Finding the Efficient Frontier of a Bi-Criteria, Spatially-explicit, Harvest Scheduling Problem

Sándor F. Tóth, Marc E. McDill, and Stephanie Rebain

- Corresponding author: **Dr. Sándor F. Tóth**
  Research Associate
  Penn State School of Forest Resources
  214B Ferguson Bldg,
  University Park, PA 16802
  Phone: (814) 865-1602
  Fax: (814) 865-3725
  E-mail: sft108@psu.edu ,
  URL: http://www.personal.psu.edu/sft108/

- Second author: **Dr. Marc E. McDill**
  Associate Professor
  Penn State School of Forest Resources
  214B Ferguson Bldg,
  University Park, PA 16802
  Phone: (814) 865-1602
  E-mail: mem14@psu.edu

- Third author: **Stephanie A. Rebain**
  Forester
  USDA Forest Service, Forest Management Service Center
  2150A Centre Avenue, Suite 341A
  Ft. Collins, CO 80526-1891
  Phone: (970) 295-5793, E-mail: sarebain@fs.fed.us

Acknowledgements: The authors thank the Associate Editor and two anonymous reviewers for their helpful comments. Thanks also to the Pennsylvania Bureau of Forestry for providing financial support for this research.

Abstract: This paper evaluates the performance of five traditional methods and one new method of generating the efficient frontier for a bi-criteria, spatially-explicit harvest scheduling problem. The problem is to find all possible efficient solutions, thus defining the trade-offs between two objectives: (1) maximizing the net present value of the forest and (2) maximizing the minimum area over the planning horizon in large, mature forest patches. The methods for generating the efficient frontier were tested using a hypothetical forest consisting of 50 stands. The methods were compared based on the number of efficient solutions each method can identify and on how quickly the solutions were identified. The potential to generalize these algorithms to three- or n-criteria cases is also assessed. Three of the traditional approaches, the ε- constraining-, the Triangles method, the decomposition algorithm based on the Tchebycheff metric, and the new, proposed method are capable of generating all or most of the efficient solutions. However, the Triangles and the new method far outperformed the other approaches in terms of solution time. The new method, called Alpha-Delta, appears to be the simplest to generalize to the tri-criteria case.

Key words: Multi-criteria optimization, wildlife habitat, trade-off analysis, 0-1 programming

Introduction

Society expects more from its forest resources than merely timber production. Increasingly, values such as wildlife habitat, recreation, water quality, aesthetics and spiritual values are also recognized. In accordance with these expectations, the Multiple-Use Sustained-Yield Act (1960) requires the national forests of the United States to be managed for the multiple uses of water, timber, wildlife, fish, recreation, and range (Fedkiw 1997). The emerging field of multiple-objective forest planning reflects this diverse nature of forest resources management (Pukkala 2002, p. v). Sustaining large patches of mature forests (forest stands that are older than a certain age) throughout the planning horizon can contribute to fulfilling many of the multiple uses demanded by society (Rebain and McDill 2003a). In addition, adjacency constraints, which limit the size of harvest openings, have been promoted as contributing to these objectives (e.g. Thompson et al. 1973, Jones et al. 1991, Murray and Church 1996a, 1996b, Snyder and ReVelle 1996a, 1996b, 1997a, 1997b, Carter et al. 1997, Murray 1999). However, adjacency constraints tend to work against the goal of developing and preserving large, mature patches of forest (Harris 1984, Franklin and Forman 1987, Rebain and McDill 2003a). As adjacency constraints are intended to prevent large clearcuts, they tend to disperse harvesting activities across the forest in relatively small patches. Large, contiguous tracts of mature forests are not likely to be maintained this way.

One way of tackling this problem is to include constraints that require the models to maintain a minimum total area in mature patches meeting both a minimum age and a minimum size requirement, while maximizing the net present value (NPV) of the forest (Rebain and McDill 2003). However, it might be difficult to identify an appropriate total area of large, mature patches that will adequately meet conservation goals but not be overly restrictive. Nevertheless, single-objective models have often been applied to forest planning problems with multiple objectives where the minimum or maximum level of other outputs or values are defined by constraints (Leuschner et al. 1975, Mealy and Horn 1981, Cox and Sullivan 1995, Bettinger et al. 1997). *A priori* methods, such as goal programming (Field 1973, Kao and Brodie 1979, Field et al 1980, Arp and Lavigne 1982, Hotvedt 1983, Mendoza 1987, Rustagi and Bare 1987, or Davis and Lui 1991) also suffer from the limitation that the decision maker (DM) is required to identify his or her preferences prior to the solution process. Expecting the DM to specify the desired level of achievement or to specify his or her preferences for the various objectives without knowing what is possible is not only unrealistic but might also lead to poor management decisions. An interactive method, where the DM helps drop certain regions of the feasible solution set by comparing and ranking a limited number of alternative solutions, is a feasible approach that might remedy this shortcoming. With an interactive approach, at each iteration the DM progressively articulates his or her preferences and the focus of the search becomes more confined. This way, the search converges toward a solution that maximizes the DM's utility – the best compromise solution. The major drawback of the interactive approach is that it requires an active and possibly lengthy involvement of the DM. Still, in cases with three or more criteria, the interactive approach might be the only viable option, since the complete set of alternatives and the trade-offs among them are usually too difficult for the DM to visualize, let alone to analyze and rank. Shin and Ravindran (1991) and Miettinen (1999, p.131-213) provide comprehensive discussions of these interactive methods.

