_2 partition the vertices V into two sets such that no edges in E are in (V_1, V_2). Form the network $N_G = (N, A)$ as follows:

Nodes N: $N = \{s, t\} \cup V$.

Arcs A: $A = \{(s, v) | v \in V_1\} \ \cup \ \{(v, t) | v \in V_2\} \ \cup \ \{(v_1, v_2) | \{v_1, v_2\} \in E\}$. Arcs of the form (s, v) or (v, t), where $v \in V$, have capacity W_v; arcs of the form (v_1, v_2), where $e = \{v_1, v_2\} \in E$, have capacity P_e.

See Figure 4 for an illustration of what such a network looks like.

We now provide a 1-to-1 correspondence between s-t cuts in N_G and (not necessarily optimal) solutions to the *Generalized Independent Set* problem in G, such that sum of the capacity of the cut and the value of the generalized independent set is constant.

Definition: For $(N_S, \bar{N_S})$ an s-t cut in N_G, let $S = (N_S \cap V_1) \cup (\bar{N_S} \cap V_2)$ be its corresponding subset of vertices in G.

It is easy to see that this establishes a bijection (i.e. 1-to-1 correspondence) between s-t cuts in N_G and subsets of the vertices of G. We adopt the convention that $W(S) = \sum_{v \in S} W_v$, and that $\mathsf{GIS}(S)$ is the value of the generalized independent set S.

Lemma 1 *Let $(N_S, \bar{N_S})$ be an s-t cut in N_G and let S be its corresponding subset of vertices in G. Then, the sum of the capacity of the cut and the value of the corresponding Generalized Independent Set solution is a constant. More precisely,*

$$\mathsf{Cap}(N_S, \bar{N_S}) \ + \ \mathsf{GIS}(S) \ = \ W(V) \ .$$

Proof: First of all,

$$\begin{aligned}
\mathsf{Cap}(N_S, \bar{N_S}) &= W(\bar{N_S} \cap V_1) + W(N_S \cap V_2) + \sum_{(i,j) \in A: i \in (N_S \cap V_1), j \in (\bar{N_S} \cap V_2)} P_{\{i,j\}} \\
&= W((\bar{N_S} \cap V_1) \cup (N_S \cap V_2)) + \sum_{\{i,j\} \in E: i,j \in S} P_{\{i,j\}}
\end{aligned}$$

Secondly,

$$\begin{aligned}
\mathsf{GIS}(S) &= W(S) - \sum_{\{i,j\} \in E: i,j \in S} P_{\{i,j\}} \\
&= W((N_S \cap V_1) \cup (\bar{N_S} \cap V_2)) - \sum_{\{i,j\} \in E: i,j \in S} P_{\{i,j\}}
\end{aligned}$$

So, the result follows by adding together the expressions for $C(N_S, \bar{N_S})$ and $GIS(S)$ (notice that the terms involving the penalties, $P_{\{i,j\}}$, cancel out):

$$\begin{aligned}
\mathsf{Cap}(N_S, \bar{N_S}) + \mathsf{GIS}(S) &= W((N_S \cap V_1) \cup (\bar{N_S} \cap V_2)) + W((\bar{N_S} \cap V_1) \cup (N_S \cap V_2)) \\
&= W((N_S \cap V_1) \cup (\bar{N_S} \cap V_2) \cup (\bar{N_S} \cap V_1) \cup (N_S \cap V_2)) \\
&= W(V)
\end{aligned}$$

■

In other words, the smaller the capacity of the s-t cut in N_G, the larger the value of the corresponding generalized independent set in G – and vice-versa. Thus we have the following corollary to Lemma 1:

Corollary 1 *Let S^* be a subset of vertices of G corresponding to an s-t cut $(N_{S^*}, \bar{N_{S^*}})$ in N_G. If S^* is an optimal solution to Generalized Independent Set in G, then $(N_{S^*}, \bar{N_{S^*}})$ is a minimum capacity s-t cut in N_G, and vice-versa.*

