Finding Efficient Harvest Schedules under Three Conflicting Objectives

Abstract: Public forests have many conflicting uses. Designing forest management schemes that provide the public with an optimal bundle of benefits is therefore a major challenge. Although a capability to quantify and visualize the tradeoffs between the competing objectives can be very useful for decision makers, developing this capability presents unique difficulties if three or more conflicting objectives are present and the solution alternatives are discrete. This study extends four multi-objective programming methods that generate spatially-explicit forest management alternatives that are efficient (non-dominated) with respect to three or more competing objectives. The algorithms were applied to a hypothetical forest planning problem with three timber and wildlife-related objectives. While the ε-Constraining and the proposed Alpha-Delta Method found a larger number of efficient alternatives, the Modified Weighted and the Tchebycheff methods provided a better overall estimation of the timber and non-timber tradeoffs associated with the test problem. Additionally, the former two methods allowed a greater degree of user control and are easier to generalize to *n*-objective problems.

Keywords: multi-objective forest planning, spatial optimization, integer programming, tradeoffs

Introduction

Management planning problems with conflicting objectives occur frequently in forestry. The public expects more from forest resources than merely timber production, including watershed protection, wildlife habitat management, aesthetics, recreation and carbon sequestration. Stakeholder groups such as the timber industry and environmental organizations often hold strongly conflicting values related to these uses, and conflicts between timber and non-timber objectives are common. Harvesting can fragment sensitive habitats, obstruct the movement of wildlife, increase fire risk, and reduce the aesthetic value of the forest. On the other hand, eliminating timber production from public forests in industrialized nations, as many suggest, would only increase the pressure on forest resources in less developed nations, where environmental controls may be less effective (Thomas 2000). Moreover, other uses, such as recreation, can also stress forest ecosystems. However, the tradeoffs between conflicting goals can often be balanced effectively within a landscape or forest through careful planning (Rosenbaum 2000). In most cases, quantifying the tradeoffs to determine the degree of incompatibility between competing forest uses can help decision makers (DM) select the best compromise management alternatives.

The spatial layout of forestry operations such as harvesting or road construction can have a profound impact on many non-timber objectives. Spatially-explicit forest management planning models are useful for efficiently designing the location and timing of these operations while also addressing wildlife habitat concerns (e.g., Rebain and McDill 2003a, 2003b). These models, usually formulated as integer programs (IPs), are used to determine when specific harvest units should be cut and when and where other site-specific management interventions should be performed in order to balance various forest uses. Most often, these models have been formulated as single objective problems, where one forest use is optimized subject to a range of restrictions (e.g., Leuschner et al. 1975; Mealey and Horn 1981; Cox and Sullivan 1995; Bettinger et al. 1997; Rebain and McDill 2003a, 2003b). Some of these restrictions ensure that minimum requirements on both timber and non-timber objectives are met. One example would be to maximize timber output or the discounted net revenues from a forest subject to constraints requiring a balanced ending age-class distribution, a smooth flow of timber production over time, and maintaining a minimum amount of mature forest habitat in large compact patches while never exceeding a maximum harvest opening size. Alternatively, the amount of mature forest habitat could be maximized subject to minimum net present value or minimum timber output constraints.

In these types of formulations, the DM(s) is required to specify the minimum requirements on some forest uses prior to the optimization process. Defining harvest targets might be relatively straightforward, but setting limits on the amount of mature forest habitat patches, for instance, might not.

Without knowing what habitat requirements are feasible and which are too modest, specifying such limits is typically guesswork and can lead to poor decisions. Furthermore, better decisions can be made if the DM(s) understand the tradeoff structure between competing objectives before setting requirements on various objectives. Another frequently used approach, goal programming (Charnes et al. 1955), does not completely overcome this problem either, as it requires the DM to set up targets on the objectives that, unlike constraints, do not have to be met. Goal programs (GP) minimize the deviations from the targets in either an order of preference (preemptive GP) or in line with a set of weights assigned to each objective (non-preemptive GP). Either way, the DM has to specify both the targets and the weights or a preference list for the objectives prior to formulating the model.

When possible, generating and visualizing the complete – or an effectively filtered – set of *efficient* (a.k.a., *Pareto-optimal*, Pareto 1909) solutions to forest planning problems should help the DM(s) acquire a holistic view of the problem and enable a more informed decision when selecting a best compromise management alternative. As opposed to dominated solutions, efficient alternatives are those whose objective achievements cannot be further improved without compromising at least one objective. This unique set of solutions defines the so-called *efficient frontier*. Studying this frontier can be valuable to the DM(s) for two reasons. First, the efficient frontier separates the region where additional solutions do not exist from the region where dominated solutions might exist (Tóth et al. 2006). Thus, the DM can assess the limits of simultaneously achieving several conflicting objectives. In other words, the efficient frontier answers the question: what is possible? Second, by moving along this frontier (i.e., by moving between Pareto-optimal solutions), one can also assess the amount of one objective that must be forgone in order to achieve a given increase in the amount of another objective. Thus, the efficient frontier identifies the structure of the tradeoffs between the competing objectives represented by the axes of the efficient frontier space.

Since in spatial forest planning many management decisions are binary, such as whether or not a certain forest unit with predefined boundaries should be cut within a given time interval, the set of feasible model solutions is discrete (Tóth et al. 2006). As a result, the set of attainable objective function values, and hence the efficient frontier itself, is also discrete, and therefore non-convex. This property makes it computationally challenging to generate the efficient frontiers for spatially-explicit forest management planning problems.

Tóth et al. (2006) evaluated and tested four traditional methods of generating the efficient frontier for a bi-objective spatially-explicit forest management planning problem. The four approaches, (1) the Weighted Objective Function method (Geoffrion 1968), (2) the ε-Constraining method (Haimes et al. 1971), (3) the Decomposition method based on the Tchebycheff-metric (Eswaran et al. 1989), and (4) the Triangles method (Chalmet et al. 1986), were tested on a 50-management unit hypothetical forest planning problem. Tóth et al. (2006) also proposed a new approach called the "Alpha-Delta" method that performed well compared to the other approaches. The two objectives of interest in their test case were to maximize the net present value (NPV) of the forest and to maximize the minimum amount of mature forest habitat in large patches over the planning horizon.

Most forest planning problems involve more than two competing objectives. A limited number of theoretical studies on generating the set of efficient alternatives for three- or more objective IPs have been documented. One primary area of research has been the family of the so-called reference point methods (Ehrgott and Wiecek 2005). The concept is simple: an efficient solution can be found by minimizing the distance between a reference point, which can be any unattainable solution in the objective space, such as the *ideal solution* or goal programming targets, and potential non-dominated solutions (the ideal solution is a vector of objective function values that are gained by optimizing the problem for each objective, one at a time, without regard to the rest of the objectives). The distance measure that is most often used is the weighted (or not) Tchebycheff-metric and its variants. Different efficient solutions can be found by changing the weights on the distance metric (e.g., Eswaran et al. 1989). Moreover, the reference points themselves can be varied to identify other solutions (e.g., Alves and Clímaco 2001). However, due to the discrete nature of integer programming, different weight combinations and different reference points can also lead to identical solutions. Thus, a “smart” decomposition of the weight space (or the reference point space) into regions that lead to the same solutions is needed to reduce the time spent finding redundant solutions. These regions are called *indifferent sets* and are convex in the weight space and non-convex in the reference point space (Alves and Clímaco 2001). The reference point methods find efficient solutions by determining or approximating these indifference sets.

An algorithm proposed by Chalmet et al. (1986) finds $|Z|$ efficient solutions with respect to n objectives by solving at least $n|Z|+1$ IPs. The efficiency of this approach is questionable given that the ε-Constraining method (Sadagopan and Ravindran 1982), which appeared to be slow in the bi-objective case (Tóth et al. 2006), finds the same number of solutions by solving only $n|Z|$ IPs.

Case studies assessing the numerical and computational performance of the available multi-objective methods as applied to larger than “illustrative” problem instances are not common. This is particularly true for the area of forest resources and wildlife management. Although efficient frontiers with respect to two objectives have been studied in (Roise et al. 1990; Holland et al. 1994; Cox and Sullivan 1995; Arthaud and Rose 1996; Church et al. 1996; Snyder and ReVelle 1997; Williams 1998; Church et al. 2000; Richards and Gunn 2000), the efficient frontiers of discrete forest management decision problems with three or more objectives have apparently not been researched. Generalizing algorithms that work well for bi-objective problems to problems with three or more objectives is challenging because of the added mathematical and computational complexity. Nevertheless, this task is important as few forest planning problems involve only one or two competing management objectives.

The present study builds on Tóth et al. (2006) by extending three of the four traditional approaches, the Weighted Objective Function method, the ε-Constraining method, and the

Decomposition method based on the Tchebycheff-metric, and the proposed Alpha-Delta method to handle three or more objectives. The same 50-management unit hypothetical forest planning problem used in Tóth et al. (2006) is used to demonstrate and evaluate the mechanics of the extended methods in generating the efficient frontier of a tri-objective spatial forest planning problem. Two of the objectives are the same as in Tóth et al. (2006): (1) maximize discounted net timber revenues, and (2) maximize the minimum area of mature forest habitat that evolves in large patches over the planning horizon. The third is to minimize the total perimeter of the patches (summed over all periods). The formulation of this objective was introduced in Tóth and McDill (2008). Maximizing the area of mature forest habitat patches while minimizing their total perimeter promotes forested landscapes with several desirable spatial characteristics. This approach fosters the development of patches with low perimeter-area ratios, increases their temporal overlap, and results in fewer and larger patches.