With a bi-objective model such as the one discussed in this paper the DM can be spared this potentially lengthy interaction and need not define his or her preferences until the potential solution alternatives are identified. This approach allows the DM to explore all possible trade-offs between the two objectives – in this case, the net present value of the forest and the minimum area over time in large, mature patches. This approach provides the DM with a more holistic understanding of the trade-offs and more alternatives to choose from. This type of approach is called an 'a posteriori' approach in the operations research (OR) literature (Miettinen 1999, p. 63).

Thus, when objectives conflict, as in the spatially-explicit harvest scheduling problem discussed in this paper, it might be useful to identify the set of *Pareto Optimal*, or efficient, solutions – i.e., the potential management alternatives. An efficient solution (such as Point E in Figure 1), as opposed to a dominated solution (such as Point C in the figure), occurs when it is not possible to increase the attainment of one objective without reducing the attainment of another. Knowing the set of efficient solutions can help the DM understand the trade-offs between the competing objectives.

In a multi-objective optimization problem, the level of achievement of each objective defines each axis of the objective space (Figure 1). Because the problems in this paper are mixed-integer programming (MIP) problems, the set of attainable objective values, which can be represented in this space, is not a convex set. In fact, it is not a continuous set; it consists of a set of discrete points corresponding to the potentially large, but finite number of feasible solutions such as Points A, B, C, D, and E in Figure 1. The fact that this set is not convex requires us to distinguish between supported and non-supported Pareto optimal solutions. A series of weighted objective functions, where weights are assigned to each of the problem objectives and summed to obtain a single objective function value, can be used to identify the corner points of the convex hull of the efficient solution set, such as Points A and B in Figure 1. These points are commonly called supported strong (or strict) Pareto optima (T'kindt et al. 2002, p. 47). Efficient solutions that are not on the border of the convex hull, such as Point E in Figure 1, are called non-supported strict Pareto optima. Such optima will not be identified by a weighted objective function approach.

The set of strong Pareto optima, both supported and non-supported, define the outside (convex) corners of a line called the efficient frontier or trade-off curve. Points on the vertical or horizontal line segments between these corners may represent dominated solutions, such as Point D in Figure 1. However, there does not necessarily exist a solution at every point on these line segments due to the integer nature of the problem. Solutions on these line segments, such as the one represented at Point D, are called weak Pareto optima. The efficient frontier separates the region where additional efficient solutions are known not to exist from the region where dominated solutions may exist. Knowing the efficient frontier can be valuable to decision makers because it demonstrates the possible trade-offs between the objectives of a given problem.

When only two objectives are of interest, a 2-dimensional efficient frontier can be generated to describe the trade-offs between these objectives. Such curves can help determine which forest management plans will result in the best combination of achievements with respect to each goal. Importantly, trade-off curves allow the DM to assess the amount of one goal that must be given up in order to achieve a given increase in the amount of another goal. Trade-off curves for forest and wildlife management problems have been presented in Roise et al. (1990), Holland et al. (1994), Cox and Sullivan (1995), Arthaud and Rose (1996), Church et al. (1996) and (2000), Snyder and Revelle (1997), Williams (1998), and Richards and Gunn (2000). Cohon et al. (1979) developed a technique for approximating the efficient frontier for convex bi-criteria problems.

This research addresses the question of how to identify the efficient frontier as efficiently as possible for spatially-explicit harvest scheduling models where the set of solutions is not convex in the objective space. Efficiency is important because the time required to identify even a single efficient solution can be long (Miettinen 1999, p. 77). This is especially true with spatially-explicit models such as the one used in this paper. These models are typically formulated as mixed-integer programming (MIP) problems, which are, in general, $\mathcal{NP}$-Hard. Essentially this means that solution times may increase with problem size faster than any polynomial function of problem size. Wolsey (1998, p. 88) provides a more precise, but less intuitive, definition of the $\mathcal{NP}$-Hard property. This paper tests the performance of five traditional methods and one proposed method of generating the efficient frontier for a bi-criteria, spatially-explicit harvest scheduling problem.

The Bi-Criteria Formulation

This section describes the formulation of the example spatially-explicit harvest scheduling model. It includes harvest flow constraints, maximum harvest opening size constraints, constraints that define the minimum area of large, mature patch habitat over time, and a minimum average ending age constraint. The model formulation of the mature forest patch criterion is essentially the same as the one presented in Rebain and McDill (2003). Formulation of the maximum harvest area constraints is a generalization of the formulation presented in McDill et al. (2003).