So, we have reduced an instance of *Generalized Independent Set* on a bipartite graph to an instance of the minimum s-t cut problem. The minimum s-t cut problem is of fundamental concern in the theory of network flows and, as such, has been extensively studied. Many efficient algorithms have been developed for this problem. In particular, there exists an $O(nm \log \frac{n^2}{m})$ algorithm due to Goldberg and Tarjan (1988) for finding a minimum cut in a network with $O(n)$ nodes and $O(m)$ arcs; this yields an $O(n^2 \log n)$ algorithm for sparse networks in which $m = O(n)$. Observe that if n and m are the number of vertices and edges, respectively, in the graph G, then $n + 2$ and $n + m$ are the number of nodes and arcs, respectively, in the corresponding network N_G. This implies that *Generalized Independent Set* on a bipartite graph with n vertices and m edges can be solved in $O(n^2 \log n)$ time. There are many other simpler, though theoretically less efficient, algorithms for minimum s-t cut. All known minimum cut algorithms solve a maximum network flow problem first, from which the minimum cut is derived; see Ahuja *et al.* (1993) for a thorough treatment of the general topic of network flows.[2]

Efficient Algorithms for the Forest Harvesting Problems

Recall the graph corresponding to a given cell structure, with vertices and edges in the graph representing cells and borders in the cell structure, respectively. The goal is then to optimally select (i.e. harvest) a subset of vertices, where the criteria for optimality depends on the weights associated with the vertices and edges (cells and borders) and the forest harvesting problem being considered.

Notice that when a grid-like cellular structure is assumed, then the corresponding graph is bipartite. Recall that a bipartite graph is a graph whose vertices can be partitioned into two sets such that each edge in the graph has exactly one endpoint in each set. That the grid-like cell structure is represented by a bipartite graph can be observed by assigning (x, y) coordinates to each cell, indicating position, and partitioning the cells into two groups based on the parity of $x + y$; in other words, cells for which $x + y$ is even are in one set, and cells for which $x + y$ is odd are in another set. It is easy to see that neighboring cells never have the same parity, which implies that the resulting graph is bipartite. We make the further observation that the number of edges m in the graph is equivalent to the number of shared borders in the cellular structure, which for the grid is $O(n)$ $(m \leq 4n)$.

We show, in this section, how to reduce an instance of each forest harvesting problem to an instance of *Generalized Independent Set*. When a grid-like cell structure is assumed, or indeed any cell structure whose corresponding graph is bipartite, this implies that the forest harvesting problem can be solved in polynomial time. When assuming a grid-like cell structure, the number of edges, m, is $O(n)$, which yields a running time of $O(n^2 \log n)$ for determining an optimal harvesting strategy. More generally, we note that any cell structure, where each cell is a contiguous piece of land, has a corresponding graph that is *planar* (that is, the graph can be drawn so that no two edges cross) and that planar graphs also have a number of edges m that is $O(n)$.

Problem 1: Cutover Adjacency

Recall that in this forest harvesting problem, each cell i has a benefit H_i associated with the decision to harvest that cell; however, if two cells sharing a common border e are both harvested, then a cost C_e is incurred. So, if we assign a weight $W_i = H_i$ to each vertex i and a penalty

[2] We note that the minimum s-t cut problem can also be solved in polynomial time via linear programming (see Ahuja *et al.* 1993). Thus, while not as efficient as many of the special purpose algorithms that have been developed for this problem, the linear programming formulation may be attractive to many due to the widespread accessibility of and familiarity with linear programming software.

$P_e = C_e$ to each edge e, then clearly we have an instance of the *Generalized Independent Set* problem. As noted earlier, when a grid-like cell structure is assumed this leads to an $O(n^2 \log n)$ time algorithm for determining an optimal spatial layout. An example is given in the appendix.

Consider two extremes in the harvesting problem. If $C_e = 0$ for all borders e, then the optimal land management decision is to harvest every cell. On the other hand, if $C_e = \infty$ for all borders e, then the forest harvesting problem reduces to an instance of the standard *Independent Set* problem.

Finally, we observe that this forest harvesting problem and the *Generalized Independent Set* problem are essentially equivalent. One difference, however, is that because the harvesting problem arises from a partitioning into contiguous cells of an area of forest to be spatially managed, the underlying graph will always be planar, whereas no such restriction is assumed in *Generalized Independent Set.* We note that, because *Independent Set* remains NP-hard even when restricted to instances on planar graphs (see Garey, Johnson and Stockmeyer 1976).