The Model Formulation

This section describes a tri-criteria integer programming formulation that (1) maximizes the net present value of the forest, (2) maximizes the minimum amount of mature forest habitat in large patches over the planning horizon, and (3) minimizes the total length of the edges of these patches over the planning horizon. The model includes harvest flow constraints, maximum harvest opening size constraints, constraints that define the minimum area of mature forest habitat patches, and an average ending age constraint. The formulation of the mature forest patch criterion is a slightly modified version of the one presented in Rebain and McDill (2003b). Formulation of the maximum harvest area constraints is a generalization of the formulation presented in McDill et al. (2002).

$$Max\, Z = \sum_{m=1}^{M} A_m [c_{m0} X_{m0} + \sum_{t=h_m}^{T} c_{mt} X_{mt}] \qquad (1)$$

$$Max\ \lambda \qquad (2)$$

$$Min \sum_{t=1}^{T} \mu_t \tag{3}$$

subject to:

$$X_{m0} + \sum_{t=h_m}^{T} X_{mt} \leq 1 \qquad \text{for } m = 1, 2, \ldots M \tag{4}$$

$$\sum_{m \in M_{ht}} v_{mt} \cdot A_m \cdot X_{mt} - V_t = 0 \qquad \text{for } t = 1, 2, \ldots T \tag{5}$$

$$b_{lt} V_t - V_{t+1} \leq 0 \qquad \text{for } t = 1, 2, \ldots T\text{-}1 \tag{6}$$

$$-b_{lt} V_t + V_{t+1} \leq 0 \qquad \text{for } t = 1, 2, \ldots T\text{-}1 \tag{7}$$

$$\sum_{m \in M_p} X_{mt} \leq n_{P_i} - 1 \qquad \text{for all } p \in P \text{ and } t = h_i, \ldots, T \tag{8}$$

$$\sum_{j \in J_{mt}} X_{mj} - O_{mt} \geq 0 \qquad \text{for } m = 1, 2, \ldots M, \text{ and } t = 1, 2, \ldots, T \tag{9}$$

$$\sum_{j \in J_{mt}} X_{mj} - \left| J_{mt} \right| O_{mt} \leq 0 \qquad \text{for } m = 1, 2, \ldots M, \text{ and } t = 1, 2, \ldots, T \tag{10}$$

$$\sum_{m \in M_c} O_{mt} - n_c B_{ct} \geq 0 \qquad \text{for } c \in C, \text{ and } t = 1, 2, \ldots, T \tag{11}$$

$$\sum_{m \in M_c} O_{mt} - B_{ct} \leq n_c - 1 \qquad \text{for } c \in C, \text{ and } t = 1, 2, \ldots, T \tag{12}$$

$$\sum_{c \in C_m} B_{ct} - BO_{mt} \geq 0 \qquad \text{for } m = 1, 2, \ldots M, \text{ and } t = 1, 2, \ldots, T \tag{13}$$

$$\sum_{c \in C_m} B_{ct} - \left| C_m \right| BO_{mt} \leq 0 \qquad \text{for } m = 1, 2, \ldots M, \text{ and } t = 1, 2, \ldots, T \tag{14}$$

$$\sum_{m=1}^{M} A_m BO_{mt} \geq \lambda \qquad \text{for } t = 1, 2, \ldots T \tag{15}$$

$$\sum_{m=1}^{M} P_m BO_{mt} - 2 \sum_{pq=1}^{N} CB_{pq} \Omega^t{}_{pq} = \mu_t \qquad \text{for } t = 1, 2, \ldots T \tag{16}$$

$$BO_{pt} + BO_{qt} - 2\Omega^t{}_{pq} \geq 0 \qquad \text{for } t = 1, 2, \ldots T,\ pq = 1, 2, \ldots N \tag{17}$$

$$BO_{pt} + BO_{qt} - \Omega^{t}{}_{pq} \leq 1 \qquad \text{for } t = 1,2,\ldots T,\ pq = 1,\ 2,\ \ldots N \qquad (18)$$

$$\sum_{m=1}^{M} A_m[(Age^T_{m0} - \overline{Age}^T)X_{m0} + \sum_{t=h_m}^{T}(Age^T_{mt} - \overline{Age}^T)X_{mt}] \geq 0 \qquad (19)$$

$$X_{mt} \in \{0,1\} \qquad \text{for } m = 1,\ 2,\ \ldots M, \text{ and } t = 0,\ h_m,\ h_m + 1,\ldots,T \qquad (20)$$

$$B_{ct} \in \{0,1\} \qquad \text{for } c \in C,\ \ t = 1,\ 2,\ \ldots,T \qquad (21)$$

$$O_{mt},\ BO_{mt} \in \{0,1\} \qquad \text{for } m = 1,\ 2,\ \ldots M, \text{ and } t = 0,\ 1,\ldots,T \qquad (22)$$

$$\Omega^t_{pq} \in \{0,1\} \qquad \text{for } pq \in N \qquad (23)$$

where the decision variable is:

X_{mt} = a binary variable whose value is 1 if management unit *m* is to be harvested in period *t* for $t = h_m, h_{m+1},\ldots,T$; when $t = 0$, the value of the binary variable is 1 if management unit *m* is not harvested at all during the planning horizon (i.e., X_{m0} represents the "do-nothing" alternative for management unit *m*);

the auxiliary and accounting variables are:

O_{mt} = a binary variable whose value may equal 1 if management unit *m* meets the minimum age requirement for mature patches in period *t*, i.e., the management unit is old enough to be part of a mature patch;

B_{ct} = a binary variable whose value is 1 if all of the management units in cluster *c* meet the minimum age requirement for mature patches in period *t*, i.e., the cluster is part of a mature patch;

BO_{mt} = a binary variable whose value is 1 if management unit *m* is part of a cluster that meets the minimum age requirement for large mature patches, i.e., the management unit is part of a patch that is big enough and old enough to constitute a large, mature patch;

Ω^t_{pq} = a binary variable whose value is 1 if adjacent management units p and q are both part of a cluster that meets the minimum age requirement for large mature patches in period t;

V_t = a continuous variable indicating the total volume of sawtimber in m^3 harvested in period t;

μ_t = the total perimeter of mature forest habitat patches in period t; and

the parameters are:

h_m = the first period in which management unit m is old enough to be harvested;

λ = the minimum area of mature forest habitat patch over all periods;

M = the number of management units in the forest;

N = the number of pairs of management units in the forest that are adjacent;

T = the number of periods in the planning horizon;

c_{mt} = the discounted net revenue per hectare if management unit m is harvested in period t, plus the discounted residual forest value based on the projected state of the management unit at the end of the planning horizon;

A_m = the area of management unit m in hectares;

P_m = the perimeter of management unit m in meters;

CB_{pq} = the length of the common boundary between the two adjacent management units p, q in meters;

v_{mt} = the volume of sawtimber in m^3/hectare harvested from management unit m if it is harvested in period t;

M_{ht} = the set of management units that are old enough to be harvested in period t;

b_{lt} = a lower bound on decreases in the harvest level between periods t and $t+1$ (where, for example, $b_{lt} = 1$ requires non-declining harvest; $b_{lt} = 0.9$ would allow a decrease of up to 10%);

b_{ht} = an upper bound on increases in the harvest level between periods t and $t+1$ (where, for example, $b_{ht} = 1$ allows no increase in the harvest level; $b_{ht} = 1.1$ would allow an increase of up to 10%);

P = the set of all paths, or groups of contiguous management units, whose combined area is just above the maximum harvest opening size (the term "path," as used in this paper, is defined in the following discussion);

M_p = the set of management units in path p;

n_{M_p} = the number of management units in path p;

h_i = the first period in which a management unit in path i can be harvested;

J_{mt} = the set of all prescriptions under which management unit m meets the minimum age requirement for mature patches in period t;

C = the set of all clusters, or groups of contiguous management units whose combined area is just above the minimum large, mature patch size (the term "cluster," as used in this paper, is defined in the following discussion);

M_c = the set of management units that compose cluster c;

n_c = the number of management units in cluster c;

C_m = the set of all clusters that contain management unit m;

Age_{mt}^{T} = the age of management unit m at the end of the planning horizon if it is harvested in period t; and

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

Equation (1) specifies one of the three objective functions of the problem, namely to maximize the discounted net revenue from the forest during the planning horizon, plus the discounted residual value of the forest. Equation (2) maximizes the minimum amount of total area in large, mature forest patches over the time periods in the planning horizon. The rationale behind this objective is to ensure that the needs of sensitive species that require a minimum area of contiguous old forest habitat at any particular point in time to survive and to disperse will be met by the solution to the model. Equation (3) minimizes the sum of the perimeters of these patches over the entire planning horizon with the goal of promoting patch shapes that contain as much interior habitat versus edge habitat as possible. Our primary goal with these objective functions was to provide a realistic example that demonstrates the mechanisms and the utilities of the proposed multi-criteria methods.

Constraint set (4) consists of logical constraints that allow only one prescription to be assigned to a management unit, including a do-nothing prescription. To prevent management units from being scheduled for harvest before they reach a minimum harvest age, harvest variables (X_{mt}) are created only for periods where management unit *m* is old enough to be harvested. Constraint set (5) consists of harvest accounting constraints that assign the harvest volume for each period to the harvest variables (V_t). Constraint sets (6) and (7) are flow constraints that restrict the amount by which the harvest level is allowed to change between periods. In the example below, harvests were allowed to increase by up to 15% from one period to the next or to decrease by up to 3%.

Constraint set (8) consists of adjacency constraints generated with the Path Algorithm (McDill et al. 2002). These constraints limit the maximum size of a harvest opening, often necessary for legal or policy reasons, by prohibiting the concurrent harvest of any contiguous set of management units whose combined area just exceeds the maximum harvest opening size. The exclusion period imposed by these constraints equals one planning period. A "path" is defined for the purposes of the algorithm as a group of contiguous management units whose combined area just exceeds the maximum harvest opening size. These paths are enumerated with a recursive algorithm described in (McDill et al. 2002). A constraint is written for each path to prevent the concurrent harvest of all of the management units in that path, since this would violate the maximum harvest opening size. This is done for each period in which it is actually possible to harvest all of the management units in a path. (In the initial periods of the planning horizon, some of the management units in a path may not be mature enough to be harvested.)

Constraint sets (9)–(15) are the mature patch size constraints. Constraint sets (9)-(10) determine whether or not management units meet the minimum age requirement for mature patches. These constraints sum over all of the prescription variables for a management unit under which the unit would meet the age requirement for mature patches in a given period. O_{mt} is equal to 1 if and only if one of these prescriptions has a value of 1, indicating that the management unit will be "old enough" in that period. One pair of these constraints is written for each management unit in each period.

Constraint sets (11)-(12) determine whether or not a cluster of management units meets the minimum age requirement for mature patches. Clusters are defined here as contiguous groups of management units whose combined area just exceeds the minimum mature patch size requirement. All possible clusters are enumerated using a recursive algorithm described in Rebain and McDill (2003b). A cluster meets the age requirement for mature patches in period *t* if all of the management units that compose that cluster meet the age requirement, as indicated by the O_{mt} variables for the management units in that cluster. B_{ct} takes a value of 1 if and only if cluster *c* meets the age requirement in period *t*. These pairs of constraints are written for each cluster in each period.

Constraint sets (13)-(14) determine whether or not individual management units are part of a cluster that meets the minimum age requirement, i.e., whether a management unit is part of patch that is big enough and old enough. Since the clusters may overlap, this constraint set is necessary to properly account for the total area of large, mature patch habitat. These constraints say that a management unit is part of a patch that meets the minimum age and size requirement for large, mature patches in period t (BO_{mt} = 1) if and only if at least one of the clusters it belongs to meets the age requirement in that period. Constraint set (15), working in concert with objective function (2), assigns the smallest of the three total mature patch areas that correspond to the three planning periods to an accounting variable: λ. This is done by specifying through constraint set (15) that λ cannot be larger than the mature forest patch area in any period. Objective function (2) maximizes λ to ensure that it will not take a value that is less than the smallest of the three total habitat areas.

Constraint sets (16)-(18) also work together. Constraint (16) calculates the total perimeter of the mature forest patches that arise in each period, and assigns this value to accounting variable μ_t (denoting the total perimeter of the patches in period t). The sum of the total perimeters over the planning horizon is minimized by objective function (3). Constraints (17)-(18) define a new binary variable $\Omega^t{}_{pq}$ that substitutes for what would otherwise be a non-linear cross-product term ($\Omega^t_{pq} = BO_{pt}BO_{qt}$) in (16). Constraint set (18) is not necessary if objective function (3) is to minimize the perimeter. On the other hand, if the objective was to maximize edge habitat then constraint set (18) would be necessary and (17) could be dropped.

