

# Imposing Connectivity Constraints in Forest Planning Models

Rodolfo Carvajal

H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332,  
Escuela de Negocios, Universidad Adolfo Ibáñez, Santiago, Chile, rocarvaj@gatech.edu

Miguel Constantino

Centro de Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa, miguel.constantino@fu.ul.pt

Marcos Goycoolea

Escuela de Negocios, Universidad Adolfo Ibáñez, Santiago, Chile, marcos.goycoolea@uai.cl

Juan Pablo Vielma

Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA 15260, jvielma@pitt.edu

Andrés Weintraub

Departamento de Ingeniería Industrial, Universidad de Chile, Santiago, Chile; Instituto Milenio Sistemas Complejos de Ingeniería, Santiago, Chile, aweintraub@dii.uchile.cl

Connectivity requirements are a common component of forest planning models, with important examples arising in wildlife habitat protection. In harvest scheduling models, one way of addressing preservation concerns consists in requiring that large contiguous patches of mature forest are maintained. In the context of nature reserve design, it is common practice to select connected regions of forest in such a way as to maximize the number of species and habitats protected. While a number of integer programming formulations have been proposed for these forest planning problems, most are impractical in that they fail to solve reasonably sized scheduling instances. We present a new integer programming methodology and test an implementation of it on five medium-sized forest instances publicly available in the FMOS repository. Our approach allows us to obtain near-optimal solutions for multiple time-period instances in less than four hours.

*Key words:* integer programming; cutting plane; natural resources

---

## 1. Introduction

During the last decades there has been much interest in incorporating environmental and aesthetic concerns into forest planning models. While there is no consensus on how these concerns should be fully addressed, a number of management practices have been consistently promoted by voluntary initiatives and certifications such as the USA 2010-2014 Sustainable Forestry Initiative (SFI) (2010)

or the Forest Stewardship Council (FSC) (2006). Three very important management practices with which we are concerned in this paper are as follows:

- Maximum clear-cut size constraints. Large clearcut areas are unaesthetic, facilitate erosion and may have a negative impact on the wildlife living in surrounding areas. To counter this, maximum clearcut size constraints dictate a maximum opening size of contiguous harvested areas. For example, the SFI recommends an average 50 hectare size limit. This type of norm is a legal requirement in many countries (for example Sweden, 20 ha.) and the United States (for example Oregon, 48 ha., and Maine, 101 ha.).

- Conservation of Old-Growth-Forests. Wildlife protection has been mainly ensured by the existence of reserves. Nevertheless, the importance of managed forests as a complement to nature reserves has been widely recognized as a means to protect wildlife and biodiversity (Aldrich et al. 2004). While there are a variety of biodiverse habitats in forests, many species of animals mostly live in old-growth forest habitats, which are in short supply due to resource exploitation. Thus it is common to require that large contiguous patches of mature forest (typically over 120 or 180 years old) be preserved. Rebas and McDill (2003a) review different sources on why small habitats may not accommodate wildlife species. Some reasons include insufficient territory for mating and breeding, lack of food, and increased predation and brood parasitism. Recent studies highlighting the global importance of old-growth forests as a carbon sink have sparked renewed interest in their conservation (Keith et al. 2009, Luyssaert et al. 2008). Promoting old-growth patches is also a way of mitigating the negative effects of maximum clear-cut size constraints Barrett et al. (1998), Gustafson and Crow (1998).

- Natural reserve site selection. This consists in designing one or several contiguous sites for a natural reserve (wildlife refuge, national park, etc) in such a way as to protect a specific list of species and/or preserve certain habitat types. Both the SFI and FSC standards are very specific in the need to have protected areas for threatened and endangered species. However, there is much debate on *how* reserves should be designed. For example, if there should be wildlife corridors

connecting different sites (Earn et al. 2000) or if a few large sites are preferable to many smaller sites (Etienne and Heesterbeek 2000).

These three management practices have a very important feature in common: they all require that stands selected for harvesting or protection comply with some form of connectivity. In this article we study the problem of modeling connectivity in integer programming (IP) models. Our aim is not to advocate any particular way of addressing environmental concerns, but rather, to provide a more unified computational methodology capable of imposing connectivity in larger data sets and for different environmental constraints.

A number of authors have studied IP approaches for these type of problems.

The first attempts to incorporate maximum clearcut size constraints are due to Thompson et al. (1973). Murray (1999) formalizes this problem in terms of basic management stands by introducing the Area Restriction Model (ARM). Goycoolea et al. (2005), McDill et al. (2002) and Constantino et al. (2008) propose different integer programming formulations for the ARM. Vielma et al. (2007) and Tóth (2005) introduce computational improvements for these formulations and Goycoolea et al. (2009) present a survey with computational results and modeling extensions. An important feature is that all of these models (with the exception of that found in Constantino et al. (2008)) deal with connectivity by explicitly enumerating all possible combinations of connected subsets satisfying certain properties. That these formulations work well in practice is due to the fact that maximum clearcut size constraints are typically three to four times the size of an average stand in the instances considered. Hence the number of potential clearcut regions is not too large.

Caro et al. (2003) propose an extension of the ARM model that considers old-growth patches with enough area to be a wildlife habitat. Though an IP formulation is introduced, Tabu search is used to obtain feasible solutions for the model. Rebain and McDill (2003b) consider a similar model and propose solving it with integer programming. They consider decision variables for every possible connected patch that has an area that just exceeds the one required by the problem. This approach is illustrated with an instance with 50 stands and 3 periods. Martins et al. (2005) use

a column generation approach to tackle the large amount of variables in this formulation, solving atemporal instances with up to 400 units.

The first attempts to quantitatively model the natural reserve selection problem dates back to Kirpatrick (1983). See Williams et al. (2005) for a comprehensive survey of articles that have followed. The problem of connecting dispersed reserves has also been considered. Sessions (1992) formulates this problem as a Steiner network problem and proposes a heuristic based on shortest paths. Williams (1998) uses an IP formulation with edge variables and flow conservation constraints to connect fixed reserves.

A number of articles study the problem of connectivity while also considering the shape of the natural reserve or old-growth patch. Tóth et al. (2006) use a bi-criteria approach to find a harvest schedule maximizing both the profit and the area of standing old-growth forest. Their model is based on the formulation of Rebain and McDill (2003b). This work is extended to minimize the perimeter of standing old-growth forest in Tóth and McDill (2008). They present results for a 50 stand data set. Öhman and Låmas (2005) consider a two-objective model where the *Shape Index* (deviation from a circle) is heuristically minimized with simulated annealing. Öhman and Wikström (2008) consider a two-objective problem where they minimize perimeter as in Tóth and McDill (2008), but without requiring connectivity of the old forest. Öhman (2000) considers the creation of contiguous patches with a large core area. Önal and Briers present a formulation for promoting compactness (2002) and for minimizing perimeter (2003). Connectivity is not guaranteed in these models. However, these same authors tackle the additional requirement of connectivity by using a characterization of trees in graphs in Önal and Briers (2006), performing computational experiments on a data set with 391 stands. Williams and ReVelle (1998) present an IP formulation that promotes but does not guarantee connectivity. As part of their work, Williams et al. (2005) provide a rich discussion on different types of connectivity and approaches used to model them.