Problem 2: Harvesting with Edge Effects

Let x_i be a decision variable (taking on the value 0 or 1) indicating whether or not cell i is to be harvested. In graph terminology,

$$x_i = \begin{cases} 1, & \text{if } i \in S \\ 0, & \text{otherwise.} \end{cases}$$

For decision variables x_i and x_j, $x_i \odot x_j$ is referred to as the *exclusive-or* of x_i and x_j; $x_i \odot x_j$ takes on the value 0 if $x_i = x_j$, and 1 otherwise. Observe that,

$$x_i \odot x_j = (x_i - x_j)^2 = x_i{}^2 + x_j{}^2 - 2x_i x_j = x_i + x_j - 2x_i x_j \ ,$$

where the final equality uses the fact that $x^2 = x$ for a 0-1 decision variable x.

Recall that in this problem, each cell i has a benefit H_i associated with harvesting and a benefit U_i associated with not harvesting, and each border $\{i,j\}$ has a benefit $B_{\{i,j\}}$ associated with a space allocation decision that calls for harvesting exactly one of cells i or j. Now, for a given choice of x_i's, the benefit associated with the corresponding space allocation decision is thus given as ($\bar{x}_i = 1 - x_i$ denotes the decision opposite to that of x_i),

$$\begin{aligned} & \sum_{i \in V} H_i x_i + \sum_{i \in V} U_i \bar{x}_i + \sum_{\{i,j\} \in E} B_{\{i,j\}} [x_i \odot x_j] \\ = & \sum_{i \in V} [H_i x_i + U_i \bar{x}_i] + \sum_{\{i,j\} \in E} B_{\{i,j\}} (x_i + x_j - 2x_i x_j) \\ = & \sum_{i \in V} [H_i x_i + U_i (1 - x_i)] + \sum_{\{i,j\} \in E} B_{\{i,j\}} (x_i + x_j - 2x_i x_j) \end{aligned}$$

$$= \sum_{i\in V} U_i + \sum_{i\in V}(H_i - U_i)x_i + \sum_{\{i,j\}\in E} B_{\{i,j\}}(x_i + x_j - 2x_ix_j)$$
$$= \sum_{i\in V} U_i + \sum_{i\in V}[H_i - U_i + \sum_{\{i,j\}\in E} B_{\{i,j\}}]x_i - \sum_{\{i,j\}\in E} 2B_{\{i,j\}}x_ix_j$$

As we have noted, x_i indicates whether or not i is included in the set S of vertices that corresponds to cells selected to be harvested. Thus, the benefit associated with the space allocation decision to harvest the cells corresponding to the vertices S is given as,

$$\sum_{i\in V} U_i + \sum_{i\in S}[H_i - U_i + \sum_{\{i,j\}\in E} B_{\{i,j\}}] - \sum_{\{i,j\}\in E: i,j\in S} 2B_{\{i,j\}}\ .$$

The first term is a constant, so our *optimal* harvesting strategy corresponds to solving an instance of the *Generalized Independent Set* problem on the graph describing the given cell structure, with vertex i having weight $W_i = H_i - U_i + \sum_{\{i,j\}\in E} B_{\{i,j\}}$, and edge $\{i,j\}$ having penalty $P_{\{i,j\}} = 2B_{\{i,j\}}$. As before, when a grid-like cell structure is assumed this leads to an $O(n^2 \log n)$ time algorithm for determining an optimal spatial layout. An axample is given in the appendix.

We make a few further observations. First, notice that if W_i is less than 0 then there is no point in selecting that vertex; this makes sense, for it means that the benefit U_i of not harvesting outweighs the benefit of harvesting H_i plus all potential edge benefits for those borders incident to cell i. Second, recall that we specify two values, U_i and H_i, for each cell i. The expression derived above measuring the benefit of a particular space allocation decision suggests that an optimal harvesting strategy can be found by simply specifying a "net value" $H_i - U_i$ for each cell, rather than requiring each quantity to be given individually. Finally, we note that in the paper Hof and Joyce (1992), in which this harvesting problem was first raised, it was assumed that the three types of benefits H_i, U_i, and $B_{\{i,j\}}$, were constant for all i, j; that is, only three values H, U, and B were needed to describe the benefits. For problems with such additional structure, it is possible that more efficient algorithms can be derived to find the optimal harvesting strategy; indeed, one would expect the optimal solution to have some sort of regular pattern to it when the benefits are constant and a grid-like cell structure is assumed, suggesting perhaps that one could even describe the optimal spatial layout analytically.