Constraint (19) is an ending age constraint. It requires the average age of the forest at the end of the planning horizon to be at least $\overline{Age}^T$ years, preventing the model from over-harvesting the forest. In the example below, the minimum average ending age was set at 40 years, or ½ the optimal economic rotation.

Constraint sets (20)-(23) identify the management unit prescription, mature patch size, and the $\Omega^t{}_{PQ}$ variables as binary.

Methods

This section describes how the four bi-objective generating techniques tested in Tóth et al. (2006) can be generalized to identify the efficient set with respect to three or more objectives. These techniques are the Alpha-Delta, the ε-Constraining, the Modified Tchebycheff and the Weighted methods.

The Alpha-Delta Method

The Alpha-Delta Method finds efficient solutions by progressively moving from one end of the efficient frontier to the other (Tóth et al. 2006). A slightly sloped weighted objective function is used with weights that are constant throughout the algorithm. The weights are normalized and the criteria values are scaled using the ideal solution vector. The relative difference in the weights assigned to the respective objectives is controlled by the parameter α (slope). This parameter has to be small enough to not miss any solutions but must be greater than zero to avoid dominated solutions. At each iteration a new efficient solution is found and the search space is constrained using the achievement vector corresponding to the new solution. Constraining the search space for problems with three or more objectives is not trivial, however.

Suppose that the tri-criteria problem described in Section 2 is solved after the three objectives are scalarized using the slope parameter (α) of the Alpha-Delta Algorithm. Let N_1, H_1, and E_1 denote the achievements on objective function (1), (2), and (3), respectively. As long as $\alpha > 0$, the following will be true for the rest of the efficient solutions: $f_1(x) < N_1$ and either $f_2(x) > H_1$ or $-f_3(x) > E_1$, where $f_i(x)$ denotes the value of objective function *i*. The latter two of these constraints can be used to restrict the search space for the remaining solutions. Inequality $f_1(x) < N_1$ must hold for any solution in this region, because if $f_1(x) \geq N_1$ was true for any one of the remaining efficient solutions then that solution would have been found at the first iteration. As long as α is small enough, we can rule out the region where $f_2(x) \leq H_1$ and $-f_3(x) \leq E_1$ both hold since if there were a solution in that region that dominates

(N_1, H_1, E_1), it would have had a higher $f_1(x)$ than N_1 and should have been found at the first iteration. To ensure that the search space is confined to objective function values of $f_2(x) > H_1$ or $-f_3(x) > E_1$ at the second iteration, the following set of constraints are added to the original problem.

$$\lambda \geq (H_1 + \delta_{hab}) y_1 \tag{24}$$

$$-\sum_{t \in T} \mu_t \geq (E_1 + \delta_{edge}) y_2 \tag{25}$$

$$y_1 + y_2 = 1 \tag{26}$$

$$y_1, y_2 \in \{0,1\} \tag{27}$$

where λ = the minimum area of mature forest habitat in patches over all periods;

μ_t = the total perimeter of mature forest habitat patches in period t;

H_1 = achievement on λ from iteration one;

E_1 = achievement on $-\sum_{t \in T} \mu_t$ from iteration one;

δ_{hab}, δ_{edge} = user-defined, sufficiently small constants;

y_1, y_2 = binary variables that ensure that only one of the constraints (24) and (25) is enforced.

Constraints (24)-(27) ensure that either the minimum area of mature forest habitat patches over all periods (λ) is strictly greater than H_1 or the total perimeter of the patches $\left(\sum_{t \in T} \mu_t\right)$ is strictly smaller than E_1. The strictly greater (or smaller) requirement is needed to avoid repeatedly picking up the same solution. This requirement is achieved by adding sufficiently small constants, δ_{hab} and δ_{edge} to the bounds on habitat area (H_1) and perimeter ($-E_1$). The either-or relationship between constraints (24) and (25) is achieved by using constraints (26)-(27), which require that either $y_1 = 1$ and $y_2 = 0$ or vice versa. If $y_1 = 1$ then only constraint (24) is enforced, and if $y_2 = 1$ then only constraint (25) is enforced. The problem is then solved again with these additional constraints. After each iteration, a new quadruplet of constraints like (24)-(27) is added to the formulation. The process is repeated until the problem becomes infeasible.

/Figure 1/

Figure 1 illustrates the general implementation of this algorithm when applied to *n*-objective problems. In the first step, the ideal solution is identified, as it is needed for scaling and normalization. In Step 2, the weighted objective function is generated with slope α. This objective function is then maximized subject to the original set of constraints ($x \in X$). If this problem is infeasible then the algorithm terminates; there are no solutions to the problem. Otherwise the problem is solved and the attainment values on the objectives with the smaller weights (F_i^k , for $i \in P \setminus \{j\}$) are used to build and add a new set of constraints to the original problem (Step 3 and 4). These constraints will be similar to (24)-(27), except now there are (*n*-1) restricted objectives and only one of the *n* constraints, $f_i(x) \geq F_i^k + \delta_i$ (for $i = 1,...,n$ but $i \neq j$), must hold. Step 5 checks whether the newly constructed constraint set from Step 4 dominates any of the previously constructed sets. If it does, i.e., if $F_i^m \leq F_i^k$ (for each $i \in P \setminus \{j\}$ and for $m < k$) then the dominated constraint set (the one that was generated in Iteration *m*) can be eliminated from the problem. After optionally removing the redundant constraints (we found in test runs that this step did not improve efficiency and therefore we did not use it), the new IP is solved and the process (Steps 2-5) is repeated until the problem becomes infeasible.

The ε-Constraining Method

The implementation of the ε-Constraining method is very similar to that of the Alpha-Delta Method. The key difference is that at each iteration *n* IPs are solved, as opposed to one (for an *n*-

objective problem), to guarantee an efficient solution. Suppose the following iteration is the k^{th} iteration. First, one of the n objectives, say $f_1^k(x)$ is maximized without regard to the rest of the objectives. If this problem is infeasible, the algorithm terminates; no more efficient solutions exist. Otherwise, the problem is solved and the resulting objective function value, F_1^k is recorded. Next, another objective is maximized, say $f_2^k(x)$ subject to $f_1^k(x) \geq F_1^k$. This problem is feasible since we know there exists at least one solution with $f_1^k(x) = F_1^k$. Call the objective function value of the resulting solution F_2^k. Now, a third objective is maximized subject to $f_1^k(x) \geq F_1^k$ and also to $f_2^k(x) \geq F_2^k$. The process is repeated until each of the objectives is maximized.

When the last objective, $f_n^k(x)$, is maximized, the rest of the objectives are constrained to $f_1^k(x) \geq F_1^k$, $f_2^k(x) \geq F_2^k$, … , $f_{n-1}^k(x) \geq F_{n-1}^k$, where F_i^k $(i = 1,2,...,n-1)$ is the objective function value that was obtained by maximizing $f_i^k(x)$. The resulting objective function value, F_n^k, together with the previously obtained F_1^k, F_2^k,..., F_{n-1}^k constitute the attainment values on the n objectives for efficient solution k (Step 3). Steps 4 and 5, as well as the stopping rule, are exactly the same as in the Alpha-Delta Algorithm.

An important common characteristic of the two methods is that as the solutions are progressively found along the efficient frontier, the attainment on one objective gradually gets worse at each new solution while the attainment on the other objectives gradually, although not necessarily monotonically, improves. This algorithmic property can be beneficial in decision making as it enables one to find efficient management alternatives that are similar in achievement values.

The Weighted Objective Function and the Tchebycheff-metric Based Methods

Both the Weighted and the Tchebycheff methods make use of an efficient decomposition of weights when applied to bi-objective problems (Tóth et al. 2006). In the case of the Weighted Method, these weights are assigned to the competing objectives and the sum of these weighted objectives is maximized. In the case of the Tchebycheff approach, the weights are assigned to the components of the Tchebycheff Metric, which measures the maximum difference between the attainment values of a potential solution and that of the ideal solution. The Tchebycheff Metric is then minimized to obtain solutions that are as close to ideal as possible. One problem with using the Tchebycheff Metric is that it may find weak Pareto-optima; solutions that lead to objective values that lie on the efficient frontier but are not corner points. In other words, at least one of the objectives can still be improved. This problem is well documented in the literature and can be overcome by using the augmented (Steuer and Choo 1983, Steuer 1986) or the modified (Kaliszewski 1987) version of the metric. In this study we used the latter approach. Instead of minimizing the maximum, we minimized the weighted differences between the attainment values with a much higher weight put on the difference in NPV than on the difference in minimum habitat area or edge length. This weight allocation, which results in a slightly sloped Tchebycheff Metric, is kept constant throughout the decomposition process.

Although varying the relative weights on the competing objectives or on the components of the Modified Tchebycheff Metric will often yield different efficient solutions, it is also possible that two different combinations of weights result in the same solution. To minimize the number of redundant solutions and the amount of computer time that is needed to find these solutions, Tóth et al. (2006) used an algorithm that decomposes the set of possible normalized weight combinations into sections (line segments in the bi-criteria case) that correspond to the same efficient solutions (Eswaran et al. 1989). The decomposition for bi-objective problems is based on the fact that if two different weight combinations yield the same solution then any linear combination of these weights will do so as well. Thus, these linear combinations can be eliminated from further consideration.

The decomposition of the weight space is not as straightforward with three or more objectives. For three objectives, the set of possible normalized weight combinations can be mapped as a triangle (Figure 2). The apexes of the triangle represent the combinations when a weight of one is assigned to one objective (or to one component of the Tchebycheff Metric) and zeros are assigned to the other two.

This triangle is illustrated in Figure 2 with apexes (1,0,0), (0,1,0) and (0,0,1). The proposed procedure, the Triangles Algorithm, decomposes this triangle into triangular sections (indifference regions) that correspond to the same efficient solutions. At each iteration one triangle is considered. If the three weight combinations that represent the three apexes of the triangle yield the same solution then no further decomposition of that triangle is necessary. Any point within the triangle (or, equivalently, any linear combination of apex-weights) will yield the same solution. If, however, the weights at the apexes yield two or three different solutions, the triangle must be divided into four smaller but identically shaped sub-triangles. These are: $\big((1/2, 1/2, 0), (0, 1/2, 1/2), (1/2, 0, 1/2)\big)$, $\big((1, 0, 0), (1/2, 1/2, 0), (1/2, 0, 1/2)\big)$, $\big((1/2, 1/2, 0), (0, 1/2, 1/2), (0, 1, 0)\big)$ and $\big((0, 0, 1), (0, 1/2, 1/2), (1/2, 0, 1/2)\big)$ in Figure 2. The apexes of the sub-triangles are either identical to one of the apexes of the "parent" triangle $\big((1,0,0), (0,1,0), (0,0,1)\big)$ or are constructed as the mean of the two of those apexes. If two of the three solutions from the "parent" triangle was the same, e.g., weights (1,0,0) and (0,1,0) both yielded the same solution, then weight combination (1/2, 1/2, 0), which is a linear combination of (1,0,0) and (0,1,0), can be assigned that solution as well. There is no need to solve the problem with weights (1/2, 1/2, 0).