### **1.1. Our contribution**

In this article we describe a new integer programming framework for modeling connectivity in graphs that is well suited for the forestry applications described above because (a) it only uses

variables on the graph nodes, (b) it can be used to model problems where one or multiple connected components are required, and (c) it can be used to model problems where these components are required (or not) to contain specific nodes or stands. To illustrate our approach we focus on the specific problem where we would like to solve the ARM subject to additional old-growth forest conservation constraints. However, it is easy to see that the approach extends to the other applications previously discussed. As we will see in the computational results section our approach is able to handle instances with more than one thousand stands and multiple time periods. This is a considerable improvement on the results of Rebain and McDill (2003a) and Martins et al. (2005), as well as on those observed in other forestry applications where connectivity is considered. Moreover, we find that the formulations we propose enjoy very tight linear programming relaxation gaps in practice. All of our tests are performed on publically available data sets.

## 2. An Example: Harvest scheduling model and old-growth forest

To motivate the need for connectivity constraints and illustrate the applicability of our proposed approach we consider a harvest scheduling problem with both maximum clear-cut constraints and old-growth conservation requirements.

The objective of this problem is to schedule the harvesting of forest stands to maximize profits, while preventing large clear-cut areas, providing a steady timber flow, maintaining a minimum average ending age of the forest and a connected region of old-growth forest (old-growth patch). With the exception of the connectivity requirements on the old-growth forest all this constraints can be effectively modeled using MILP (Goycoolea et al. 2009). We now describe one such model that utilizes a formulation of the clear-cut requirements that was introduced in McDill et al. (2002).

To describe the model we use the following notation. Let  $V$  be the set of forest stands or management units. Two stands are said to be adjacent if they share a non discrete boundary in the plane. Let  $T = \{1, \dots, N\}$  be a planning horizon of time periods in which each unit can be harvested at most once. We assume period  $t$  is  $Y_t$  years long and also that the management actions (harvesting) occur at the end of each time period. We also let  $W_t = \sum_{s=1, \dots, t} Y_t$  be the number of

years from the beginning of period one up to the end of period  $t$ . We suppose the following data associated to stand  $v$  is known: the area  $a_v$ ; the volume of timber obtained  $\alpha_{v,t}$ , if it is harvested in period  $t$ ; the age of the trees  $b_v$ , at the beginning of period 1; the net present value of the profit of harvesting in period  $t$ ,  $p_{v,t}$ . We also assume that the average age of the forest at the end of the planning horizon must be greater than  $H$ . For the clear-cut requirements we assume that the maximum clear-cut area is  $A_{max}$  and let  $\Lambda^+$  be the collection of contiguous groups of stands whose combined areas exceed  $A_{max}$  and are minimal for this property under inclusion (i.e. the so-called minimal infeasible clusters (Goycoolea et al. 2009) or paths (McDill et al. 2002)). Finally, for the old-growth requirements we let the minimum area of the old-growth patch be  $A_{min}$  and the minimum age for a stand in the patch  $O_{age}$ . We assume  $O_{age} > W_T$  and hence after a stand is harvested it cannot belong to the old-growth patch in subsequent periods.

The formulation contains two sets of binary variables:

- $y_{v,t}$ : Harvesting variable. 1 if stand  $v$  is scheduled for harvesting on period  $t$  and 0 otherwise.
- $z_{v,t}$ : Patch variable. 1 if stand  $v$  belongs to the old-growth patch on period  $t$  and 0 otherwise.

and is given by

$$\max \sum_{v \in V, t \in T} p_{v,t} y_{v,t} \quad (1a)$$

$$\text{s.t. } z_{v,q} + \sum_{t=1}^q y_{v,t} \leq 1, \quad \forall v \in V, q \in T \quad (1b)$$

$$\sum_{v \in V} a_v z_{v,t} \geq A_{min} \quad \forall t \in T \quad (1c)$$

$$z_{v,t} = 0, \quad \forall v \text{ such that } b_v + W_t < O_{age}, \forall t \in T \quad (1d)$$

$$\sum_{v \in C} y_{v,t} \leq |C| - 1, \quad \forall C \in \Lambda^+, \forall t \in T \quad (1e)$$

$$\sum_{v \in V} \alpha_{v,t} y_{v,t} \geq \left(1 - \frac{L}{100}\right) \sum_{v \in V} \alpha_{v,t-1} y_{v,t-1}, \quad \forall t \in T \setminus \{1\} \quad (1f)$$

$$\sum_{v \in V} \alpha_{v,t} y_{v,t} \leq \left(1 + \frac{U}{100}\right) \sum_{v \in V} \alpha_{v,t-1} y_{v,t-1}, \quad \forall t \in T \setminus \{1\} \quad (1g)$$

$$\sum_{v \in V} a_v \left( b_v + W_N - \sum_{t \in T} (b_v + W_t) y_{v,t} \right) \geq \sum_{v \in V} a_v H \quad (1h)$$

$$y_{v,t} \in \{0, 1\} \quad \forall v \in V, \forall t \in T \quad (1i)$$

$$z_{v,t} \in \{0, 1\} \quad \forall v \in V, \forall t \in T \quad (1j)$$

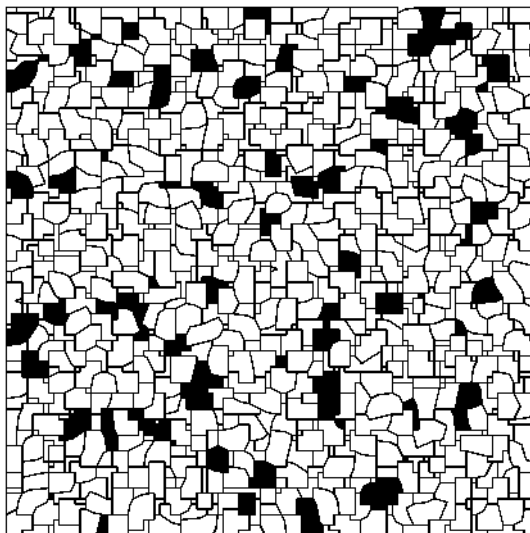
$$\text{For each } t \text{ the set of stands } Z_t = \{v : z_{v,t} = 1\} \text{ is connected.} \quad (1k)$$

Objective (1a) maximizes the net present value of the schedules' profit. Constraints (1b) ensure that each stand unit is harvested at most once during the planning horizon and that a stand can be in the old-growth patch at the end of a period only if it has not been previously harvested. Constraints (1c) and (1d) impose a lower bound on the area of the old-growth patch and only allow old enough stands in the patch. Constraints (1e) are the so called minimal infeasible or path constraints that enforce the maximum clear-cut requirements. Constraints (1f) and (1g) are even volume flow requirements, stating that the volume of timber harvested in period  $t$  cannot be  $U$  percent above or  $L$  percent below the volumes of timber in period  $t - 1$ . Constraint (1h) requires that the average age of the forest at the end of the harvesting horizon is at least  $H$  years. Constraints (1i) and (1j) make the variables binary and constraints (1k) enforce connectivity of the old-growth patch.