Concluding Remarks

In this paper, we have considered two forest harvesting problems in which a decision as to how to manage the land is to be made. These were one-stage decision problems in which it was to be decided which parts of the land were to be harvested and which were not. Non-timber concerns of wildlife habitat were taken into consideration, in slightly different ways, in the two problems considered. For both problems, we provided a reduction to *Generalized Independent Set*, a newly introduced graph optimization problem. We developed a polynomial time solution method for *Generalized Independent Set* on bipartite graphs and showed that when a grid-like cell structure is assumed in the harvesting problems, then this efficient algorithm can be applied to determine an optimal spatial layout for either forest harvesting problem.

We raise several issues with regards to the forest harvesting problems given and the solution strategy developed:

With regards to the forest harvesting problems there are two issues.

1. It must be decided when constructing the cell structure how fine the resolution of the land to be managed should be. In the problem concerning preserving old growth habitat, the cell size is almost predetermined by the problem definition and there may little flexibility in altering it; for the problem dealing with edge effects, however, one can imagine that an optimal harvesting strategy determined for a cell structure with finer resolution may lead to a better spatial allocation of the land. The finer the resolution, the greater the number of cells. By knowing the running time of the algorithm for determining an optimal spatial layout, the tradeoff between cell resolution and algorithm running time can be evaluated.

2. There is the issue of determining the parameters (benefits/costs) for the two harvesting problems. It is clearly difficult to give values for both timber and non-timber (wildlife habitat) concerns that everyone will agree to and that can be calibrated against each other. We suggest that future work should take sensitivity analysis issues into account to determine how the optimal spatial layout changes as some of the parameter values change.

Now, with regards to the solution strategy developed there are also two issues.

1. Our polynomial time algorithm for *Generalized Independent Set* requires that the graph be bipartite. For certain cell structures, this may not be the case. Nevertheless, one can observe that the graph describing a cell structure will be planar; It would be interesting to develop a sub-exponential algorithm or a polynomial approximation scheme for planar instances of *Generalized Independent Set*, as was done for *Independent Set* (see Lipton and Tarjan (1980) and Baker (1994), resp.).

   Alternatively, one can consider solving a linear integer programming formulation of *Generalized Independent Set*, either by developing special-purpose methods or via some general-purpose technique such as branch-and-bound. Consider an instance of *Generalized Independent Set* on the graph $G = (V, E)$ (recall the formal problem definition given previously). Letting x_i be a binary variable indicating the inclusion of vertex i in S, and letting x_e be a binary variable corresponding to the edge $e = \{i, j\}$ indicating that both vertices i and j are in S, we can formulate *Generalized Independent Set* as a linear integer program:

$$\begin{array}{llll} \max & \sum_{i \in V} W_i x_i \; - \; \sum_{e \in E} P_e x_e & & \\ \text{subject to} & x_e \geq x_i + x_j - 1 \; , & \forall \; e = \{i, j\} \in E \\ & x_i \in \{0, 1\} \; , & \forall \; i \in V \\ & x_e \in \{0, 1\} \; , & \forall \; e \in E \end{array}$$

The correctness of the above formulation follows by observing that, in an optimal solution, $x_e = 1$ if and only if $x_i = x_j = 1$ for any edge $e = \{i, j\} \in E$.

2. Consider the forest harvesting problem dealing with edge effects. While we reduced any instance of this problem to an instance of *Generalized Independent Set*, and showed how to efficiently solve instances where the graph corresponding to the underlying cell-structure was bipartite, we observe that the forest harvesting problem can also be viewed as a generalized *maximum cut* problem (see Garey and Johnson 1979 for a description of the maximum cut problem). While in maximum cut the objective is to maximize the weight of the edges in a cut, the graph optimization problem corresponding to this forest harvesting problem also includes different benefits for each vertex, depending on which side of the cut it is assigned. By generalizing the results of Goemans and Williamson (1994) we have been able to develop (Hochbaum and Pathria 1995) a polynomial time algorithm for this forest harvesting problem that finds, for any cell structure, a spatial layout with overall benefit within a factor of .878 of the optimal. We also note that while the maximum cut problem is NP-hard for general graphs, it can be solved in polynomial time for planar graphs (see Hadlock 1975). Whether or not this forest harvesting problem is also amenable to a polynomial time solution for any cell structure is an open question.