/Figure 2/

At the next iteration, one of the four sub-triangles is selected and the same process is followed as in the first iteration. The weights that define the apexes of the sub-triangle are applied to the objectives of the problem (or to the components of the Tchebycheff Metric, depending on which method is used). It is entirely possible that one of the weight combinations corresponding to one of the apexes of the sub-triangle has already been applied to the problem and solved at a previous iteration, either as part of the larger triangle or when an adjacent triangle was explored. In this case, there is no need to solve the problem with these weights again. The solution from the adjacent triangle can be used in the comparisons needed to determine if the current sub-triangle should be further decomposed. Those of the three problems that have not been solved before or are not linear combinations of other weights that yield the same solution are then solved and their solutions are compared with the solutions at the other apexes. The algorithm terminates either when there are no more sub-triangles left to decompose or the largest difference between the weight combinations that correspond to the apexes of the remaining sub-triangles that could potentially require decomposition is smaller than this predefined limit: a minimum mesh or triangle size.

The following notation is used to illustrate the mechanism of the Triangles Algorithm. Let T be the set of "active" (unexplored) triangles. Let $W_i = \begin{pmatrix} w_{11}^i & w_{12}^i & w_{13}^i \\ w_{21}^i & w_{22}^i & w_{23}^i \\ w_{31}^i & w_{32}^i & w_{33}^i \end{pmatrix}$ denote the weights associated with the apexes of triangle $\Delta_i \in T$, and $\Lambda_i = -Max\left(\left|w_{11}^i - w_{21}^i\right|, \left|w_{11}^i - w_{31}^i\right|\right)$ denote the depth of Δ_i. This latter metric, 'depth' describes the size of a given triangle and is used in the algorithm to identify the largest triangles. The greater the value of Λ_i, the larger triangle Δ_i is. The algorithm decomposes the largest active triangles first. Lastly, let F_i^k (for k = 1, 2, 3) denote the objective function values that correspond to the solutions of problems $P_i^k = Max\left\{w_{k1}^i f_1(x) + w_{k2}^i f_1(x) + w_{k3}^i f_3(x) : x \in X\right\}$ for k = 1, …, 3, respectively, where $f_1(x)$, $f_2(x)$, and $f_3(x)$ are the objective functions.

Step1 and 2 is the initialization phase of the algorithm. Step 1 is to obtain the ideal solution, which is needed to scale and normalize the weights for both methods. The minimum mesh size parameter, ε is also defined (by the user) to limit the size of the triangles to be decomposed. At this point, set T is empty. Step 2 is to add the first triangle to the list of active triangles (set T). This triangle is the so-called 'parent' triangle whose apexes represent weight combinations (1,0,0), (0,1,0) and (0,0,1). The solutions to these single-objective problems have already been obtained in Step1 when the ideal solution was identified.

At the beginning of each iteration, set T is checked. If set T is empty, the algorithm terminates. If set T is non-empty, then one of the largest triangles, say triangle Δ_i is selected. If Δ_i is smaller than the predefined ε or the solutions that correspond to the apexes of Δ_i are identical, then Δ_i is removed from set T. Otherwise, four new triangles are created ($\Delta_{|T|+1}$, $\Delta_{|T|+2}$, $\Delta_{|T|+3}$, $\Delta_{|T|+4}$) with the following weights on the apexes (Step 3): $W_{|T|+1} = \begin{pmatrix} \tfrac{1}{2}\left(w_{11}^t + w_{21}^t\right) & \tfrac{1}{2}\left(w_{12}^t + w_{22}^t\right) & \tfrac{1}{2}\left(w_{13}^t + w_{23}^t\right) \\ \tfrac{1}{2}\left(w_{21}^t + w_{31}^t\right) & \tfrac{1}{2}\left(w_{22}^t + w_{32}^t\right) & \tfrac{1}{2}\left(w_{23}^t + w_{33}^t\right) \\ \tfrac{1}{2}\left(w_{11}^t + w_{31}^t\right) & \tfrac{1}{2}\left(w_{12}^t + w_{32}^t\right) & \tfrac{1}{2}\left(w_{13}^t + w_{33}^t\right) \end{pmatrix}$,

$W_{|T|+2} = \begin{pmatrix} \tfrac{1}{2}\left(w_{11}^t + w_{21}^t\right) & \tfrac{1}{2}\left(w_{12}^t + w_{22}^t\right) & \tfrac{1}{2}\left(w_{13}^t + w_{23}^t\right) \\ \tfrac{1}{2}\left(w_{21}^t + w_{31}^t\right) & \tfrac{1}{2}\left(w_{22}^t + w_{32}^t\right) & \tfrac{1}{2}\left(w_{23}^t + w_{33}^t\right) \\ w_{21}^t & w_{22}^t & w_{23}^t \end{pmatrix}$, $W_{|T|+3} = \begin{pmatrix} \tfrac{1}{2}\left(w_{11}^t + w_{21}^t\right) & \tfrac{1}{2}\left(w_{12}^t + w_{22}^t\right) & \tfrac{1}{2}\left(w_{13}^t + w_{23}^t\right) \\ w_{11}^t & w_{12}^t & w_{13}^t \\ \tfrac{1}{2}\left(w_{11}^t + w_{31}^t\right) & \tfrac{1}{2}\left(w_{12}^t + w_{32}^t\right) & \tfrac{1}{2}\left(w_{13}^t + w_{33}^t\right) \end{pmatrix}$,

$W_{|T|+4} = \begin{pmatrix} w_{31}^t & w_{32}^t & w_{33}^t \\ \tfrac{1}{2}\left(w_{21}^t + w_{31}^t\right) & \tfrac{1}{2}\left(w_{22}^t + w_{32}^t\right) & \tfrac{1}{2}\left(w_{23}^t + w_{33}^t\right) \\ \tfrac{1}{2}\left(w_{11}^t + w_{31}^t\right) & \tfrac{1}{2}\left(w_{12}^t + w_{32}^t\right) & \tfrac{1}{2}\left(w_{13}^t + w_{33}^t\right) \end{pmatrix}$. The next step is to generate and solve the 12 problems

($P_{|T|+1}^i$, $P_{|T|+2}^i$, $P_{|T|+3}^i$, and $P_{|T|+4}^i$ for i = 1, 2, 3) with the weight combinations that correspond to the apexes of the four triangles (Step 4). At most, only three of the 12 problems would have to be solved, because the same weight combinations are assigned to more than one apex (Figure 2). Furthermore, if any pair of apexes in the parent triangle led to the same solution, then any linear combination of these apexes will do so as well. There is no need to solve for a new apex if it corresponds to the linear combination of parent apexes that led to identical solutions. Finally, the four new triangles are added to set T (Step 5) and the process starts all over again by selecting another triangle.

A case study

In order to demonstrate the four methods as they generate the efficient set with respect to three objectives, a hypothetical forest planning problem, the same as in Tóth et al. (2006), was used. This forest consisted of 50 management units and could be considered slightly over-mature, since approximately 40% of the area is between 60-100 years old and the optimal rotation is 80 years. The average management unit size was 18 ha, and the total forest area was 900 ha. A 60-year planning horizon was considered, composed of three 20-yr periods. The four possible prescriptions for a given management unit were to harvest the management unit in period 1, period 2, or period 3, or not at all. The minimum rotation age was 60 years. A maximum harvest opening size of 40 ha was imposed, and groups of contiguous management units were allowed to be harvested concurrently as long as their combined area was less than the maximum opening size. All management units were smaller than the maximum harvest opening size. The wildlife species under consideration was assumed to need habitat patches that are at least 50 ha in size and at least 60 years old. Since the minimum patch size was greater than the maximum harvest size, these patches had to be composed of more than one unit. We also specified in the IP formulation that at least one habitat patch must develop over the 60 year planning horizon. This meant that no efficient management alternatives were sought below 50 hectares of mature forest patch production.

/Table 1/

The tri-criteria IP described in the Model Formulation section was built for this hypothetical forest, resulting in a model with 4,794 constraints and 2,412 variables. Table 1 shows the distribution of the constraints by constraint types. The algorithms introduced in the Methods section were implemented using a 2-thread CPLEX 11.1 (ILOG CPLEX 2008) on a dual processor Intel® XEON™ CPU 3.00 GHz computer with 3.25 GB RAM under a Windows platform (Microsoft Windows XP Professional Version 2002, Service Pack 3). Programs to automate the algorithms were written in Microsoft Visual Basic 6 and .Net 2005 using the ILOG CPLEX Callable Libraries. The relative MIP gap tolerance parameter (optimality gap) was set to 0 and the integrality tolerance parameter was set to $1.e-07$ (0.00001%). These strict settings were needed to avoid dominated solutions and to make sure that the numerically sensitive algorithmic constructs, such as the either-or constraints in the Alpha-Delta and the ε-Constraining methods and the composite weighted expressions in the other two techniques, would work properly. The working memory limit was set to default 1MB. The parameters of the Alpha-Delta and the ε-Constraining algorithms, α, δ_{hab}, and δ_{edge} were set to $1°$, 0.01 ha, and 0.47 m (1 pixel), respectively (α only applies to the Alpha-Delta Method). The latter two settings instruct the algorithms not to look for solutions that simultaneously lead to a less than 0.01ha difference in patch area production and less than 0.47 m in edge production. We tried smaller values for these parameters but found no change in the efficient set. The depth parameter in the Triangles Algorithm (ε) was set to zero for both the Weighted Objective Functions and the Modified Tchebycheff-metric-based approaches meaning that running time was the only constraint for these algorithms to find efficient solutions. We set the parameters for each technique so that the highest number of efficient solutions would be found given a time limit. Although rigorous parameter-tuning was outside of the scope of this study, we made an attempt to optimize the performance of the algorithms. Our primary goal was to provide an insight for the reader about the pros and cons of the mechanisms of the proposed methods.

The experiment addressed the following questions: (1) How many of the efficient solutions can each algorithm identify within 20 hours of computer time? (2) How evenly are these solutions distributed along the efficient frontier? (3) How easily can a user filter the solutions in line with the DM's interests? and (4) How easily do the methods generalize to the *n*-objective case?

Results and Discussion

The number and distribution of efficient solutions

The ε-Constraining Method found the highest number of Pareto-optimal management alternatives (99) within the preset time interval of 20 hours. The Alpha-Delta found 97, the Tchebycheff Method found 76, and the Weighted Method found 35 solutions. Figure 4 and 5 graph the solutions in the objective space: Figure 4 in a 3-dimensional rendering while Figure 5 in 2-dimensional projections.