With the exception of (1k) all constraints of (1) are linear inequalities or integrality requirements. Hence to transform model (1) to a MILP formulation we only need to replace (1k) by an appropriate MILP formulation of connectivity. We will study such formulation in Section 3. However, before that, we now illustrate why this connectivity constraint is needed for old-growth patches. To achieve this we study the solutions of MILP formulation (1a)–(1j) obtained by removing the connectivity requirements over the patch. We solve this formulation over four three period instances (*El Dorado*, *Shulkell*, *NBCL5A* and *FLG9A*), obtained from the repository of forestry instances of the Forest Management Optimization Site (FMOS) (2008). A detailed description of these instances is given in Section 4. For each instance, Table 1 presents the number of contiguous patches of old-growth forest (Patches) and the area of the largest of them, measured as a percentage of the minimum required area  $A_{min}$ , (Largest) in the best feasible solution found. Figure 1 shows in black the stands selected to be part of the patch in the third period in a solution for the FLG9A forest.

**Table 1** Fragmentation without connectivity.

	Patches	Largest (%)
El Dorado	70	17.6
FLG9A	57	5.8
NBCL5A	78	4.3
Shulkell	10	25

**Figure 1** Old-growth patch for FLG9A without connectivity.

We observe that without explicitly forcing connectivity we obtain solutions that are highly fragmented in terms of the selected old-growth forest. For example, as shown in Table 1, the old-growth stands selected in the FLG9A forest instance are divided into 57 disconnected pieces, each having a total area of no more than 5.8% the total requirement.

### 3. Modeling Connectivity with Integer Programming

An important difficulty in the model described in the previous section is to ensure that the stands selected in the patch are in fact connected to each other, that is, enforcing constraints (1k). In this section we begin by describing different types of connectivity requirements that are common in forest planning models. We then formally describe how these connectivity requirements may be enforced in integer programming models.

Formally, a patch in a forest is said to be *connected* if it is possible to travel between any two points of the patch through a path fully contained in the patch. More often than not, connectivity will not be the only requirement made in forest planning applications. The existence of other

requirements or conditions (which we discuss below) can have an important effect on how connectivity itself is modeled with integer programming, as well on how computationally tractable the different models are.

In some forest planning models decisions are made over time. In this context, if we are to impose connectivity of a patch, it is important to be specific on whether or not the patch is allowed to change over time. If not, we say that the patch is *statically connected*. Otherwise, we say that it is *dynamically connected*. In the latter case it is assumed that the patch must be connected in every time period. In some cases, it may be desirable to impose some limits on how much dynamically connected patches are allowed to change over time. For example, it may be appropriate to require that from one time period to the next, the patch retain at least some specific number of stands. We call this a *temporal connectivity* requirement.

*Rooted connectivity* occurs when some predetermined set of stands is required to belong to a patch. For example, we may want an old-growth patch to contain some riparian zones. To make a distinction, we refer to *unrooted connectivity* if no specific stands are required to be in the patch.

It is often the case that we wish to impose some limit on the size of a patch. For example, we may want to impose a maximum area if the patch is to be a clearcut zone, or a minimum area if the patch is to be an old-growth patch. We refer to these as *constrained connectivity* requirements.

In some applications with constrained connectivity requirements we may allow a selected set of stands (or patch) to be disconnected, as long as each connected subset meets some minimum area condition. We refer to these as *multi-patch connectivity* requirements. Examples of this type of connectivity are found in Caro et al. (2003) and Rebas and McDill (2003a) in the context of old-growth patch modeling. The situation in which exactly one connected patch is required is referred to as *single-patch connectivity*.

### 3.1. Background: Connectivity, Graphs and Integer Programming

In this section we describe how each of the different connectivity requirements described in the previous section can be modeled with integer programming. We begin by introducing some notation.

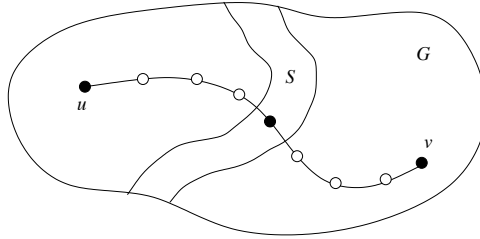
We next review the literature on graph connectivity and integer programming. Finally, we introduce new formulations for modeling connectivity.

A forest can be represented by an undirected graph  $G = (V, E)$ , where the vertices  $V$  correspond to stands or management units, and the edges  $E$  correspond to pairs of adjacent stands.

Let  $G = (V, E)$  be an undirected graph. Consider  $U \subseteq V$ . We define  $G[U]$  to be the graph on  $U$  whose edges are precisely the edges of  $G$  having both endpoints in  $U$ . We say that  $G$  is *connected* if any two of its vertices are linked by a path in  $G$ . We say  $U \subseteq V$  is connected if  $G[U]$  is connected. A set  $U \subseteq V$  is a *connected component* of  $G$  if it is maximally connected (i.e.,  $U \cup \{v\}$  is disconnected for all  $v \in V \setminus U$ ).

There are a number of integer programming formulations for modeling connectivity in graphs. Most formulations use variables associated to the edges. For a comprehensive survey of polyhedral and computational aspects of these formulations see, Magnanti and Wolsey (1995). An example of such approaches applied to forest planning problems is Önal and Briers (2006). Formulations for the Node Weighted (or Prize Collecting) Steiner Tree Problem (Segev 1987) are similar, but consider variables for each node and each edge in the graph. Recent studies of this problem include Cordone and Trubian (2006), da Cunha et al. (2009), Ljubic et al. (2006), Lucena and Resende (2004). We are not aware of any applications of the formulation presented here to forest planning. However, after finishing this research, it was brought to our attention that the same approach had been used in the context of design of sheet metal plates in Mechanical Engineering (Fügenschuh and Fügenschuh 2008). This paper only considers small (30 nodes) single period problems, but includes additional non-linear constraints which make the problems significantly different.

In what follows we propose several new integer programming formulations for modeling node-connectivity. All of the techniques described require defining only a single variable for each node in the graph. We find that defining variables for nodes rather than for arcs is most convenient for forest planning problems because the most important decision unit in such problems is the node (or stand). However, it should be noted that the approaches described next can also be used with the Node Weighted Steiner Tree Problem formulation referred to above.

**Figure 2**  $uv$ -node cut.

The integer programming techniques we describe are mainly based on the notion of node-cut sets, which we define next.

Given nodes  $u, v \in V$  that are non-adjacent ( $\{uv\} \notin E$ ), a set of nodes  $S \subseteq V \setminus \{u, v\}$  is a *node-cut set* separating  $u$  and  $v$  (or simply a  *$uv$ -node cut*) if there is no path between  $u$  and  $v$  in  $G[V \setminus S]$ . It is well-known that given  $U \subseteq V$  and a non-adjacent pair of nodes  $u, v \in U$ , then there exists a path in  $G(U)$  between  $u$  and  $v$  if and only if all  $uv$ -node cuts  $S$  are such that  $S \cap U \neq \emptyset$ . Thus, connectivity of a graph can be characterized by its node-cut sets (see Figure 2).