In addition to providing efficient solutions to two forest harvesting problems, we believe that there are two other important contributions of this paper. First of all, it is hoped that we have demonstrated that a familiarity with areas of operations research can be valuable for decision making in applications in the sciences and engineering in general, and applications in forestry in particular. It can be very useful for the decision maker to have familiarity with analytic tools other than linear programming and dynamic programming, the two optimization techniques that seem to be most widely known outside of the operations research community. Second, in introducing the generalization of the *Independent Set* problem, we have exploited the idea of relaxing the structural constraints required in a solution at the cost of incurring a penalty. In *Generalized Independent Set*, the structural constraint imposed in *Independent Set* of disallowing two vertices that are adjacent from both being in the selected set is relaxed, but at the cost of paying a penalty associated with the "violated" edge. We believe that this *"relaxed-with-penalty"* idea can be a useful and stimulating concept with respect to applications in other areas where constraints are stated but solutions remain feasible even if some constraints are violated.

Finally, we note that the harvesting problems presented in this paper can be generalized and complicated in many ways. For example, one can consider multi-period harvest scheduling problems, problems with more general cell structures, etc. Indeed, Lookwood and Moore (1993) have applied simulated annealing to a timber harvesting problem with spatial constraints, in which "requirements" to comply with exclusion periods and maximum clearcut size restrictions can be violated by incurring a cost. It would be interesting to extend the ideas presented in this paper to other forest harvesting problems.

Appendix 1. Numerical Examples of Forest Harvesting Problems

We consider an example for each of the two forest harvesting problems on a 3×3 grid-like cell structure as depicted below.

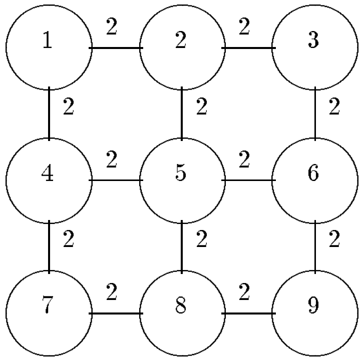
1	2	3
4	5	6
7	8	9

Cutover Adjacency

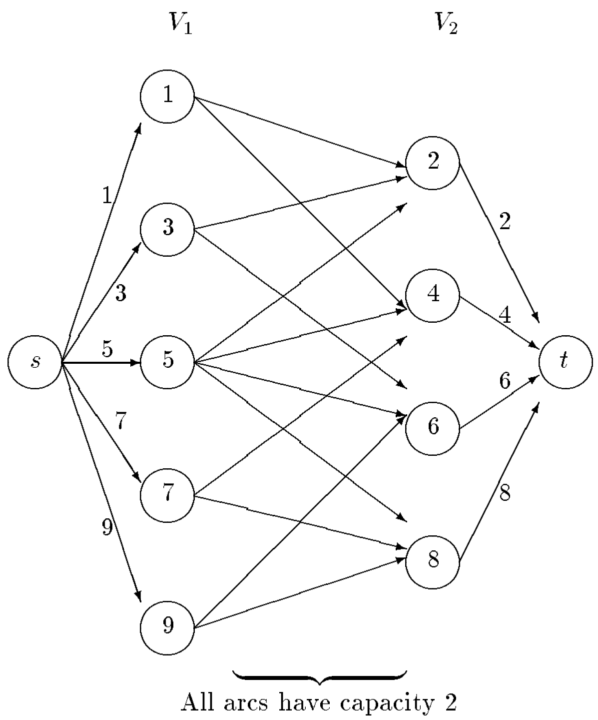
Instance: Consider the following instance of the problem: each cell i has a benefit $H_i = i$ if harvested, and each border e has a cost $C_e = 2$ associated with it if both cells incident to e are harvested.

Solution:

1. *Reduce to an instance of Generalized Independent Set.*
   The corresponding instance of *Generalized Independent Set* is given below, with weight $W_i = H_i$ associated with vertex i, and penalty $P_e = C_e$ associated with edge e.