The solution times are summarized in Figure 6, where the main diagram displays the cumulative solution times for each method. The four smaller charts show the individual solution times that were required to find each new efficient solution. Note that since the ε-Constraining finds each new solution in three steps, the Alpha-Delta in one step and the Tchebycheff and the Weighted Method finds them in many steps, the individual solution times do not necessarily correspond to individual IPs. One common trend that can be seen from the diagrams is that, after a very productive initial phase, finding new efficient solutions became increasingly time-consuming for each method. We provide an explanation for this trend in the discussion that follows.

/Figure 4./

/Figure 5./

The Alpha-Delta and the ε-Constraining methods found solutions mostly on one side of the efficient frontier, while the solutions identified by the other two methods were more evenly distributed. Even though the Weighted and especially the Tchebycheff Method provided a better overall estimation of the frontier, the ε-Constraining and the Alpha-Delta methods described one part of the frontier in more detail. The main reason for the difference is that while the Alpha-Delta and the ε-Constraining methods worked gradually off of one starting point and found solutions sequentially, the other two methods found more evenly distributed solutions as they start out with solutions that are as contrasting with respect to their assigned weights as possible. It is important to point out, however, that both the Alpha-Delta and the ε-Constraining methods can be instructed to find solutions that are more separated from each other along the frontier by assigning higher values to parameters δ_{hab}, and δ_{edge}.

/Figure 6./

All methods except the Weighted Objectives are capable of identifying non-supported Pareto-optima (non-supported Pareto-optima are efficient solutions that are not corner points of the convex hull of the efficient set, Tóth et al. 2006). As a result, the three methods can explore the efficient frontier in more detail than the Weighted Method. However, as the Alpha-Delta and the ε-Constraining methods explore the efficient frontier starting from one end (from the highest levels of NPVs) and they require a set of either-or constraints with associated binary variables to be added to the problem at each iteration, the IP that needs to be solved becomes increasingly hard as the algorithms proceed (see bottom charts in Figure 6). The structure of the tri-criteria forest management planning problem used in this experiment might also account for this increasing combinatorial complexity. Optimal solutions that lead to greater amounts of mature forest habitat in large patches and to lesser amounts of timber revenues might be harder to identify. This is because a higher number of spatial arrangements of mature forest habitat exist if larger areas are allowed and this larger set must be evaluated in the optimization process to find optimal solutions. Depending on when the IP becomes too time-consuming to solve and what time constraints are imposed, the Alpha-Delta and the ε-Constraining methods might or might not be able to scan the entire efficient frontier. One way to mitigate this problem is to run three separate algorithms each starting from a different end of the frontier. The Alpha-Delta can be instructed to work its way off the highest possible level of minimum habitat area or from the lowest possible length of perimeter. This way, only the central part of the efficient frontier would have to be explored using a larger burden of 'either-or' constraints, and the peripheral solutions might be found relatively easily. This combined algorithm can stop once the three sub-algorithms (*n* sub-algorithms for an *n*-objective problem) "meet" somewhere in the middle of the efficient frontier – i.e., when they find an identical efficient solution. Of course, there is no guarantee that the IPs would not become intractable before these sub-algorithms meet. This combined approach, however, should increase the number and diversity of efficient solutions found.

Although the Weighted Method found only 35 solutions, it might still be a preferred alternative to solve multi-objective IPs if the original problem has a special structure (e.g., total unimodularity: Wolsey 1998) that would be destroyed by using other methods (e.g., ReVelle 1993). In these cases, destroying this structure in order to find the non-supported Pareto-optima, and thus better describing the efficient frontier, might not be worthwhile. Totally unimodular or other special structures are unlikely, however, in realistic forest planning problems where in most cases a large variety of complicating constraints need to be imposed in the model (Tóth et al. 2006).

In conclusion, as in the bi-criteria case, a combined approach can be recommended. The forest planner could use the Weighted or the Tchebycheff methods to obtain a well-distributed efficient set and present the results to the DM. Then, in line with the DM's interests, one area of the efficient frontier could further be explored using the ε-Constraining or the Alpha-Delta Method. This approach would take advantage of the algorithmic differences in each method.

Filtering the efficient solutions

Besides using the Weighted or the Tchebycheff methods as initial filters, there are several other, indirect ways to control the 'spacing' of the efficient solutions along the efficient frontier with the proposed algorithms. Filtering the efficient set might be advantageous because: (1) it can significantly reduce computing time, and (2) a well-distributed subset of the efficient frontier might provide a sufficient pool of alternatives for the DM to choose from or to guide him to further explore a particular sub-region of interest along the frontier.

Increasing the values of parameters α, δ_{hab} and δ_{edge} in the Alpha-Delta-, and parameters δ_{hab} and δ_{edge} in the ε-Constraining Algorithm will increase the spacing of the efficient solutions in the objective space. The settings of δ_{hab} and δ_{edge} will ensure that no two solutions will be found whose achievements in terms of minimum habitat area and edge length are both within δ_{hab} and δ_{edge}, respectively. The spacing of the efficient solutions with the Weighted Objective Functions and the Tchebycheff-metric-based approaches can be controlled to some extent by adjusting the minimum mesh size parameter ε in the Triangles Algorithm. In practice, it might be useful to start with a large mesh size and cover the entire weight space, and then focus the search to a sub-region of interest using a smaller mesh size in that region. How large should the initial mesh size be? In our test case, both the Weighted and the Tchebycheff methods were processing triangles at the $1/128$ level after 20 hours of computing time. The Weighted Method decomposed 88%, the Tchebycheff 77% of these triangles.

Generalization to the *n*-objective case

One additional advantage of the Alpha-Delta- and the ε-Constraining methods over the other two approaches is that their generalization to the *n*-objective case is fairly straightforward – at least from a technical, integer programming point of view (see Figure 1). Generalizing the Triangles Algorithm to the *n*-objective case is not as straightforward as with the above two methods. Instead of the triangles in the tri-objective case, *n*-dimensional polyhedra (tetrahedra in the 4-objective case) would have to be decomposed into *n*-dimensional sub-polyhedra. Since each *n*-dimensional polyhedron has *n* apexes, *n* new problems would have to be solved and compared ($n!$ pair-wise comparisons) at each iteration. This increased computational burden, however, might be offset by the fact that the IP sub-problems (the problems that are solved at the apexes of the *n*-dimensional polyhedra) are simpler than those of the Alpha-Delta- or the ε-Constraining methods. Unlike the feasible region of the latter two methods, the feasible region of the IP sub-problems in the Triangles Algorithm is constant, only the weights on the objectives (or the components of the Tchebycheff-metric) change.

Ecological and management implications

The tradeoff information generated for the hypothetical test problem demonstrates the utility that one can expect from the proposed techniques in real applications. Looking at Figure 4 or the diagram in the center of Figure 5, one can conclude for instance that minimum mature forest habitat patch production costs roughly \$70-100,000 in terms of forgone timber revenues for every 50 ha increase between the 50 ha required minimum and the 170 ha potential maximum. An extra $100-200,000 is needed if minimum boundary patches are desired.

The vertical clustering of efficient solutions around some minimum habitat area thresholds, such as 135, 150 or 170 ha (Figure 5 bottom) implies that the forest manager would have some flexibility (as much as $200,000 at the 170 ha level, Figure 5 top) to simultaneously generate mature forest patches as well as timber revenues. The potential losses or gains associated with edge production are traded off against potential gains or losses in timber revenues when one switches between management alternatives within these clusters. This demonstrates the benefits of the multi-criteria techniques in that they offer a range of choices, whenever possible and without the DMs' *a priori* bias that would otherwise manifest in the form of targets or hard constraints if alternative approaches such as goal programming were used.

The vertical clustering around some minimum habitat area thresholds might have landscape ecological implications as well. Consider for instance the cluster around the 170 ha level one more time (Figure 5 bottom). Starting at the option that leads to the least amount of NPV, approximately $2M (see the lowest point in the center diagram in Figure 5), a fair amount of additional harvesting is possible to achieve as much as $2.18M in timber revenues without having to forgo any of the 170 ha of minimum mature forest habitat production. The only loss is the increase in perimeter-area ratios of the patches. Once, however, anything beyond the $2.18M is desired, the extra harvests that would be needed would cause a significant drop in minimum mature forest habitat production. One would have to switch to the 150 ha cluster to find more profitable alternatives. Are these discrete drops between the clusters characteristic to this particular problem? The forest planner would benefit from knowing about thresholds where small amounts of change in harvest intensity can cause significant losses in habitat structure or cohesion. The techniques proposed in this paper can help in identifying these thresholds.

Finally, any clustering of solutions in the objective space can carry significant value for the DMs. Clusters of solutions that are similar in terms of achieving the objectives that are explicitly incorporated in the model can be analyzed in terms of their contribution to a fourth or fifth objective. Although there is no guarantee that any of the points in the cluster would be Pareto-optimal with respect to an additional criterion, this criterion might help the DMs eliminate some solutions from the pool.

Numerical issues

It is important to point out that the algorithms proposed in this study build on numerically sensitive constructs such as the either-or constraints in the ε-Constraining and the Alpha-Delta methods or the composite objective functions in the other two methods. Some of these constructs do not function properly if the optimality and integrality tolerance parameters in the IP solvers are not set tight enough. If a weighted objective function is used for example, the smallest amounts of sub-optimality might lead to solutions that are totally irrespective of the assigned weights. Consider the example on Figure 7: although the weighted objective function (dashed line) in this case in maximized at Point A, using a small optimality tolerance gap could easily lead to Point D. It is easy to see how this loss of optimality can make the triangular decomposition process dysfunctional. The assumption that the linear combination of two sets of weights that lead to identical solutions would also lead to the same solution does not hold anymore. Another obvious consequence of sub-optimality is that the multi-objective techniques might produce dominated solutions. The only way to avoid this situation is to set the optimality tolerance gap as small as possible, which has the associated cost of increased computing time.

Conclusions

The primary value of generating the set of efficient solutions to forest management problems is to help decision makers acquire a more holistic understanding of the problem by providing information about the tradeoffs, the production possibilities, and the degree of incompatibility between competing objectives. This understanding should facilitate selecting the best compromise management alternatives.

We presented four ways to generate the set of Pareto-optimal solutions to spatial forest planning problems with three or more competing objectives. While generalizing the bi-objective algorithms to three objectives is not trivial, generalizing the proposed tri-criteria algorithms to handle problems with four or more objectives is methodologically straightforward. The results from one test of the four algorithms suggest that there is no clear winner in terms of computational performance: each method has positives and negatives. Given the algorithmic differences behind the techniques and the snapshot of computational results, we can conclude that a combined utilization of the beneficial properties of either the Weighted or the Modified Tchebycheff methods and either the ε-Constraining or the Alpha-Delta methods would probably work the best in practice. Using one of the former two methods to generate an initial rough estimate of the tradeoffs can be followed up by one of the latter two techniques to find further solutions in the decision makers' regions of interest.