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

$$Max\ \lambda \tag{2}$$

subject to:

$$X_{m0} + \sum_{t=h_m}^{T} X_{mt} \leq 1 \qquad \text{for } m = 1, 2, \ldots M \qquad (3)$$

$$\sum_{m \in M_{ht}} v_{mt} \cdot A_m \cdot X_{mt} - H_t = 0 \qquad \text{for } t = 1, 2, \ldots T \qquad (4)$$

$$b_{lt} H_t - H_{t+1} \leq 0 \qquad \text{for } t = 1, 2, \ldots T\text{-}1 \qquad (5)$$

$$-b_{ht} H_t + H_{t+1} \leq 0 \qquad \text{for } t = 1, 2, \ldots T\text{-}1 \qquad (6)$$

$$\sum_{m \in M_p} X_{mt} \leq n_{P_i} - 1 \qquad \text{for all } p \in P \text{ and } t = h_i, \ldots, T \qquad (7)$$

$$\sum_{j \in J_{mt}} X_{mj} - O_{mt} \geq 0 \qquad \text{for } m = 1, 2, \ldots M, \text{ and } t = 1, 2, \ldots, T \qquad (8)$$

$$\sum_{m \in M_c} O_{mt} - n_c B_{ct} \geq 0 \qquad \text{for } c \in C, \text{ and } t = 1, 2, \ldots, T \qquad (9)$$

$$\sum_{c \in C_m} B_{ct} - BO_{mt} \geq 0 \qquad \text{for } m = 1, 2, \ldots M, \text{ and } t = 1, 2, \ldots, T \qquad (10)$$

$$\sum_{m=1}^{M} A_m BO_{mt} \geq \lambda \qquad \text{for } t = 1,2,\ldots T \qquad (11)$$

$$\sum_{m=1}^{M} A_m [(Age_{0t}^{T} - \overline{Age}^{T}) X_{0t} + \sum_{t=h_m}^{T} (Age_{mt}^{T} - \overline{Age}^{T}) X_{mt}] \geq 0 \qquad (12)$$

$$X_{mt} \in \{0,1\} \qquad \text{for } m = 1, 2, \ldots M, \text{ and } t = 0, h_m, h_m + 1, \ldots, T \qquad (13)$$

$$B_{ct} \in \{0,1\} \qquad \text{for } c \in C, \; t = 1, 2, \ldots, T \qquad (14)$$

$$O_{mt}, \; BO_{mt} \in \{0,1\} \qquad \text{for } m = 1, 2, \ldots M, \text{ and } t = 0, 1, \ldots, T \qquad (15)$$

where X_{mt} = a binary decision variable whose value is 1 if management unit m is to be harvested in period t for $t = h_m, h_{m+1},\ldots,T$. In other words, X_{mt} represent a harvesting prescription for management unit m. 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} is the "do-nothing" alternative for management unit m). Note: in some cases, the index j is used to denote the harvest period. In these cases X_{mj} is the same as X_{mt} if $j = t$.

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;

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 stand at the end of the planning horizon;

A_m = the area of management unit m in hectares;

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*;

H_t = a continuous variable indicating the total volume of sawtimber in m^3 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 the youngest management unit in path i is old enough to be harvested;

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;

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;

B_{ct} = a binary variable whose value is 1 if all of the stands 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;

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 the first objective function 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. For age classes up to the optimal rotation, residual forest values are equal to the present value of the timber management costs and revenues on the management unit, assuming that it will be harvested at the optimal economic rotation, plus the present value of the land expectation value (LEV) representing future rotations. The LEV is the present value, per unit area, of the projected costs and revenues from an infinite series of identical even-aged forest rotations, starting initially from bare land. For age classes beyond the optimal economic rotation, residual forest values are equal to the liquidation value – i.e., the value of immediately harvesting the timber, plus the LEV for future rotations.

Equation (2) maximizes the minimum amount of total area in large, mature forest patches over the time periods in the planning horizon. This is the same objective specified by Rebain and McDill (2003a and 2003b). The logic of this objective is that the period with the least amount of habitat will represent the key bottleneck affecting the viability of populations of species that depend on this type of habitat. Such an objective automatically precludes the possibility of increasing the amount of mature patch habitat over what currently exists, however. In situations where the current amount of habitat is considered to be less than what is desirable a different formulation of the objective would be more appropriate.

Constraint set (3) consists of logical constraints that allow only one prescription to be assigned to a management unit, including a do-nothing prescription. Constraint sets (4), (5) and (6) are flow constraints.

Constraint set (7) 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, but the constraints can be modified easily to impose longer exclusion periods in integer multiples of the 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 (8)–(11) are the mature patch size constraints. Constraint set (8) determines 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. If any of these prescriptions have a value of 1, then O_{mt} may also equal 1, indicating that the management unit will be “old enough” in that period. One of these constraints is written for each management unit in each period.

Constraint set (9) determines whether or not a cluster of management units meets the minimum age requirement for mature patches. Clusters are defined here as groups of contiguous 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 (2003a). 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. If cluster c meets the age requirement in period t, then B_{ct} is allowed to take a value of 1. These constraints are written for each cluster in each period.

Constraint set (10) determines 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 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 at least one of the clusters it is a member of meets the age requirement in that period. Constraint set (11) specifies that the total mature patch area for each period must be larger than λ in all periods. Thus, λ cannot be larger than the area of large, mature forest patch habitat in any period. Equations (2) and (11) work together to capture the minimum amount of total area in the large, mature forest patches over all the time periods (the value of the variable λ) and to maximize this minimum area.

Constraint (12) 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 (13), (14) and (15) identify the stand prescription and mature patch size variables as binary.