2. *Formulate corresponding minimum s-t cut problem.*
   The minimum s-t cut problem corresponding to the given instance of *Generalized Independent Set* is depicted below, with arc capacities given:



3. *Find optimal solution via minimum cut in network.*
   The minimum cut in the network is $(N_S, \bar{N}_S)$, where

$$N_S = \{s, 7, 9\} \text{ and } \bar{N}_S = \{1, 2, 3, 4, 5, 6, 8, t\} .$$

   The corresponding optimal solution to the *Generalized Independent Set* problem is

$$\begin{aligned} S &= (N_S \cap V_1) \cup (\bar{N}_S \cap V_2) \\ &= \{7, 9\} \cup \{2, 4, 6, 8\} \\ &= \{2, 4, 6, 7, 8, 9\} . \end{aligned}$$

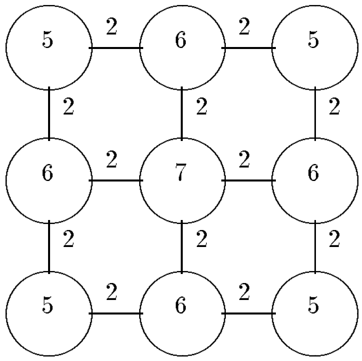
   Thus, the optimal harvesting strategy is to harvest cells $\{2, 4, 6, 7, 8, 9\}$.

Harvesting with Edge Effects

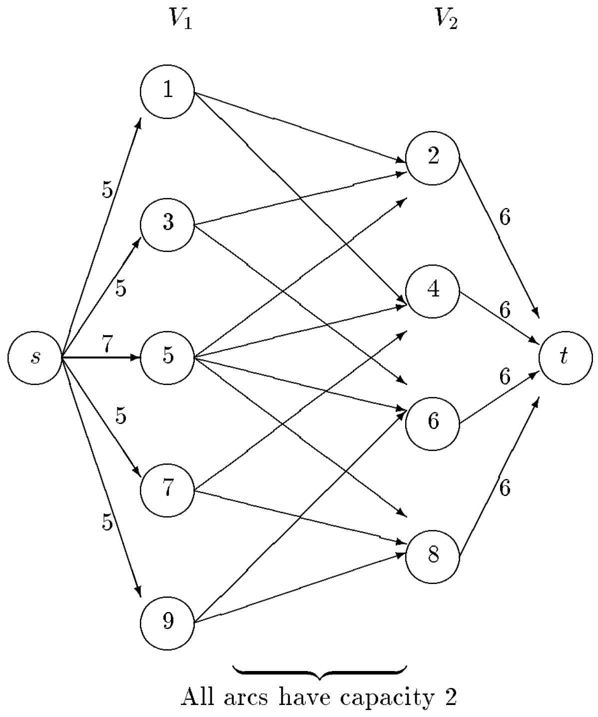
Instance: Consider the following instance of the problem: each cell i has a benefit $H_i = 5$ if harvested or $U_i = 2$ if not harvested, and each border e has a benefit $B_e = 1$ if exactly one of the two cells sharing the border e is harvested.

Solution:

1. *Reduce to an instance of Generalized Independent Set.*
   The corresponding instance of *Generalized Independent Set* is given below, with weight $W_i = H_i - U_i + \sum_{\{i,j\} \in E} B_{\{i,j\}}$ associated with vertex i, and penalty $P_e = 2B_e$ associated with edge e.



2. *Formulate corresponding minimum s-t cut problem.*
   The minimum s-t cut problem corresponding to the given instance of *Generalized Independent Set* is depicted below, with arc capacities given:



3. *Find optimal solution via minimum cut in network.*
The minimum cut in the network is $(N_S, \bar{N}_S)$, where

$$N_S = \{s, 1, 3, 7, 9\} \text{ and } \bar{N}_S = \{2, 4, 5, 6, 8, t\} .$$

The corresponding optimal solution to the *Generalized Independent Set* problem is

$$\begin{aligned} S &= (N_S \cap V_1) \cup (\bar{N}_S \cap V_2) \\ &= \{1, 3, 7, 9\} \cup \{2, 4, 6, 8\} \\ &= \{1, 2, 3, 4, 6, 7, 8, 9\} . \end{aligned}$$

Thus, the optimal harvesting strategy is to harvest cells $\{1, 2, 3, 4, 6, 7, 8, 9\}$.