Applying the four methods to larger problems might be computationally expensive if, as in this study, truly optimal solutions are sought. Reducing the optimality tolerance, i.e., accepting sub-optimal solutions is certainly an option if computing time is a constraint: the efficient frontier could be approximated in a fraction of the time that would be needed to provide an exact representation. Due to the numerically sensitive nature of multi-objective programming, however, some of the proposed techniques might exhibit adverse algorithmic behaviors leading to dominated solutions or to too few efficient solutions. How much optimality can be forgone while still providing a meaningful representation of the tradeoff frontier for the DMs? This question can only be answered by comprehensive computational experiments that explore the tradeoffs between solution times and Pareto-optimality on larger forest planning problems. In the meantime, small-scale, pilot applications of the techniques with real decision makers still remain important as they can answer questions such as (1) how important it is to find non-supported Pareto-optima, (2) how to visualize and present the alternatives to the DMs, and (3) to what extent can these methods promote consensus between multiple stakeholders. By the time these questions are answered, our computational capabilities might improve to such a degree that allows us to solve large problems to desired levels of optimality. In the last three years alone, optimization technology improved so drastically that we could run the experiments presented in this study in 20 hours in 2008 instead of the three weeks we needed in 2005.

Finally we emphasize that finding the best ways to visualize the efficient management alternatives and optimizing the interaction with the DMs are key issues that need to be addressed in future research in order to successfully apply these methods. Visualizations with the potential to display three or more management objectives have already been proposed in the past (e.g., Schilling 1976 or Lotov et al. 2004) and recent improvements in computer aided 3-dimensional rendering and animation could significantly enhance the viability of these tools in natural resource decision making.

References

Alves, M.J., and J. Clímaco. 2001. Indifference sets of reference points in multi-objective integer linear programming. *Journal of Multi-Criteria Decision Analysis*. 10(4):177-189.

Arthaud, G.J., D.W. Rose 1996. A methodology for estimating production possibility frontiers for wildlife habitat and timber value at the landscape level. *Can. J. For. Res.* 26(12): 2191-2200.

Bettinger, P., J. Sessions, and K. Boston. 1997. Using Tabu Search to schedule timber harvests subject to spatial wildlife goals for big game. *Ecol. Model*. 94(2-3):111-123.

Chalmet, L.G., L. Lemonidis, and D.J. Elzinga. 1986. An algorithm for the bi-criterion integer programming problem. *Eur. J. Oper. Res.* 25(2):292-300.

Charnes, A., W.W. Cooper and R. Ferguson. 1955. Optimal estimation of executive compensation by linear programming. *Manage. Sci.* 1(2):138-151.

Church, R., R. Gerrard, A. Hollander, and D. Stoms. 2000. Understanding the tradeoffs between site quality and species presence in reserve site selection. *For. Sci.* 46(2):157-167.

Church, R.L., D.M. Stoms, and F.W. Davis. 1996. Reserve selection as a maximal covering location problem. *Biol. Conserv*. 76(2):105-112.

Cox, E.S., and J. Sullivan. 1995. Harvest scheduling with spatial wildlife constraints: an empirical examination of tradeoffs. *J. Environ. Manage*. 43(4):333-348.

Ehrgott, M., and M.M. Wiecek. 2005. Multiobjective Programming. In: *Multiple Criteria Decision Analysis - State of the Art Surveys*. Figuera, J., S.Greco, and M. Ehrgott (eds). Springer Science+Business Media, Inc.: New York; p. 1045.

Eswaran, P.K., A. Ravindran, and H. Moskowitz. 1989. Algorithms for nonlinear integer bicriterion problems. *J. Optimiz. Theory App*. 63(2):261-279.

Geoffrion, A.M. 1968. Proper efficiency and the theory of vector maximization. *J. Math. Anal. and Appl*. 22:(3)618-630.

Haimes, Y.Y., L.S. Lasdon, and D.A. Wismer. 1971. On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization. *IEEE T. Syst. Man. Cyb*. 1(3):296-297.

Holland, D.N., R.J. Lilieholm, D.W. Roberts, and J.K. Gilless. 1994. Economic trade-offs of managing forests for timber production and vegetative diversity. *Can. J. For. Res*. 24(6):1260-1265.

Kaliszewski, I. 1987. A modified weighted tchebycheff metric for multiple objective programming. *Comput. Oper. Re*s. 14(4):315.

Leuschner, W.A., J.R. Porter, M.R. Reynolds, and H.E. Burkhart. 1975. A linear programming model for multiple-use planning. *Can. J. For. Res*. 5(3): 485-491.

Lotov, A. V., Bushenkov, V., Kamenev, G.K. 2004. Interactive Decision Maps: Approximation and Visualization of Pareto Frontier. Kluwer Academic Publishers, Boston. p16-21.

McDill, M.E., S. Rebain, and J. Braze. 2002. Harvest scheduling with area-based adjacency constraints. *For. Sci*. 48(4):631-642.

Mealey, S.P., and J.R. Horn. 1981. Integrating wildlife habitat objectives into the forest plan. *T. N. Am. Wildl. Nat. Res.* 46():488-500.

Pareto, V. 1909. *Manuel d'Economie Politique*. Marcel Giard, Paris

Rebain, S., and M.E. McDill. 2003a. Can mature patch constraints mitigate the fragmenting effect of harvest opening size restrictions? *Int. T. Oper. Res*. 10(5):499-513.

Rebain, S., and M.E. McDill. 2003b. A mixed-integer formulation of the minimum patch size problem. *For. Sci.* 49(4):608-618.

ReVelle, C. 1993. Facility siting and integer-friendly programming. *Eur. J. Oper. Res.* 65(2):147-158.

Richards, E.W., and E.A. Gunn. 2000. A model and tabu search method to optimize stand harvest and road construction schedules. *For. Sci*. 46(2):188-203.

Roise, J., J. Chung, R. Lancia, and M. Lennartz. 1990. Red-cockaded woodpecker habitat and timber management: production possibilities. *South. J. Appl. For*. 14(1):6-12.

Rosenbaum, K.L. 2000. Discussion - Forestry in the New Millenium. In: *A vision for the U.S. Forest Service. Resources for the Future*. Sedjo, R. A. (ed). Washington, DC; p. 273.

Sadagopan, S., and A Ravindran. 1982. Interactive solution of bi-criteria mathematical programs. *Naval Res. Logist. Q.* 29(3):443-459.

Schilling, D. 1976. Multiobjective and Temporal Considerations in Public Facility Location. PhD. Thesis, Dep. of Geog. and Environ. Engrg., John Hopkins University, Baltimore, Maryland. 118p.

Snyder, S., and C. ReVelle. 1997. Multiobjective grid packing model: an application in forest management. *Loc. Science*. 5(3):165-180.

Steuer, R.E. 1986. *Multiple criteria optimization: theory, computation, and application.* John Wiley, New York. 546p.

Steuer, R.E., and E.U. Choo. 1983. An interactive weighted Tchebycheff procedure for multiple objective programming. *Math. Program*. 26(3):326-344.

Thomas, J.W. 2000. What now? From a former chief of the Forest Service. In: *A vision for the U.S. Forest Service. Resources for the Future*. Sedjo, R. (ed). Washington, DC. 273 p.

Tóth, S.F., and M.E. McDill. 2008. Promoting large, compact mature forest patches in harvest scheduling models. *Environ. Model. Assess.* 13(1):1-15.

Tóth, S.F., M.E. McDill, and S. Rebain. 2006. Finding the efficient frontier of a bi-criteria, spatially-explicit harvest scheduling problem. *For. Sci*. 52(1): 93-107.

Williams, J.C. 1998. Delineating protected wildlife corridors with multi-objective programming. *Environ. Model. Assess.* 3(1-2):77-86.

Wolsey, L.A. 1998. *Integer programming*. John Wiley, New York. 288 p.

List of Figures

List of Tables

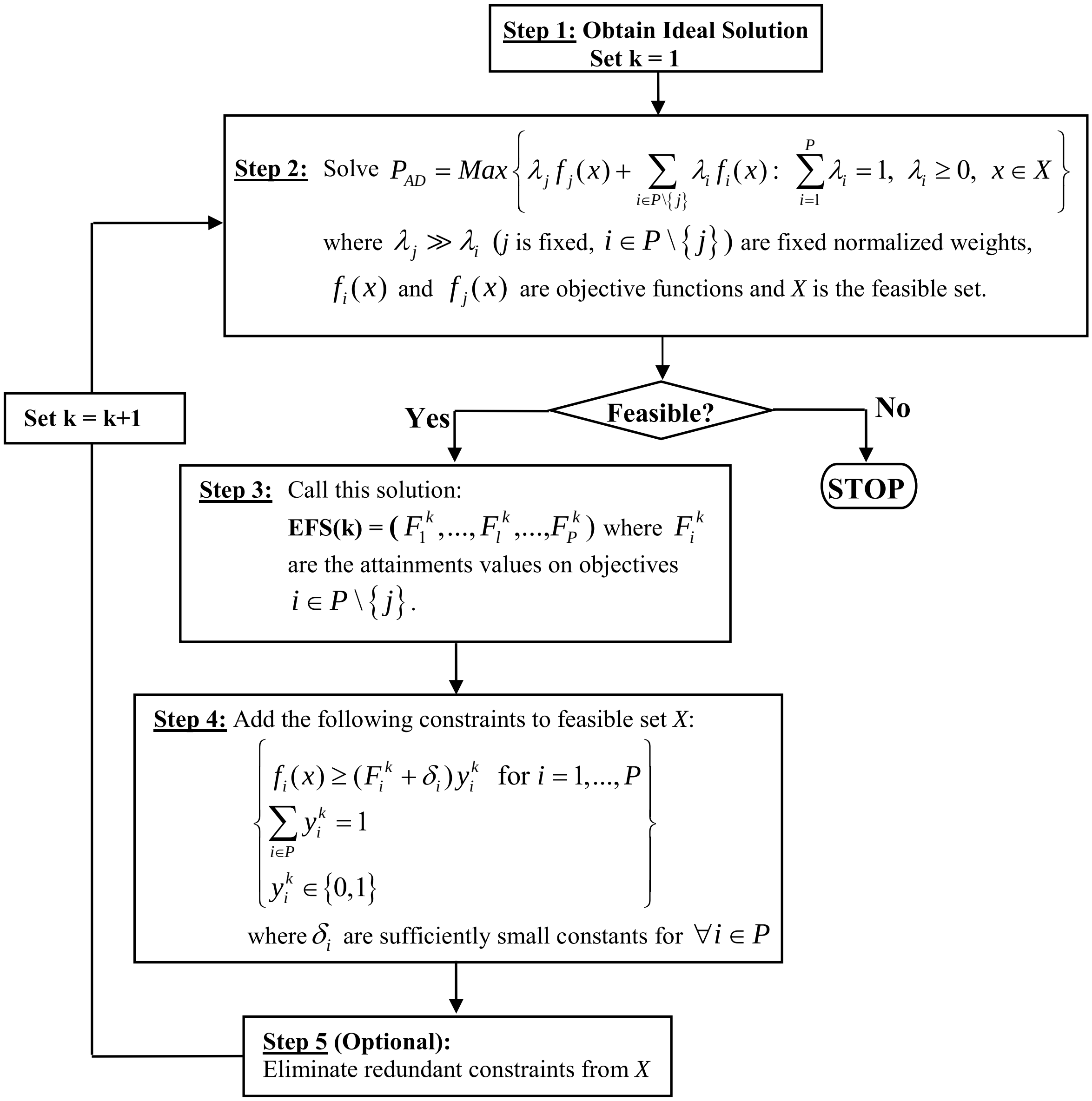

Step 1: Obtain Ideal Solution
Set k = 1
Step 2: Solve $P_{AD} = Max\left\{\lambda_j f_j(x) + \sum_{i \in P\setminus\{j\}} \lambda_i f_i(x) : \sum_{i=1}^{P} \lambda_i = 1,\ \lambda_i \geq 0,\ x \in X\right\}$
where $\lambda_j \gg \lambda_i$ (j is fixed, $i \in P \setminus \{j\}$) are fixed normalized weights,
$f_i(x)$ and $f_j(x)$ are objective functions and X is the feasible set.
Feasible?
Yes
No
STOP
Set k = k+1
Step 3: Call this solution:
EFS(k) = ($F_1^k, \ldots, F_l^k, \ldots, F_P^k$) where F_i^k are the attainments values on objectives $i \in P \setminus \{j\}$.
Step 4: Add the following constraints to feasible set X:
$\left\{\begin{array}{l} f_i(x) \geq (F_i^k + \delta_i) y_i^k \quad \text{for } i = 1, \ldots, P \\ \sum_{i \in P} y_i^k = 1 \\ y_i^k \in \{0,1\} \end{array}\right\}$
where δ_i are sufficiently small constants for $\forall i \in P$
Step 5 (Optional):
Eliminate redundant constraints from X

Figure 1.