Before describing the formulations we convene on the following notation.

For  $\{uv\} \notin E$  define

$$\Gamma(u, v) = \{S \subseteq V \setminus \{u, v\} : S \text{ is a minimal } uv\text{-node cut}\}.$$

Consider  $W > 0$ . For each node  $v \in V$  let  $w_v \geq 0$  be an associated weight. Define, for each  $v \in V$  the set

$$\mathcal{C}(v, W) = \{C \subseteq V : v \in C, \sum_{u \in C} w_u < W\}.$$

That is,  $\mathcal{C}(v, W)$  corresponds to the set of all node-sets containing  $v$  but having total weight less than  $W$ . For each set of nodes  $C \subset V$  define the neighborhood of  $C$  as follows,

$$\partial C = \{u \in V \setminus C : \exists v \in C, \{u, v\} \in E\}.$$

That is,  $\partial C$  corresponds to those vertices not in  $C$  that are adjacent to vertices in  $C$ . Note that if  $C \in \mathcal{C}(v, W)$ ,  $\partial C$  is a node-cut set separating  $v$  and each node not in  $C \cup \partial C$ . Finally, for each

node  $v \in V$  define a 0-1 variable  $z_v$  indicating if we are to select node  $v$  or not. Let  $Z$  represent the set of selected nodes. That is,

$$Z = \{v \in V : z_v = 1\}.$$

In what follows we are interested in systems of inequalities on the  $z$  variables that impose connectivity of set  $Z$ . In some cases we will consider the additional inequalities that will be valid when, in addition to imposing that  $Z$  be connected, we impose the condition that  $Z$  meets a minimum weight requirement. That is,

$$\sum_{v \in V} w_v z_v \geq W. \quad (2)$$

- **Unrooted connectivity.** To impose that  $Z$  be connected, the following inequalities suffice,

$$\sum_{w \in S} z_w \geq z_u + z_v - 1, \quad \forall S \in \Gamma(u, v), \forall u, v \in V, \{u, v\} \notin E. \quad (3)$$

Observe that if  $u$  and  $v$  are both selected ( $z_u = z_v = 1$ ), the above constraints force a node of each minimal  $uv$ -node cut  $S$  to be selected as well.

In the case graph  $G$  is disconnected some inequalities (3) will take form  $z_u + z_v \leq 1$  when  $u$  and  $v$  belong to different connected components of  $G$  (in this case the empty set is a node cut separating  $u$  and  $v$ ). If the graph  $G$  is made up of several connected components, say  $V_1, \dots, V_q$ , we may consider a stronger formulation (that is, one with a tighter linear relaxation bound) using the so-called *clique inequalities*

$$\sum_{p=1, \dots, q} z_{u_p} \leq 1, \text{ for every choice of } u_1 \in V_1, \dots, u_q \in V_q.$$

An equivalent way to strengthen the formulation is to consider additional binary variables  $\eta_p = 1$  if the selected connected set is in component  $V_p$  and  $\eta_p = 0$  otherwise, and the constraints

$$z_u \leq \eta_p, \quad \forall u \in V_p, p = 1, \dots, q \quad (4)$$

$$\sum_{p=1, \dots, q} \eta_p \leq 1. \quad (5)$$

The formulation can be further strengthened (in terms of the linear relaxation bound) when constraints (2) are imposed in addition to the connectivity requirements. This can be done using

*ring inequalities*, as described next. The main idea is as follows: any set of nodes  $C$  with weight less than  $W$  is too small to be a feasible set, so at least one of its neighbor nodes must be selected. Formally, given  $W$  and a vertex  $v \in V$ , the *unrooted ring inequalities* are as follows:

$$\sum_{u \in \partial C} z_u \geq z_v, \quad \forall C \in \mathcal{C}(v, W). \quad (6)$$

These inequalities were first proposed by Martins et al. (1999) in the context of multi-patch models (explained next).

- **Rooted connectivity.** To impose that the set of nodes in  $Z$  corresponds to a connected set containing node  $r \in V$  (eg, a root), the following inequalities suffice,

$$\sum_{u \in S} z_u \geq z_v, \quad \forall S \in \Gamma(r, v), \forall v \in V, \{r, v\} \notin E. \quad (7)$$

Observe that if  $v$  is selected ( $z_v = 1$ ), the above constraints force a node of each minimal  $rv$ -node cut  $S$  to be selected as well. This is sufficient to impose that every pair of nodes  $u$  and  $v$  in  $Z$  are linked to each other in  $Z$ . Indeed, if there is a path between  $r$  and  $u$  and a path between  $r$  and  $v$ , then by joining the two paths we obtain a path between  $u$  and  $v$ . Note that these inequalities can be obtained from the unrooted cut inequalities (3) by imposing  $u = r$  and  $z_r = 1$ , thus they are stronger (in terms of the linear programming bound), as commonly noted in the literature on Steiner Tree Problems (Magnanti and Wolsey 1995).

If there is more than one root we simply need to write constraints such as (7) for each pair  $\{r, v\} \notin E$ , where  $r$  is a root and  $v$  is not a root. If  $r_1, r_2$  are two non adjacent roots, we can add the following constraint to strengthen the formulation.

$$\sum_{u \in S} z_u \geq 1, \quad \forall S \in \Gamma(r_1, r_2). \quad (8)$$

As in the unrooted case, if constraints (2) are imposed, the system can be further strengthened (in terms of the linear programming bound) with the following *rooted ring-inequalities*:

$$\sum_{w \in \partial C} z_w \geq 1, \quad \forall C \in \mathcal{C}(r, W). \quad (9)$$

• **Multi-Patch connectivity.** In many applications it does not matter so much if  $Z$  is disconnected, as long as each connected component of  $Z$  meets a minimum size or weight requirement. We call this weaker connectivity requirement a multi-patch connectivity requirement. Let  $W_{patch}$  be the minimum weight required by each connected component in  $Z$ . Note that this weight requirement is different to the weight requirement imposed by (2) in that the former is for each connected set in  $Z$ , as opposed to for all the nodes in  $Z$ . We can meet the multi-patch connectivity requirement in  $Z$  by *not* imposing constraints (3) and imposing the unrooted ring inequalities,

$$\sum_{u \in \partial C} z_u \geq z_v, \quad \forall C \in \mathcal{C}(v, W_{patch}). \quad (10)$$

Suppose there are  $k$  roots  $r_1, \dots, r_k$  which must belong to the selected patches. In this case we must proceed as in the unrooted case, defining constraints (10) for each node, and imposing that  $z_r = 1$  for each root  $r$ . The presence of  $k$  roots can promote but does not guarantee the existence of  $k$  patches, as some roots may belong to the same connected component.

• **Dynamic connectivity.** So far we have only considered static patches. That is, patches that do not change with time. In many forest planning models, as in the one presented in Section 2, it may be possible, or even desired, to relax this condition. For example, in the context of forest harvest scheduling, it is natural to let old-growth-patches replace older old-growth-stands with younger ones as the forest ages. In order to model this it is necessary to consider time dependent variables  $z_{v,t}$  for each  $v \in V$  and  $t \in T$ , so that  $z_{v,t} = 1$  if and only if stand  $v$  is in the patch in time period  $t$ .