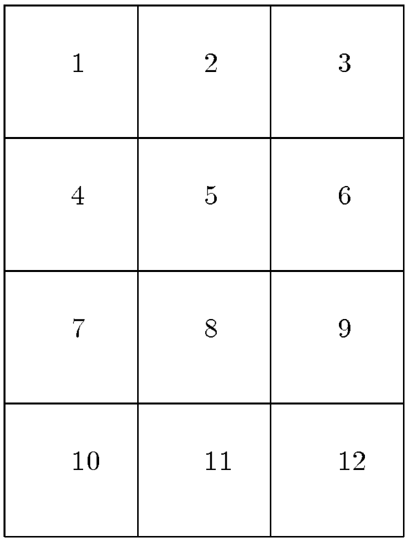


Figure 1. A 4×3 grid-like cell structure.

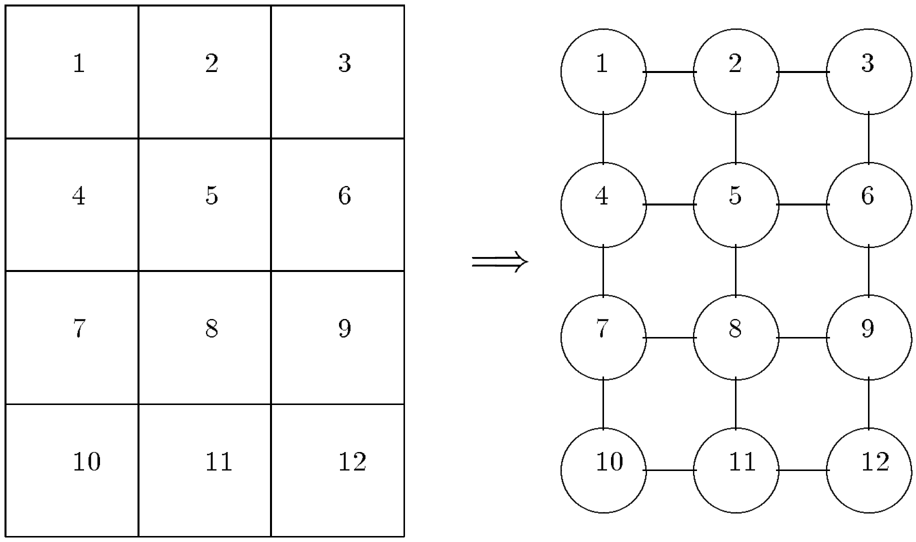
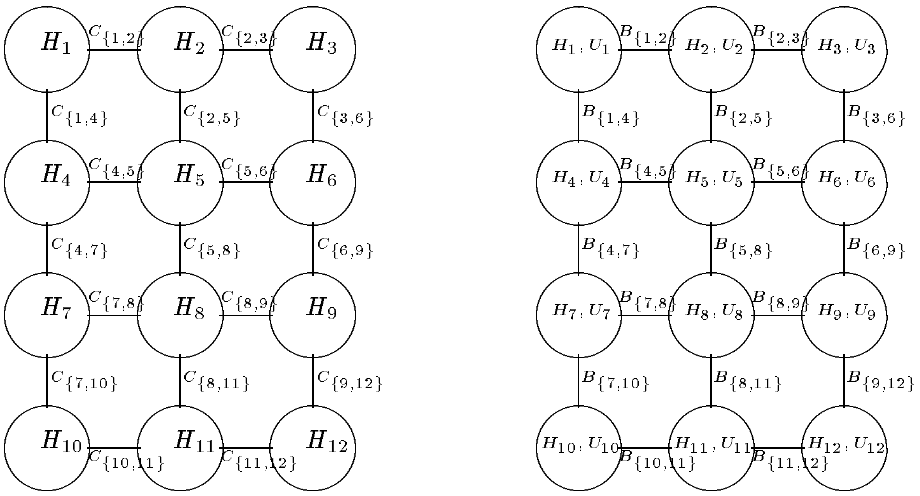


Figure 2. 4×3 grid-like cell structure, and corresponding representation as a graph.



(a) (b)

Figure 3. Graph corresponding to 4×3 grid-like cell structure, with weights labeled: (a) *cutover adjacency* problem; (b) *harvesting with edge effects* problem.