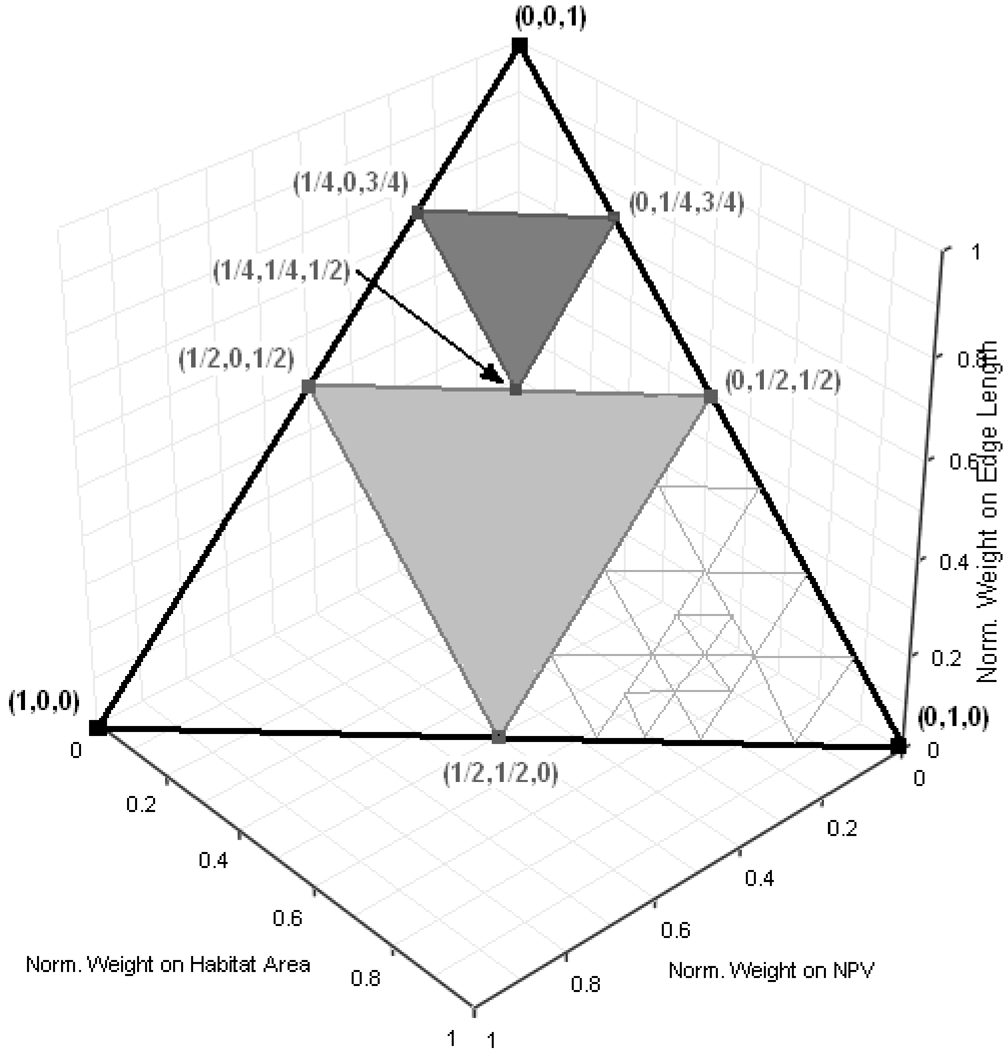
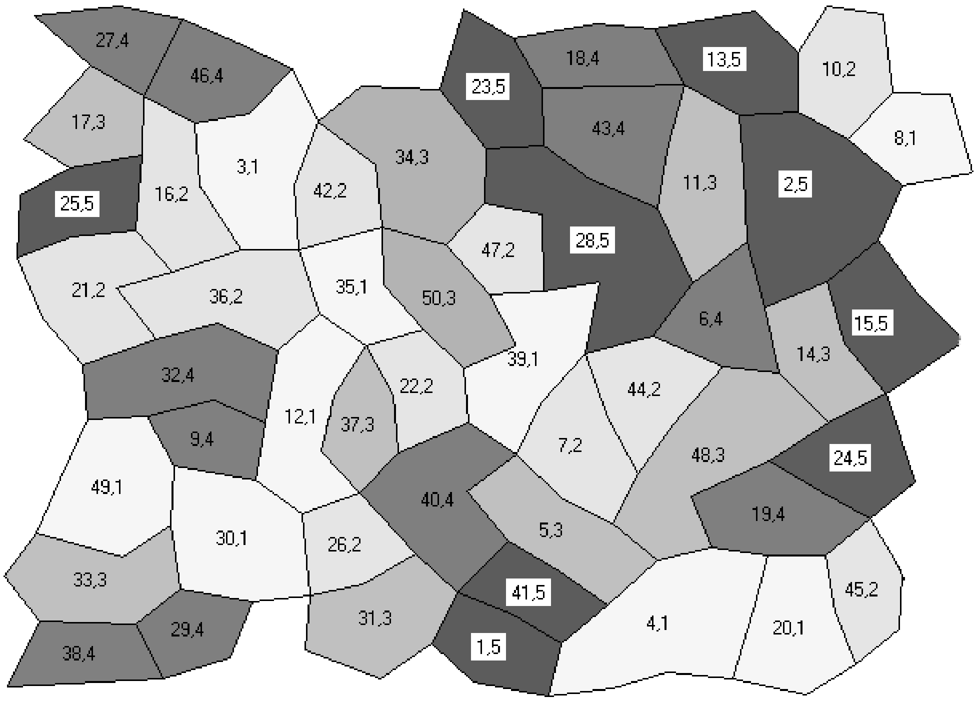

(0,0,1)
(1/4,0,3/4)
(0,1/4,3/4)
(1/4,1/4,1/2)
(1/2,0,1/2)
(0,1/2,1/2)
(1,0,0)
(0,1,0)
(1/2,1/2,0)
Norm. Weight on Edge Length
Norm. Weight on Habitat Area
Norm. Weight on NPV
0
0.2
0.4
0.6
0.8
1

Figure 2.



27,4
46,4
23,5
18,4
13,5
10,2
17,3
3,1
34,3
43,4
8,1
42,2
16,2
25,5
11,3
2,5
28,5
47,2
21,2
36,2
35,1
50,3
6,4
15,5
14,3
39,1
32,4
22,2
44,2
12,1
37,3
9,4
7,2
48,3
24,5
49,1
40,4
19,4
5,3
30,1
26,2
33,3
45,2
41,5
31,3
4,1
29,4
20,1
38,4
1,5

Figure 3.

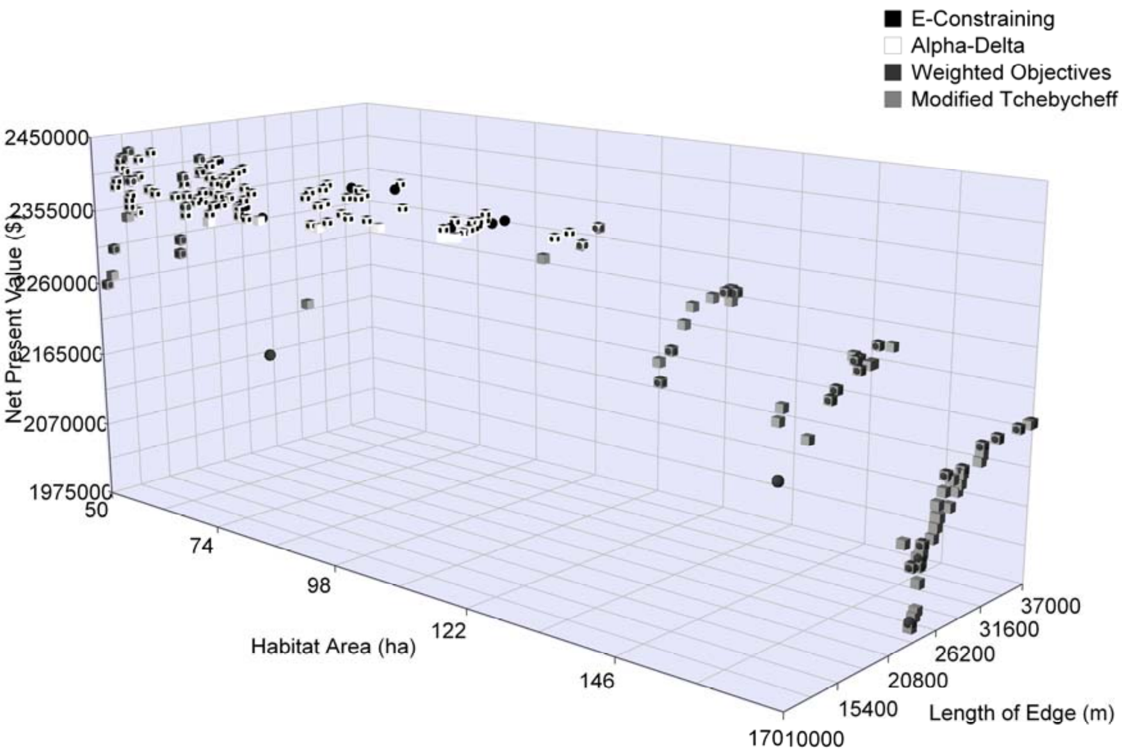

E-Constraining
Alpha-Delta
Weighted Objectives
Modified Tchebycheff
Net Present Value ($)
2450000
2355000
2260000
2165000
2070000
1975000
Habitat Area (ha)
50
74
98
122
146
170
Length of Edge (m)
10000
15400
20800
26200
31600
37000

Figure 4.