When using an unrooted model, consider constraints

$$\sum_{w \in S} z_{w,t} \geq z_{u,t} + z_{v,t} - 1, \quad \forall S \in \Gamma(u, v), \forall u, v \in V, \{u, v\} \notin E, \forall t \in T \quad (11)$$

In case of using a rooted model, use

$$\sum_{u \in S} z_{u,t} \geq z_{v,t}, \quad \forall S \in \Gamma(r, v), \forall v \in V, \{r, v\} \notin E, t \in T \quad (12)$$

Constraints (11) and (12) impose exactly the same cut condition imposed by constraints (3) and (7), but do so in each time period.

In some situations we may want to enforce *temporal connectivity*. Let us assume that, besides connectivity in each period, a minimum area of the patch, say  $A_{temp}$ , must be preserved in the patch from period  $t$  to period  $t + 1$ . Connectivity in each period is expressed by constraints (11) and (12). Here we describe two ways to enforce temporal connectivity. The first one uses additional binary variables, while the second one uses only variables  $z_{vt}$  but requires exponentially many constraints.

The first way is as follows. Consider variables  $\chi_{u,t} = 1$  if the stand  $u$  is in the selected connected component, both in period  $t$  and  $t + 1$ . Hence we may consider constraints

$$\sum_{u \in V} a_u \chi_{u,t} \geq A_{temp}, \quad \forall t \in T \setminus \{N\} \quad (13a)$$

$$\chi_{u,t} \leq z_{u,t}, \quad \chi_{u,t} \leq z_{u,t+1}, \quad \forall u \in V, \forall t \in T \setminus \{N\} \quad (13b)$$

Equations (13a) guarantee the required common area, while (13b) forces  $\chi_{u,t}$  to have the value zero if any of  $z_{u,t}$  or  $z_{u,t+1}$  is zero. Alternatively, the requirement of a common area  $A_{temp}$  can be expressed with  $z$  variables only with the following set of constraints.

$$\sum_{u \in S} a_u z_{u,t} + \sum_{u \in V-S} a_u z_{u,t+1} \geq A_{temp}, \quad \forall S \subseteq V, \forall t \in T \setminus \{N\}. \quad (14)$$

To see this, suppose there is a period  $t < N$  and a set  $S^0 \subseteq V$  such that (14) does not hold. Using (13b) we conclude that (13a) is violated for  $t$ . Conversely, suppose the common area in the connected sets  $S^0 = \{u : z_{u,t} = 1\}$  and  $S^1 = \{u : z_{u,t+1} = 1\}$  is less than  $A_{temp}$ . Let  $S = V - S^0$ . Then  $\sum_{u \in S} a_u z_{u,t} + \sum_{u \in V-S} a_u z_{u,t+1} = \sum_{u \in S^0} a_u z_{u,t+1} = \sum_{u \in S^0 \cap S^1} a_u < A_{temp}$ , so at least one of the constraints (14) is violated.

### 3.2. Solving the linear relaxations in a branch-and-cut algorithm

The formulations proposed here (either rooted or unrooted) encompass, in general, an exponential number of cut inequalities. Unlike in the ARM, these constraints cannot be explicitly enumerated and used directly in a formulation given to an integer programming solver. However, we can add them through a constraint generation procedure. From basic linear programming theory it is known

that only a few constraints are active in a Basic Solution. Hence we need, at least in theory, to add only a reduced number of constraints to solve the linear programming relaxations.

Suppose we want to solve the linear programming relaxation of an integer programming model that includes cut inequalities (3). For inequalities (7) the procedure is similar. A *cutting plane algorithm* works as follows (Nemhauser and Wolsey 1999). Consider an initial LP formulation without cut inequalities, or only a few of them. Let  $z^*$  be the optimal LP solution. Now solve the so-called *separation problem*: check if all constraints (3) are satisfied by  $z^*$ , and if this is not the case find a violated constraint. If all constraints are satisfied by  $z^*$ , then this is the optimal solution of the LP relaxation. Otherwise add the violated constraint to the formulation and solve the new LP problem. Repeat the procedure until all constraints are satisfied. Since the number of constraints is finite, this procedure is finite, as long as the separation problems are solved exactly.

The good news are that the separation problem can be solved efficiently in this case, yielding to a practical constraint generation procedure. Given as solution  $z^*$  of the linear programming relaxation, the separation problem can be stated as an optimization subproblem: find nodes  $u$  and  $v$  such that  $\{u, v\} \notin E$ , and a set  $S^* \in \Gamma(u, v)$  such that the sum  $\sum_{w \in S^*} z_w^*$  is minimum among all sets  $S \in \Gamma(u, v)$ . If  $\sum_{w \in S^*} z_w^* < z_v^* + z_u^* - 1$  the inequality (3) induced by  $u$ ,  $v$  and  $S^*$  is violated; otherwise if  $\sum_{w \in S^*} z_w^* \geq z_v^* + z_u^* - 1$  for every pair of non adjacent nodes  $u$  and  $v$  then the LP solution  $z^*$  is optimal.

In order to find a minimum node cut separating  $u$  and  $v$ , we may use a classical max-flow min-cut theorem (see e.g. Nemhauser and Wolsey 1999): Given a graph with node capacities, the maximum flow between two non adjacent nodes  $u$  and  $v$ , equals the capacity of the minimum capacity node cut separating  $u$  and  $v$ . In our case the node capacities are the values of the linear programming variables  $z_w^*$ . In practice we transform the graph in a way such that each node is replaced by two arcs, and use an efficient max flow algorithm to determine a minimum cut.

We can repeat this cutting plane procedure to solve the linear programming relaxations at node of a branch-and-bound procedure to obtain a branch-and-cut algorithm that will solve the whole connectivity constrained integer programming formulation. We note that in practice we

may not run the complete cutting plane procedure in each node of the branch-and-bound tree. While the optimal solution to the LP relaxation obtained after early termination of the cutting plane procedure might violate some of the constraints, its value will certainly provide an upper bound to the LP relaxation with all cut inequalities (this early termination solution is optimal to a maximization problem with fewer constraints). If this upper bound is accurate enough it might be enough to fathom the corresponding branch-and-bound node and if it is not we could still branch and continue the cutting plane procedure on the child nodes. Most practical branch-and-cut algorithms carefully monitor the upper bounds provided by the cutting plane procedure and dynamically decide if it should be terminated. A simple version of this monitoring that is used in our implementation is described in Section 4.1.

#### 4. Application to an Example

In this section we test our proposed branch-and-cut algorithm on the harvest scheduling model described in section 2. From the discussion in Section 3 we see that connectivity requirement (1k) in this problem is a single patch multi-period un-rooted dynamic connectivity requirement. As noted in Section 3 un-rooted connectivity requirements are usually harder than rooted requirements. For this reason we will also consider a rooted variant of problem (1). Similarly, dynamic models are usually harder than static multi-period requirements, so we also consider a static variant of problem (1). This static version is obtained by dropping the  $t$  index from the  $z_{v,t}$  variables and results in (1b)–(1d) becoming

$$z_v + \sum_{t \in T} y_{v,t} \leq 1, \quad \forall v \in V \quad (15a)$$

$$\sum_{v \in V} a_v z_v \geq A_{min} \quad (15b)$$

$$z_v = 0, \quad \forall v \text{ such that } b_v < O_{age} \quad (15c)$$

With these two modifications we obtain the following four variants:

1. Rooted-Static: (1) with  $z_v$  instead of  $z_{v,t}$  and (1k) replaced by (7).
2. Rooted-Dynamic: (1) with (1k) replaced by (12).