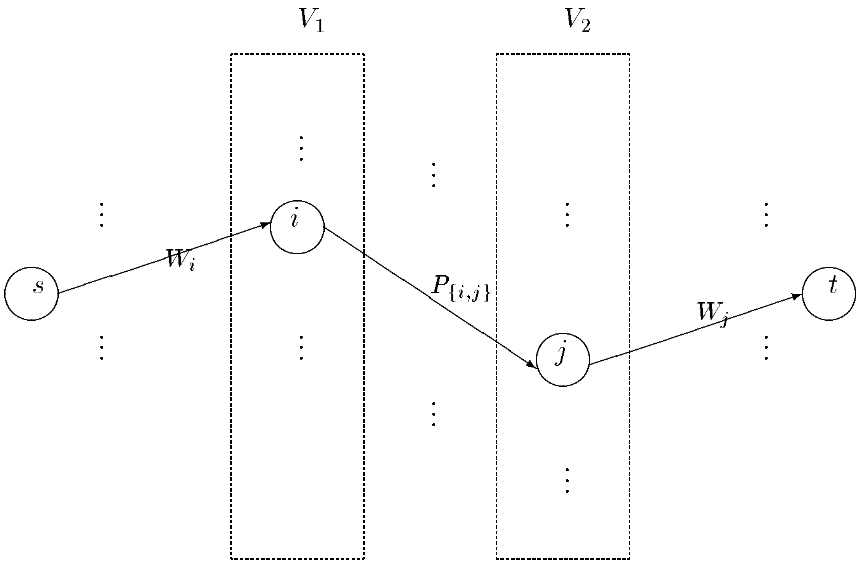


Figure 4. Network N_G on which minimum s-t cut corresponds to optimal *Generalized Independent Set* for bipartite graph $G = (V_1 \cup V_2, E)$.

	$f(n)$			
n	$\log_2 n$	n	n^3	2^n
1	0	1	1	2
10	3.32	10	10^3	1.02×10^3
100	6.64	10^2	10^6	1.27×10^{30}
1000	9.97	10^3	10^9	1.07×10^{301}

Table 1. Growth of various functions $f(n)$ in terms of n.

References

[1] Ahuja R. K., T. L. Magnanti, and J. B. Orlin. 1993. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Englewood Cliffs, NJ.

[2] Baase S. 1988. Computer Algorithms: Introduction to Design and Analysis. Addison-Wesley, Reading, MA.

[3] Baker B. S. 1994. Approximation Algorithms for NP-Complete Problems on Planar Graphs. Journal of the Association for Computing Machinery, 41(1):153–180.

[4] Barahona F., A. Weintraub, and R. Epstein. 1992. Habitat Dispersion in Forest Planning and the Stable Set Problem. Operations Research, 40:14–21.

[5] Garey M. R. and D. S. Johnson. 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco.

[6] Garey M. R., D. S. Johnson, and L. Stockmeyer. 1976. Some Simplified NP-Complete Graph Problems. Theoretical Computer Science, 1:237–267.

[7] Goemans M. X. and D. P. Williamson. 1995. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. Journal of the ACM, 42(6):1115–1145.

[8] Goldberg A. V. and R. E. Tarjan. 1988. A New Approach to the Maximum-Flow Problem. Journal of the Association for Computing Machinery, 35(4):921–940.

[9] Hadlock F. O. 1975. Finding a Maximum Cut of a Planar Graph in Polynomial Time. SIAM Journal on Computing, 4:221–225.

[10] Harary F. 1969. Graph Theory. Addison-Wesley, Reading, MA.

[11] Hochbaum D. S. and A. Pathria. 1995. An Approximation Algorithm for a Generalized MAX 2-SAT Problem. manuscript, 6p.

[12] Hof J. G. and L. A. Joyce. 1992. Spacial Optimization for Wildlife and Timber in Managed Forest Ecosystems. Forest Science, 38(3):489–508.

[13] Hof J. G. and L. A. Joyce. 1993. A Mixed Linear Integer Programming Approach for Spatially Optimizing Wildlife and Timber in Managed Forest Ecosystems. Forest Science, 39(4):816–834.

[14] Lipton R. J. and R. E. Tarjan. 1980. Applications of a Planar Separator Theorem. SIAM Journal of Computing, 9(3):615–627.

[15] Lockwood C. and T. Moore. 1993. Harvest Scheduling with Spatial Constraints: a Simulated Annealing Approach. Canadian Journal of Forest Research, 23:468–478.

[16] Papadimitriou C. H. and K. Steiglitz. 1982. Combinatorial Optimization: Algorithms and Complexity. Prentice Hall, Englewood Cliffs, NJ.