Methods for Identifying The Efficient Frontier of the Bi-Criteria Model

Several approaches have been developed to generate the efficient solution set for discrete multi-criteria optimization problems. This section briefly describes the basic methods and any variations from the original algorithms used in this research. The methods are described here primarily from the perspective of the objective space.

Whenever either the units or the scale of the values of the objectives are different, these values must be normalized if a weighted objective function is used. In this research the ‘best value’ normalization approach was used, where the weight coefficients are divided by the corresponding elements of the ideal solution. The ideal solution is a vector whose elements are defined by the optimal attainment of the respective objective without regard to any of the other objectives (Figure 1). For example, the first element of the ideal solution vector for the example problem in this research is obtained by maximizing the net present value without regard to the minimum area of mature habitat; the second element is obtained by maximizing the minimum area of mature habitat without regard to the net present value. Clearly, the ideal solution is not attainable if the criteria conflict with one another. The ideal solution is identified and the criteria values are normalized in the initialization phase of each of the algorithms discussed below.

The Weighted Objective Function Method (P_λ)

Multiple-objective programming models, where the objective function is a weighted combination of multiple goals, have been applied to many forest and wildlife management planning problems (e.g., Roise et al. 1990, Hof and Joyce 1993, Snyder and Revelle 1997, and Williams 1998). As the name implies, the weighted objective function method assigns weights to each of the objectives and combines them into a single scalar objective function. One way to determine a set of efficient solutions while maximizing the weighted objectives is to utilize the scalar maximum problem, known as the P_λ problem, as proposed by Geoffrion (1968):

$$P_\lambda = \text{Max } \left\{ \sum_{i=1}^{P} \lambda_i f_i(x) : \sum_{i=1}^{P} \lambda_i = 1,\ \lambda_i \geq 0,\ x \in X \right\} \tag{15}$$

Where (15) maximizes the sum of the *P* objective functions, $f_i(x)$, weighted by scalars $\lambda_i \geq 0$, where the sum of the weights is 1, and the values of *x* satisfy the constraints of the problem as defined by feasible set *X*. As mentioned above, since the scales and/or the units of the objectives are typically different, the weights have to be normalized. Assigning all combinations of weights to the objective functions guarantees the identification of each efficient point provided the following conditions are met:

(1) *Theorem 1*: Let $\lambda_i > 0$ (i=1,..,*P*) be fixed. If x^0 is optimal for P_λ, then x^0 is an (properly[1]) efficient solution (Geoffrion 1968).

[1] The concept of proper efficiency eliminates the situation where for some criterion the marginal gain in one objective can be made arbitrarily large relative to the marginal losses in each of the remaining criteria (Geoffrion 1968).

(2) *Theorem 2*: Let X be a convex set, and let the f_i be concave on X. Then x^0 is properly efficient if and only if x^0 is optimal in P_λ for some λ with strictly positive components (Geoffrion 1968).

Since the above spatially-explicit harvest scheduling problem involves discrete (binary) decision variables, the feasible set X cannot be assumed to be convex for this problem. There is, therefore no guarantee that this method will generate all the efficient solutions. In fact, the Weighted Objective Function Method can only identify the supported strict Pareto optima. Nevertheless, the weighted objective function method can be used to create an initial set of solution alternatives. In an interactive approach, these alternatives may be presented to the DM, who can then specify the range within which further solution alternatives can be sought using some other method.

A modification of a well-known algorithm (c.f. Eswaran 1989) was used here to decompose the weight space into sections (line segments in the bi-criteria case) that correspond to the same efficient solutions. In an ideal application of this method, a section can be eliminated from further exploration whenever its end points result in the same solution. However, the algorithm had to be modified slightly because large-scale problems cannot always be solved to exact optimality. The problems were solved with CPLEX 8.1, which uses a branch-and-cut algorithm to solve MIP problems. CPLEX was instructed to stop when the optimality gap – the percentage difference between the objective function value of the current best integer solution and the dual bound (Williams 1998, McDill and Braze 2001) – reached 0.001%. While this is a very conservative stopping rule – the default value in CPLEX is 0.01% – there were cases where the solution found with one weight combination dominated the solution found with an adjacent weight combination. By definition, the dominant solution would be better for any weight combination, so the dominant solution was assumed to be the optimal solution for both weight combinations, and also for any weight combination in between them, and the line segment between the two weight combinations was not explored further. If the solutions corresponding to the end points of the line segment were different and neither dominated the other, a new weight combination was generated by calculating the mean of the two weight combinations at the end points. The new solution for the new weight combination was then compared with the solutions for the neighboring weight combinations to determine whether the subsections on the other side of the new weight combination could be eliminated from further consideration. The algorithm was terminated when there were no sections left to decompose. This process is referred to as the decomposition algorithm; similar decomposition algorithms are used in some of the other methods described below.

The ε-Constraining Method

This approach, introduced in Haimes et al. (1971), involves the following steps.

Step (1): determine the ideal solution by optimizing each objective without regard to the other. Call these optimal values Maximum Net Present Value (MNPV) and Maximum HABitat (MHAB), respectively.