3. Unrooted-Static: (1) with  $z_v$  instead of  $z_{v,t}$  and (1k) replaced by (3).
4. Unrooted-Dynamic: (1) with (1k) replaced by (11).

For the rooted instances for each forest we consider three randomly selected roots that satisfy the minimum age requirements.

#### 4.1. Implementation Details

We implemented our branch-and-cut algorithm in C++ using the CPLEX 11.0 callable library CPLEX (2007). For simplicity we describe the details of this cutting plane algorithm only for the unrooted static version of the problem. The details for other versions are analogous. Let  $z^*$  denote the vector of old-growth variables in the linear relaxation solution. Consider pairs of variables  $z_u, z_v$  such that  $z_u^* + z_v^* > 1$ . Then one or two minimum  $uv$ -node cuts are obtained; if the corresponding inequalities are violated by the current linear programming solution, they are added to the formulation. In order to determine a minimum cut, the graph is transformed into a digraph where each node  $i$  is replaced by two opposite directed arcs with capacity  $z_i^*$  each. Then a maximum flow from  $u$  to  $v$  is obtained in the modified graph, using an implementation of the push-relabel algorithm available in ELib (Espinoza and Goycoolea 2003). This algorithm also provides a minimum capacity cut separating  $u$  from  $v$ . In some situations more than one cut may exist. In this case, the cut obtained is the one “closer” to node  $u$ . Hence we also compute a maximum flow from  $v$  to  $u$ , which may provide a different minimum cut. When a minimum  $uv$  node cut is obtained, it is checked whether the corresponding inequality can be strengthened to a ring inequality, as mentioned in Section 3. We also implemented a fast breath first search algorithm that finds a ring cut or a  $uv$ -node cut if the set  $\{i \in V : z_i^* > 0\}$  is not connected. This separation heuristic solves the separation problem exactly if the solution is integer. It is used prior to the max flow-min cut algorithm. In preliminary tests, we discovered a large *tail-off* behavior in the effectiveness of the connectivity cuts, translating in very little changes of the upper bound after each round of adding such constraints. To counteract this effect we carry out a selective separation, adding cuts in the node zero while the change in the value of the relaxation is large enough. If the value of the LP

relaxation does not change in more than 0.5% during 10 consecutive rounds of separation, we stop adding cuts. If an integer solution is found, we check if it complies with the connectivity requirements, in which case we have a feasible solution for the problem, otherwise we add violated constraints to the problem. We do this both for solutions found at integral branch-and-bound nodes and those found by CPLEX’s heuristics.

Preliminary tests also showed that CPLEX had trouble finding feasible solutions for some instances beyond the rooted static case. For this reason we ran the problem variants in an order such that feasible solutions for previous runs could be used as heuristic solutions for subsequent runs. Specifically, we ran the variants in the order: 1) static-rooted 2) static-unrooted 3) dynamic-rooted 4) dynamic-unrooted. Then we feed the best solution found in step 1) as a heuristic solution to step 2) and the best solution between step 2) and 3) as a heuristic to step 4).

## 4.2. Description of Forests

We consider six instances in this study. The first one is the hypothetical forest considered in Rebain and McDill (2003b) which we denote by *Rebain-McDill*. The others are five instances obtained from the repository of forestry instances of the Forest Management Optimization Site (FMOS) (2008): Gavin, Hardwicke, FLG9A, Shulkell and El Dorado. Tables 2 and 3 include some information on the instances considered and the parameters selected for formulation (1) and its variants. For all instances we considered three periods in our planning horizon.

**Table 2** Forest instances information.

Name	Stands	Area (ha)	$A_{max}$ (ha)
Rebain-McDill	50	1000	40
Gavin	352	6310	40
Hardwicke	423	6948	40
FLG9A	850	9999	48.6
Shulkell	1039	4498.7	16
El Dorado	1363	21147	48.5

**Table 3** Global parameters used for the ARM formulation.

Name	Value
$L$	15%
$U$	15%
$H$	40 Years
$O_{age}$	60 Years
$A_{min}$	20% of total area

### 4.3. Computational Results

The main objective of the tests is to determine the ability of the proposed approach to obtain good solutions within an reasonable amount of time. The main measure of the level of success when solving the different test instances presented along this paper is the gap between upper and lower bounds for their optimal solution value. Given upper and lower bounds  $U_{bound}$  and  $L_{bound}$  on the optimal value, the relative gap between them is given by  $gap = 100 \times (U_{bound} - L_{bound}) \div L_{bound}$ , which is an upper bound of the true gap. The tests were performed in a Computer Xeon Quad-core with 32 Gb RAM. The running stop criteria was a maximum time of 4 hours or the finding of a feasible solution within a 0.01% gap.

In Table 4 we present the results for each instance and problem variant showing i) the final integer programming gap, between the best upper bound and the best feasible solution found after 4 hours, or the time to attain an optimal solution (with a 0.01% gap), if the problem is solved within the time limit (column “Final IP gap”); ii) the gap between the upper bound of the linear programming relaxation obtained at the root branch-and-bound node and, the best known feasible solution (column “Zero LP gap”); iii) the time to solve the linear programming relaxation at the node zero of the branch-and-bound tree (column “Zero LP time”).

From this table, we fist observe that using our proposed formulations we are able to obtain near-optimal ( $< 1\%$  GAP) solutions for most instances and variants in the allotted time. Furthermore, even though some of the instances are all relatively large (up to 1363 stands) with regard to similar problems solved in the literature we can find good ( $< 9\%$  GAP) solutions for all the instances and variants. This illustrates the effectiveness of our proposed branch-and-cut algorithm. As we