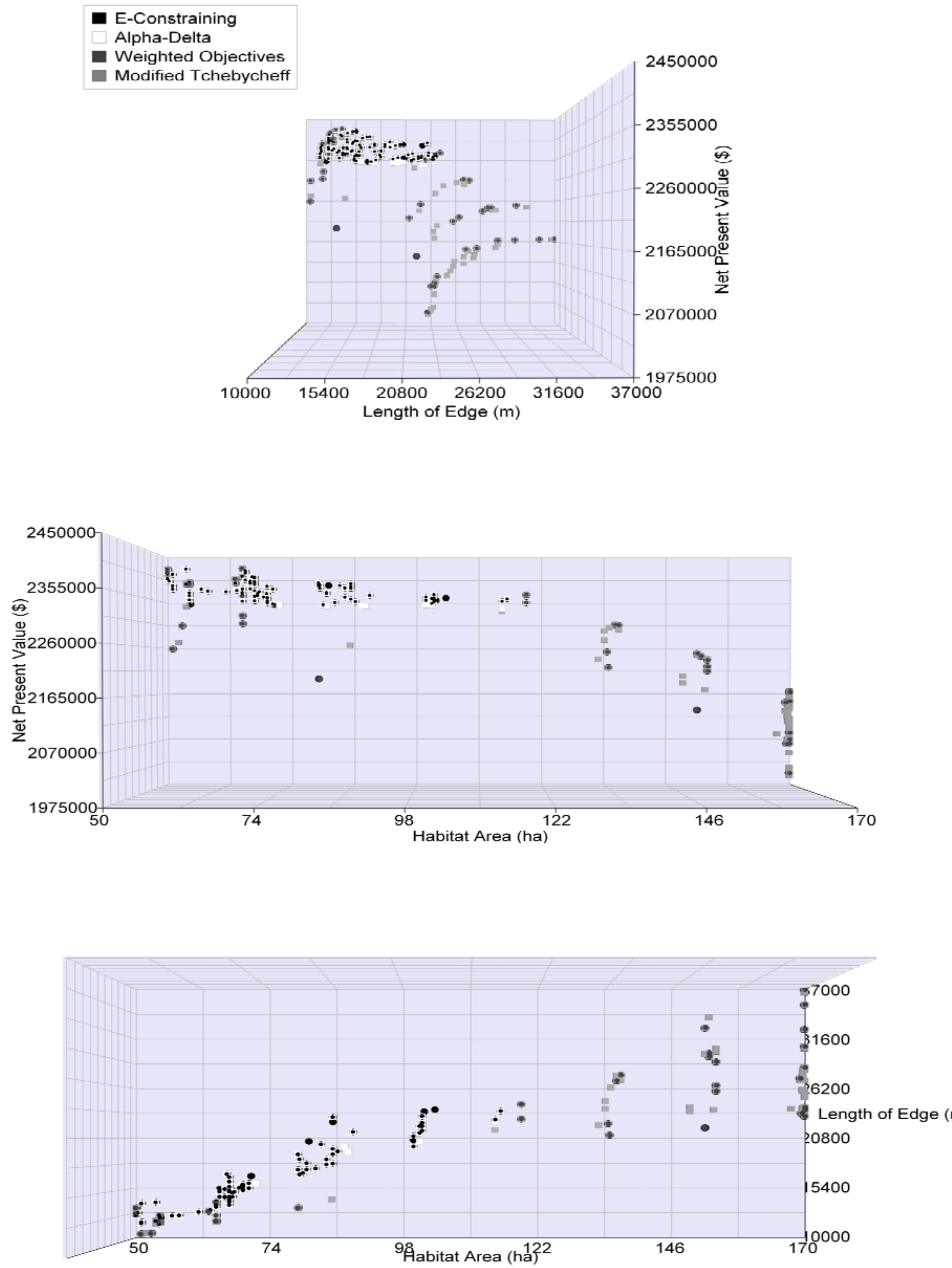

E-Constraining
Alpha-Delta
Weighted Objectives
Modified Tchebycheff
2450000
2355000
2260000
2165000
2070000
1975000
Net Present Value ($)
10000
15400
20800
26200
31600
37000
Length of Edge (m)
50
74
98
122
146
170
Habitat Area (ha)

Figure 5.

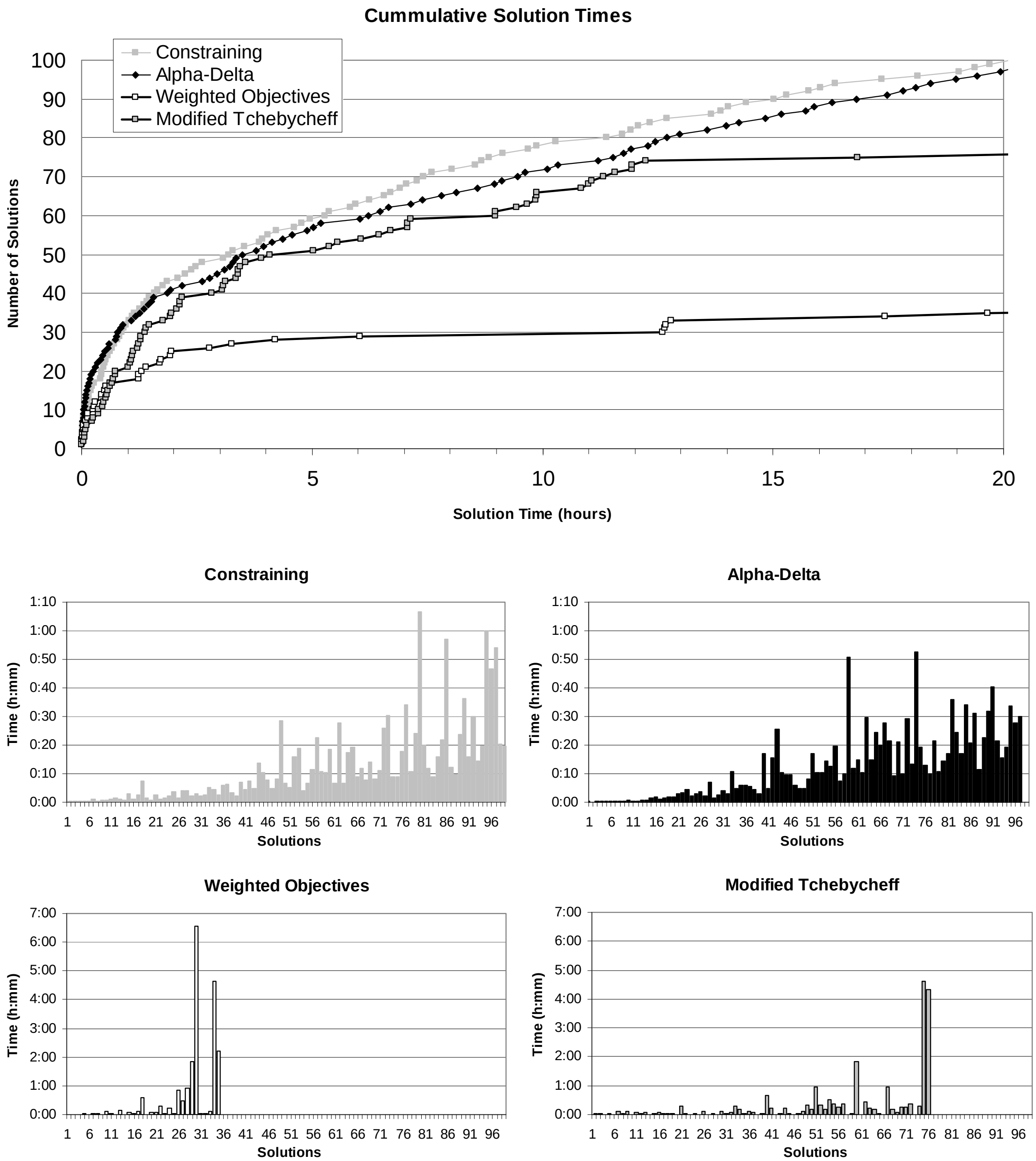

Cummulative Solution Times
Constraining
Alpha-Delta
Weighted Objectives
Modified Tchebycheff
Number of Solutions
Solution Time (hours)
Constraining
Alpha-Delta
Weighted Objectives
Modified Tchebycheff
Time (h:mm)
Solutions

Figure 6.

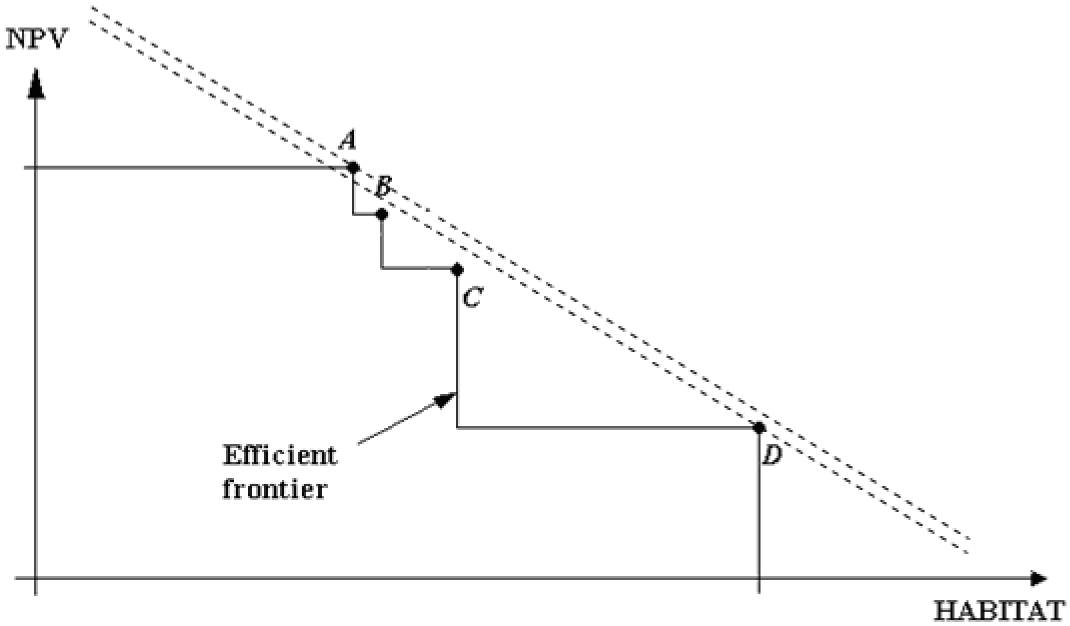


Figure 7.

Table 1.

Constraint Type (equation number)	(4)	(5)	(6,7)	(8)	(9,10)	(11,12)	(13,14)	(15,16)	(17,18)	(19)	Total
Number of Constraints	50	3	2	248	150	1,617	150	3	324	1	4,794