Step (2a): maximize NPV while constraining the minimum amount of large, mature habitat over all periods (HAB) to be larger than or equal to MHAB, and **b)** maximize HAB while constraining NPV to be larger than or equal to MNPV. This results in two efficient solutions that define the two ends of the efficient frontier. The remaining efficient solutions will be found within the rectangle defined by these two points.

Step (3): choose a point on one of the criteria axes within the interval defined by the two points found in Step 2 (we chose the HAB axis). Call this value $\overline{HAB}$. Maximize the other objective (NPV) on the feasible set, subject to an additional constraint that restricts HAB to be larger than or equal to $\overline{HAB}$. Unfortunately, this solution (call it $NPV_{\overline{HAB}}$) might only be a weak Pareto optimal solution. Therefore, a fourth step is necessary to either confirm the efficiency of $NPV_{\overline{HAB}}$ or find a solution that is efficient and dominates $NPV_{\overline{HAB}}$.

Step (4): Maximize HAB subject to the usual constraints, plus a constraint that requires NPV to be larger than or equal to $NPV_{\overline{HAB}}$. Call this problem $P_{\overline{HAB}}$. According to Sadagopan et al.'s (1982) *Theorem 2*, any solution that solves this problem is an efficient solution. This theorem enables us to find all efficient solutions by parametrically solving $P_{\overline{HAB}}$ for different values of $\overline{HAB}$ ($0 < \overline{HAB} < MHAB$).

The algorithm used in this research, outlined in Figure 2, makes use of this theorem by gradually proceeding from one end of the efficient frontier to the other. The first two steps are the same as above. Step 3 is to maximize NPV subject to a constraint that requires HAB to be larger than or equal to the HAB value from the previous solution plus a sufficiently small δ. At the first iteration, this HAB value is equal to the objective function value of the solution that maximized HAB while constraining NPV to be larger than or equal to MNPV (Step 2b). The small δ is necessary to avoid the same solution that was obtained in the previous step. Of course, this value introduces the possibility that the algorithm will miss solutions that are within the interval defined by the arbitrary δ. Step 4 is to maximize HAB subject to a constraint that restricts NPV to be larger than or equal to the NPV value obtained in step 3. The algorithm terminates when the HAB value reaches MHAB.

The Decomposition Method based on the Tchebycheff-Metric

Eswaran et al. (1989) proposed a procedure to generate the entire efficient solution set for non-linear integer bi-criteria problems that uses the Tchebycheff metric and solves the following, so-called, P_β problem for all parametric values of β:

$$P_\beta = \underset{x \in X}{Min}\left\{ \left\| f(x) - \bar{y} \right\|_\beta : \left\| f(x) - \bar{y} \right\|_\beta = \max_i \beta_i \left| f_i(x) - \bar{y}_i \right|,\ \sum_{i=1}^{P} \beta_i = 1,\ x \in X \right\} \qquad (16)$$

where $\left\| f(x) - \bar{y} \right\|_\beta = \max_i \beta_i \left| f_i(x) - \bar{y}_i \right|$ is the weighted Tchebycheff-metric, $\bar{y}$ represents the ideal solution vector, $\bar{y}_i$ represents the ideal value of objective *i*, β_i is the weight parameter corresponding to objective *i*, and $f_i(x)$ is objective function *i*.

All the solutions identified by the parametric decomposition of the β-space, which is analogous to parametric programming, are efficient solutions if the following sufficient condition, Bowman's *Theorem 4*, is met: if an efficient set is uniformly dominant, then all the solutions to the P_β problem are efficient points (Bowman 1975). An efficient set is said to be uniformly dominant if, for every dominated point $x^d \in X$, there exists an efficient point $x^* \in X$ such that $f_i(x^d) < f_i(x^*)$, for all i (Bowman 1975). In other words, Bowman's *Theorem 4* is upheld only when there are no weak Pareto optima (such as Point D in Figure 1). Since it is not possible in general to determine a priori whether weak Pareto optima exist for a given problem, we cannot conclude that the decomposition method based on the Tchebycheff metric will always identify strictly efficient solutions.

The same decomposition algorithm discussed in the Weighted Objective Function Method section above was used to decompose the weight space of the Tchebycheff-Metric. As discussed above, the algorithm applied here differs slightly from the one described by Eswaran et al. (1989) because we did not solve every problem to full optimality. Eswaran et al. (1989) assumed that all problems would be solved to optimality, so their algorithm eliminates sections of the weight space only when the end points result in the same solution. Our algorithm eliminates sections of the weight space either when the end points result in the same solution or when the solution at one end point dominates the solution at the other end point. In the latter cases, the dominant solution was assumed to be the optimal solution at both end points and for all points in between.

Hybrid Methods

A number of hybrid methods have been described in the operations research literature that, by combining some of the above basic approaches, efficiently utilize the positive features of more than one method. For example Wendell and Lee (1977) combined the Weighted Objective Function Method with the ε-Constraining Method (Wendell et al, 1977). They fixed the weight coefficients, λ_i and parametrically solved the problem below for each ε_i. The advantage of the method is that the weight coefficients do not have to be altered.

$$\text{Max } P_{hybrid} = Max\left\{\sum_{i=1}^{P}\lambda_i f_i(x):\ \ f_i(x) \geq \varepsilon_i,\ \sum_{i=1}^{P}\lambda_i = 1,\ \ \lambda_i \geq 0,\ \ x \in X\right\} \tag{18}$$

for all $i = 1,...,P$, where $f_i(x)$ is objective function *i*. Both of the algorithms below can be thought of as special cases of the hybrid method introduced by Wendell and Lee (1977). The Alpha-Delta Method was developed by the authors, and the Triangles Method is from Chalmet et al. (1986).