**Table 4** Results for the IP model.

Instance	Root	Static			Dynamic		
		Final IP gap	Zero LP gap	Zero LP time (s)	Final IP gap	Zero LP gap	Zero LP time (s)
Rebain-McDill	R1	[0.58 s]	0.42%	0.09	[2.93 s]	0.77%	0.71
	R2	[1.93 s]	0.65%	0.09	[136.75 s]	1.87%	0.50
	R3	[0.12 s]	0.37%	0.06	[9.94 s]	0.79%	0.70
	Unrooted	[11.46 s]	0.49%	0.62	[6746.89s]	2.93%	0.70
Gavin	R1	0.01%	0.28%	5.03	7.29%	7.45%	17.59
	R2	0.09%	0.38%	4.25	6.42%	6.79%	12.43
	R3	0.02%	0.33%	7.08	5.47%	5.93%	16.57
	Unrooted	0.61%	0.84%	16.85	6.68%	7.05%	39.80
Hardwicke	R1	0.15%	0.23%	5.10	3.85%	4.06%	16.67
	R2	0.26%	0.38%	6.43	2.75%	2.88%	30.73
	R3	0.23%	0.39%	6.52	1.64%	1.75%	24.42
	Unrooted	0.41%	0.51%	16.52	2.32%	2.55%	49.64
FLG9A	R1	0.46%	2.20%	305.57	5.93%	6.03%	1219.25
	R2	0.12%	2.14%	161.67	5.94%	6.12%	780.52
	R3	0.05%	2.05%	137.51	5.91%	6.02%	1143.86
	Unrooted	3.19%	3.85%	393.22	7.62%	8.36%	591.41
Shulkell	R1	0.06%	0.19%	18.30	0.10%	0.23%	76.30
	R2	0.06%	0.18%	24.14	0.08%	0.22%	18.12
	R3	0.05%	0.17%	17.14	0.08%	0.23%	18.64
	Unrooted	0.05%	0.27%	81.65	0.55%	0.63%	175.09
El Dorado	R1	0.05%	0.08%	78.60	0.08%	0.08%	66.69
	R2	0.07%	0.10%	169.92	0.12%	0.14%	78.40
	R3	0.07%	0.11%	33.53	0.11%	0.11%	89.85
	Unrooted	0.14%	0.14%	502.76	0.14%	0.14%	1174.93

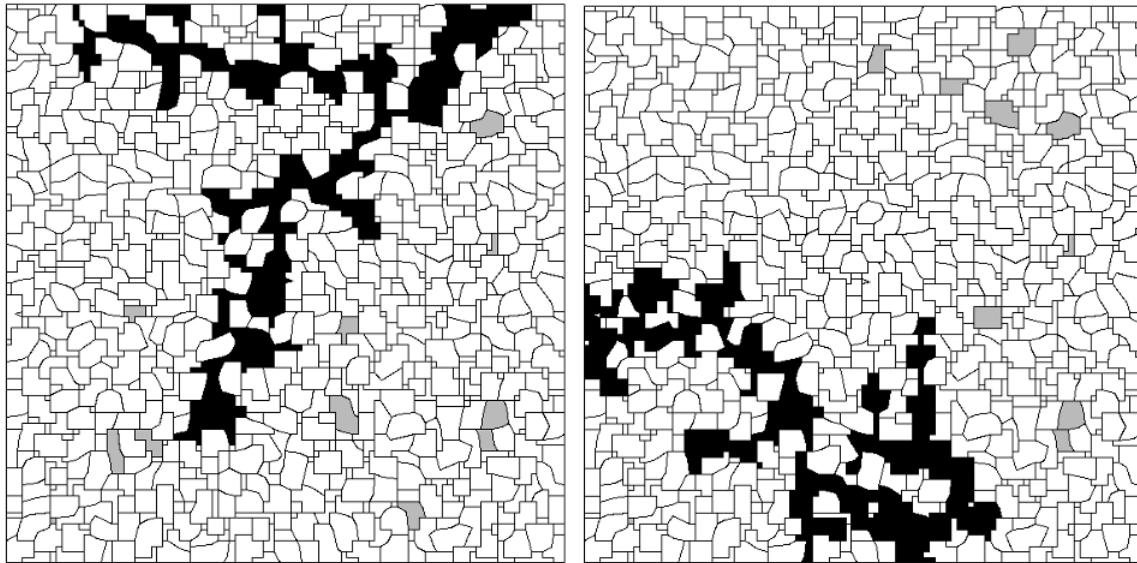
mentioned before, this means a considerable improvement in terms of the size of the instances solved, on previous results in the literature and other forestry applications that involve connectivity.

A second observation is that the node zero LP values are usually very tight and can be obtained quite fast (always in less than 21 minutes and often much faster). This suggests that coupling this approach with specialized heuristics could result in an even faster algorithm. Finally, we see that there is an important difficulty variability among the instances and variants. With respect to the variants it is clear that rooted versions are easier than unrooted ones and static versions are also easier than dynamic versions. With respect to instances the picture is less clear and, as expected, the difficulty is not always due to size as the hardest instances (FLG9A and Gavin) are not the largest.

## 5. Final remarks

In this article we have described a new integer programming formulation for modeling node connectivity in graphs and we have shown how it can be used to model connectivity requirements arising in forest planning models. This formulation has been tested solving a harvest scheduling model with maximum clearcut constraints and old-growth connectivity requirements. The tests were performed with four variants of connectivity (combinations of rooted/unrooted and static/dynamic) on five real forest instances of medium size (352 - 1363 stands). Our computations show that the formulation provides strong linear programming upper bounds ( $< 9\%$  gap in all instances) in very reasonable time ( $< 4$  hours).

The evidence that we obtain very tight linear programming relaxations, suggests that specialized rounded heuristics could be an effective way of obtaining near-optimal solutions in practice. Moreover, it suggests that the use of this formulation could be of practical use evaluating the performance of different heuristics by testing them on medium sized instances.



**Figure 3** Two solutions for FLG9A. Stands in black are the ones selected for the old-growth forest. For simplicity stands that are harvested in some period are showed in white and nonharvested stands in gray.

Visual inspection of the solutions obtained (see Figure 3) reaffirm the preliminary conclusions

of Rebain and McDill (2003a) obtained on 50 stand and single period instances: Namely, that promoting connectivity alone leads to old-growth patches that have long and narrow shapes. This suggests that adapting this formulation to promote large interior area or higher perimeter-area ratios might be a valuable research direction.

## Acknowledgments

This work was partially funded by FONDECYT grants 11075028, 1085188, and 1110674, CONICYT ANILLO grant ACT-88, Basal project CMM (Universidad de Chile), the Millenium Institute on Complex Systems (Universidad de Chile) and the Fundação para a Ciência e a Tecnologia, under the project PEST-OE/MAT/UI0152.

## References

2006. Forest Stewardship Council standard. <http://www.fsc.org/>.
2008. FMOS - forest management optimization site. <http://ifmlab.for.unb.ca/fmos/>.
2010. Sustainable Forestry Initiative standard. <http://www.sfiprogram.org>.
- Aldrich, M., A. Belokurov, J. Bowling, N. Dudley, C. Elliott, L. Higgins-Zogib, J. Hurd, L. Lacerda, S. Mansourian, T. McShane, D. Pollard, J. Sayer, K. Schuyt. 2004. Integrating forest protection, management and restoration at a landscape scale. Tech. rep., World Wide Fund for Nature.
- Barrett, T. M., J. K. Gilles, L. S. Davis. 1998. Economic and fragmentation effects of clearcut restrictions. *Forest Sci.* **44**(4) 569–577.
- Caro, F., M. Constantino, I. Martins, A. Weintraub. 2003. A 2-opt tabu search procedure for the multiperiod forest harvesting problem with adjacency, greenup, old growth, and even flow constraints. *Forest Sci.* **49**(5) 738–751.
- Constantino, M., I. Martins, J. Borges. 2008. A new mixed integer programming model for harvest scheduling subject to maximum area restrictions. *Oper. Res.* **56**(3) 542–551.
- Cordone, R., M. Trubian. 2006. An exact algorithm for the node weighted steiner tree problem. *4OR* **4**(2) 124–144.
- CPLEX, IBM ILOG. 2007. <http://www.ibm.com/software/integration/optimization/cplex/>.