The Alpha-Delta Method

This approach takes advantage of the fact that if we assign a substantially larger weight to one objective than to the other, strong Pareto optima can be identified consecutively along the efficient frontier using a procedure similar to the ε-Constraining method. Figure 3 illustrates this process. The initialization phase is the same as in the Decomposition Method based on the Tchebycheff-metric: calculate the ideal solution and then the two end points of the efficient frontier, (EFS(1) and EFS(2)), as in Figure 2. A very large weight is then assigned to one objective and a minimal weight to the other. In Figure 3, $\overline{PQ}$ demonstrates such an allocation of weights. From here on, a combined objective function with a large weight assigned to one objective (NPV in our case) and a small weight assigned to the other objective (HAB) is maximized at each step subject to a constraint that requires the achievement value of the other objective (HAB) to be greater than or equal to the achievement value obtained by the previous step plus a sufficiently small δ. At the first iteration, this achievement value is equal to the objective function value of the solution that maximized the HAB objective while constraining NPV to be larger than or equal to MNPV (MNPV is the first element of the ideal solution vector). The small δ ensures that a new solution will be found. For example, using the weighted objective function $\overline{PQ}$ in Figure 3a, Point A would be picked up repeatedly if the lower bound on HAB were not augmented by δ (*AHab* + δ). Instead, Point B will be found next (Figure 3a). The next iteration is implemented using the new lower bound of (*BHab* + δ), where the *BHab* value was obtained in the previous step.

The parameter δ has to be set to a small value to minimize the probability that efficient points will be missed. In Figure 3b, for example, Point C would be missed if δ were not reduced. Similarly, the parameter α (the slope of the weighted objective function) has to be small to minimize the probability that an efficient point will be missed. The algorithm terminates when the achievement value of the habitat reaches its upper bound (MHAB in our case). The advantage of this algorithm is that the new solution at each step will always neighbor the previous one along the efficient frontier if sufficiently small α and δ are used, and, while the ε-Constraining Method finds each new solution in two steps, this approach will do it in one.

The Triangles Method

This algorithm, developed by Chalmet et al. (1986), seeks Pareto optimal solutions between two adjacent, efficient points that have already been identified (e.g., solutions A and B in Figure 4). The weight coefficients on the objective functions are fixed and arbirary. We used equal weights for both objectives in this research. At each step, the search space (the gray area in Figure 4) is confined by two constraints. These constraints are gained by adding a small δ_1 and δ_2 to the lower achievements on the two objectives at the two adjacent solutions. In Figure 4, for example, the section of the efficient frontier between Point A and Point B is explored (the gray area in Figure 4a). A section between two adjacent efficient points will be eliminated from further investigation if no feasible solution is found there (such as in Figure 4b) or, alternatively, if the difference in one of the objective values between the two solutions is smaller than a predetermined limit. The algorithm terminates when there are no sections left to explore. Again, δ_1 and δ_2 have to be small to minimize the possibility that the algorithm will miss an efficient solution. As an example, in Figure 4b, Point C would be missed if the value of δ_1 were not reduced.

A case study

In order to illustrate and test the performance of the various algorithms for generating the efficient frontier, an example hypothetical forest was created. This forest consisted of 50 stands 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 stand 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 stand were: cut the management unit in period 1, period 2, or period 3, or do not cut it at all. The minimum rotation age was 60 years. A maximum harvest opening size of 40 ha was imposed, and adjacent stands were allowed to be harvested concurrently as long as they did not violate this maximum opening size. All management units are smaller than the maximum harvest opening size. The wildlife species under consideration is assumed to need habitat patches that are at least 50 ha in size and at least 60 years old. Since the minimum habitat patch size is greater than the maximum harvest opening size, these patches must be composed of more than one management unit. There were 139 paths and 539 clusters associated with the model formulation of the test problem.

We implemented the algorithms described in the Methods section using CPLEX 8.1 (ILOG CPLEX 2002) on a Dual-AMD Athlon ™ MP 2400+ (2.00 GHz) computer with 2.0 GB RAM. Programs to automate the algorithms were written in Microsoft Visual Basic 6 using the ILOG CPLEX Callable Libraries. The relative MIP gap tolerance parameter (optimality gap) was set to 0.00001 (0.001%), and the MIP variable selection strategy parameter was set to '3' (i.e., strong branching). The precise setting of the optimality gap was needed to minimize the chances of obtaining dominated solutions. The multi-objective techniques described above assume that each sub-problem is solved to full optimality. Achieving full optimality, however, is unrealistic even for small-scale harvest scheduling problems such as the one presented in this paper. This is why a compromise value was chosen. Lastly, the following parameter settings were used in the respective multiple-objective algorithms: $\varepsilon = 0.1$ ha in the ε-Constraining Method; $\alpha = 0.01°$ and $\delta = 0.1$ ha in the Alpha-Delta Method; and $\delta_1 = 0.1$ ha and $\delta_2 = \$1$ in the Triangles Method. The Weighted, the Triangles and the Tchebycheff Decomposition methods were terminated either if there were no weight segments left to decompose or after 60 hours of CPU time.

The experiment addressed the following questions: (1) How many of the efficient solutions can each algorithm identify? (2) How long does each algorithm take to identify all of the solutions that are found? and (3) How good are these solutions in terms of optimality? The third question refers to the fact that even though the optimality gap was set to 0.001% for each algorithm some methods might consistently generate solutions that are better than the ones generated by other methods but still within this range.

Results and Discussion

Figure 5 shows the efficient frontier generated by the various methods. The DMs, if confronted by these alternative solutions, will see that the trade-offs are relatively flat between Alternatives A and E, and between B and C. They would likely prefer E or C to A or B, because these solutions produce considerably more habitat while only a small amount of profit is forgone. Because E and C are far apart, however, they may be interested in a non-supported compromise solution such as H. Also, the DMs might be interested in a cluster of alternatives, such as those around Points F and G that, if implemented, would produce similar amounts of profit and habitat. They might be interested in a third decision factor (something other than profit- or habitat maximization) that could potentially tip the balance in favor of one of these solutions. This solution might or might not be a supported Pareto-optimum – e.g., not necessarily Point F or G that can be found by the Weighted Method. Moreover, as there are plenty of efficient solutions along the frontier, DMs with conflicting interests could select the best compromise solution from a good pool of alternatives.

The Weighted Objective Function Method identified only six efficient solutions (points A, E, C, F, G, D on Figure 5). This method misses the majority of the efficient solutions because most of the efficient solutions in this case are non-supported Pareto optimal solutions. It is hard to say, without looking at a large number of problems, whether this is a typical situation or not. Furthermore, in general, supported solutions are more likely to be the most desirable compromise solutions than non-supported solutions. However, it is clear that one cannot be sure that desirable non-supported solutions do not exist unless one looks for them, and they cannot be found with the Weighted Objective

Function Method. This is the fundamental drawback with relying only on the Weighted Objective Function Method.

The ε-Constraining, the Alpha-Delta, and the Triangles Methods all found the highest number of efficient solutions (36). In terms of solution times, however, the Alpha-Delta Method was considerably faster than the others (6.27 hrs), followed by the Triangles Method (17.13 hrs), and then the ε-Constraining Method (58.75 hrs). The Tchebycheff Decomposition Method found 34 solutions in 36.83 hours, while the Weighted Method found 6 in 1.72 hours.

Table 1 summarizes the set of efficient solutions. In terms of optimality, the ε-Constraining and the Triangles Methods performed the best. The Alpha-Delta Method produced the same solutions as those generated by the ε-Constraining and the Triangles Methods in all but five cases. In those cases, the achievements of the NPV objective obtained with the Alpha-Delta Method were slightly less (Table 2). In 3 out of 34 cases, the Tchebycheff Decomposition Method resulted in lower NPV achievements than the ε-Constraining or the Triangles Methods. As the greatest difference in NPV was only 0.0254%, these differences are probably not a significant concern. However, it is noteworthy that the differences consistently favor some algorithms over others. In the case of the Tchebycheff Decomposition Method, for instance, there is a simple explanation for the lower attainments. As described earlier (Equation 16), this method minimizes the maximum (weighted) difference in the attainments of the respective objectives between two solutions, one of which is the ideal solution. Once the maximum of these weighted differences is minimized, there is no incentive to further reduce the value of the other differences. This is why the attainment values on one objective (the NPV of the forest in this case) can be sub-optimal. By using another metric, the L_1 Metric for example, which measures the weighted sum (instead of the weighted maximum) of the differences in the attainments on the respective objectives, this problem can easily be overcome.

Table 1 also provides information on how the optimal solution changes along the efficient frontier. The rightmost columns show the IDs of those management units that form mature forest patches in each planning period. It is noteworthy that, given the various optimal harvest schedules, combinations of almost half of the units (24) may become old enough to be part of a patch over the planning horizon. The results suggest that a small change in management decisions, such as to cut a particular unit instead of another one, may lead to a much better achievement on one objective at a minimal loss on the other objective. One such example would be choosing harvest schedule no. 9 instead of no. 8. It is also worth pointing out that in most of the cases (23 out of 36), Period 1 was the constraining time period in producing mature forest habitat. This is not surprising, particularly for higher levels of habitat production, since the available mature habitat in Period 1 cannot be increased beyond 169.84 ha and harvesting decisions can only decrease the amount of mature habitat in that period. There is more flexibility to arrange harvesting decisions in earlier periods to create large mature patches in later periods.