- da Cunha, A. S., A. Lucena, N. Maculan, M. G. C. Resende. 2009. A relax-and-cut algorithm for the prize-collecting steiner problem in graphs. *Discrete Appl. Math.* **157**(6) 1198–1217.
- Earn, D. J. D., S. A. Levin, P. Rohani. 2000. Coherence and conservation. *Science* **290**(5495) 1360–1364.
- Espinoza, D., M. Goycoolea. 2003. EGlib, efficient general library. [http://www.dii.uchile.cl/~daespino/EGlib\\_doc/main.html](http://www.dii.uchile.cl/~daespino/EGlib_doc/main.html).
- Etienne, R. S., J. A. P. Heesterbeek. 2000. On optimal size and number of reserves for metapopulation persistence. *J. Theor. Biol.* **203**(1) 33–50.
- Fügenschuh, Armin, Marzena Fügenschuh. 2008. Integer linear programming models for topology optimization in sheet metal design. *Mathematical Methods of Operations Research* **68**(2) 313–331. doi:10.1007/s00186-008-0223-z. URL <http://www.springerlink.com/index/10.1007/s00186-008-0223-z>.
- Goycoolea, M., A. T. Murray, F. Barahona, R. Epstein, A. Weintraub. 2005. Harvest scheduling subject to maximum area restrictions: Exploring exact approaches. *Oper. Res.* **53**(3) 490–500.
- Goycoolea, M., A. T. Murray, J. P. Vielma, A. Weintraub. 2009. Evaluating approaches for solving the area restriction model in harvest scheduling. *Forest Sci.* **55**(2) 149–165.
- Gustafson, E. J., T. R. Crow. 1998. Simulating spatial and temporal context of forest management using hypothetical landscapes. *Environmental Management* **22**(5) 777–787.
- Keith, H., B. Mackey, D. Lindenmayer. 2009. Re-evaluation of forest biomass carbon stocks and lessons from the world’s most carbon-dense forests. *Proc. Natl. Acad. Sci.* **106**(28) 11635–11640.
- Kirpatric, J. B. 1983. An iterative method for establishing priorities for the selection of nature reserves: An example from tasmania. *Biological Conservation* **25**(2) 127–134.
- Ljubic, I., R. Weiskircher, U. Pfersch, G. Klau, P. Mutzel, M. Fischetti. 2006. An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. *Math. Programming* **105**(2-3) 427–449.
- Lucena, A., M. G. C. Resende. 2004. Strong lower bounds for the prize collecting Steiner problem in graphs. *Discrete Appl. Math.* **141**(1-3) 277–294.
- Luyssaert, S., E. D. Schulze, A. Borne, A. Knohl, D. Hessenmoller, B. E. Law, P. Ciais, J. Grace. 2008. Old-growth forests as global carbon sinks. *Nature* **455** 213–215.
- Magnanti, T. L., L. A. Wolsey. 1995. *Optimal trees, Handbooks in Operational Research and Management Science*, vol. 7. North-Holland, Amsterdam, 503–615.

- Martins, I., J. Borges, M. Constantino. 1999. Forest management models with spatial structure constraints. Working paper Centro IO, University of Lisbon.
- Martins, I., M. Constantino, J. G. Borges. 2005. A column generation approach for solving a non-temporal forest harvest model with spatial structure constraints. *Eur. J. Oper. Res.* **161**(2) 478–498.
- McDill, M. E., S. A. Rebaun, J. Braze. 2002. Harvest scheduling with area-based adjacency constraints. *Forest Sci.* **48**(4) 631–642.
- Murray, A. T. 1999. Spatial restrictions in harvest scheduling. *Forest Sci.* **45**(1) 45–52.
- Nemhauser, G. L., L. A. Wolsey. 1999. *Integer and Combinatorial Optimization*. Discrete Mathematics and Optimization, Wiley-Interscience.
- Öhman, K. 2000. Creating continuous areas of old forest in long-term forest planning. *Canadian J. Forest Res.* **30**(11) 1817–1823.
- Öhman, K., T. Lämås. 2005. Reducing forest fragmentation in long-term forest planning by using the shape index. *Forest Ecology and Management* **212**(1-3) 346–357.
- Öhman, K., P. Wikström. 2008. Incorporating aspects of habitat fragmentation into long-term forest planning using mixed integer programming. *Forest Ecology and Management* **255**(3-4) 440–446.
- Önal, H., R. A. Briers. 2002. Incorporating spatial criteria in optimum reserve network selection. *Proc. R. Soc. Lond. B* **269**(1508) 2437–2441.
- Önal, H., R. A. Briers. 2003. Selection of a minimum boundary reserve network using integer programming. *Proc. R. Soc. Lond. B* **270**(1523) 1487–1491.
- Önal, H., R. A. Briers. 2006. Optimal selection of a connected reserve network. *Oper. Res.* **54**(2) 379–388.
- Rebaun, S., M. E. McDill. 2003a. Can mature patch constraints mitigate the fragmenting effects of harvest opening size restrictions? *Internat. Trans. Oper. Res.* **10**(5) 499–513.
- Rebaun, S., M. E. McDill. 2003b. A mixed-integer formulation of the minimum patch size problem. *Forest Sci.* **49**(4) 608–618.
- Segev, A. 1987. The node-weighted Steiner tree problems. *Networks* **17**(1) 1–17.
- Sessions, J. 1992. Solving for habitat connections as a Steiner network problem. *Forest Sci.* **38**(1) 203–207.
- Thompson, E. F., B. G. Halterman, T. S. Lyon, R. L. Miller. 1973. Integrating timber and wildlife management planning. *Forestry Chronicle* **47**(49) 247–250.

- Tóth, S. F. 2005. Modeling timber and non-timber tradeoffs in spatially-explicit forest planning. Ph.D. thesis, Forest Resources and Operations Research. Penn State University.
- Tóth, S. F., M. E. McDill. 2008. Promoting large, compact mature forest patches in harvest scheduling models. *Environ. Model. and Assess.* **13**(1) 1–15.
- Tóth, S. F., M. E. McDill, S. Rebaun. 2006. Finding the efficient frontier of a bi-criteria, spatially explicit, harvest scheduling problem. *Forest Sci.* **52**(1) 93–107.
- Vielma, J. P., A. T. Murray, D. M. Ryan, A. Weintraub. 2007. Improving computational capabilities for addressing volume constraints in forest harvest scheduling problems. *Eur. J. Oper. Res.* **176**(2) 1246–1264.
- Williams, J. C. 1998. Delineating protected wildlife corridors with multi-objective programming. *Environ. Model. Assess.* **3**(1-2) 77–86.
- Williams, J. C., C. S. ReVelle. 1998. Reserve assemblage of critical areas: A zero-one programming approach. *Eur. J. Oper. Res.* **104**(3) 497–509.
- Williams, J. C., C. S. ReVelle, S. A. Levin. 2005. Spatial attributes and reserve design models: A review. *Environ. Model. Assess.* **10**(3) 163–181.