Figure 6 shows the cumulative time required to obtain each solution for each method. The figure clearly shows that the Alpha-Delta Method dominates the others in terms of solution time, finding all 36 efficient solutions in a little more than six hours. The figure also shows that the Alpha-Delta and the Triangles Methods found all 36 efficient solutions well before the ε-Constraining found five. Although the Weighted Objective Functions Method identified only six of the efficient points, these six points were found relatively quickly and this "filtered" set of alternatives are most likely to be among those that are most preferred and might be useful for interactive methods involving the DM or to find a good, distributed set of alternative solutions if solution time is a constraint. There is no guarantee, however, that the set of solutions found with this method will be evenly distributed along the efficient frontier. An additional advantage of the Weighted Method is that, unlike the other approaches, it does not require adding new constraints to the original problem and thus it preserves the original constraint structure (ReVelle 1993). This can be a huge benefit in polynomially-solvable integer programming problems that have special constraint structures, such as total unimodularity (Wolsey 1998, p.38). This structure would be destroyed if the other methods were used. This is unlikely to be the case, however, in realistic problems, where a large variety of constraints will likely be imposed in the model.

Since the Weighted Method can only identify supported solutions, its success in finding a sufficient number of efficient alternatives for larger problems depends on the proportion of supported vs. non-supported Pareto-optima. As the efficient frontier can take many shapes, the proportion of supported solutions is problem-dependent and hard to foresee. The frontier can be a strictly concave curve with only two supported solutions (and hundreds of non-supported solutions). On the other hand, the frontier may consist of many concave and convex segments, in which case many supported Pareto-optima may exist. Another factor that will influence the usefulness of the Weighted Method is the distribution of the supported solutions. If there are large gaps between supported solutions, some other method will be needed in order to explore those gaps further.

While none of the methods can guarantee that they will find all of the efficient solutions to a problem, three of the five methods that were tested found essentially the same set of 36 efficient solutions. The Tchebycheff Decomposition algorithm does not guarantee the identification of the complete set of efficient alternatives unless the uniformly dominant property of the feasible set of the harvest scheduling problem holds. This method failed to find two of the 36 efficient solutions found by the other methods. In general, it is not likely that the uniformly dominant property will hold for problems like the example problem used here. For example, there generally are a very large number of feasible solutions with a given minimum area of mature patch habitat in one period, but with varying net present values. It is hard to predict how many efficient solutions this method would actually find for any given problem. In contrast, by adjusting the parameters of the ε-Constraining (parameter δ), Alpha-Delta (parameters δ and α) or Triangles (parameters δ_1 and δ_2) Methods, one can reduce the probability of missing any of the solutions with minimal additional computational cost. It is likely that, for similar parameter settings, the chance that the ε-Constraining Method will miss an efficient solution is lower than the chance of missing a solution with Alpha-Delta or Triangles Methods as the former has only one parameter – and hence only one area – that controls the size of the area where missed solutions might exist. In addition to not finding all of the efficient solutions, the time required by the Tchebycheff Method to find the 34 efficient solutions that it found was substantially longer than the time required by either the Alpha-Delta or the Triangles Methods to find 36 efficient solutions.

The potential of the methods discussed here to generalize to the tri-criteria case is difficult to assess without actually applying them to a specific case. However, a few observations can be made at this time with regard to this issue. First, the Weighted and the Tchebycheff Methods are relatively easy to generalize by decomposing the weight space, which is a triangle in the 3-objective case, into so-called indifference regions that lead to identical efficient solutions. However, as the results in this paper suggest, these methods can miss efficient solutions. Generalizing the other three algorithms to deal with three objectives appears to be quite complicated, but still possible. Each approach would require adding a potentially large number of constraints to the problem at each iteration. For the Alpha-Delta and the ε-Constraining Methods, the set of constraints that one might add to the formulation at each step would form a non-convex feasible region in the objective space. By introducing a set of binary variables, this region can easily be described within one formulation. Generalizing the ε-Constraining approach to more than two objectives would be particularly expensive computationally since, at each step, an efficient solution can only be obtained after solving three sub-problems for the tri-criteria case (*n* sub-problems for the *n*-criteria case). This is the only way, however, to ensure that the final solution is not dominated. This problem is avoided by using a weighted objective function with non-zero, fixed weights in the Alpha-Delta algorithm. In our experience, the Alpha-Delta Method has the further advantage, over the other methods, of being very simple to translate into computer code.

In some situations when computer time is a constraint but involving the DMs in the planning process is possible, the Weighted- and the Alpha-Delta methods could be combined. The forest manager could use the Weighted Method first to obtain a rough estimate of the trade-offs relatively quickly, and present this initial set of solutions to the DMs. If the DMs are not satisfied with any of these initial solutions, then certain segments of the efficient frontier could further be explored in line with the DMs' interests using the Alpha-Delta Method.

A larger number of efficient solutions are likely to exist for problems with more stands and more area. In fact, the number of efficient solutions could explode as the problem size increases due to the combinatorial nature of spatially-explicit harvest scheduling. This would make the discussed methods computationally very expensive or even intractable. In these cases, it may be that the Weighted Objectives method will provide a large number of well-distributed solutions, and the other approaches would not be needed. However, there is no guarantee this will occur. One way of reducing solution times is by widening the optimality tolerance gap. This would, however, increase the likelihood of obtaining dominated solutions. Another option is to increase the spacing of the efficient solutions in the objective space, which can be done by increasing the values of parameters α and δ in the Alpha-Delta-, ε in the ε-Constraining Algorithm, or by changing the stopping rule for the other methods. This approach can reduce the cumulative solution times without jeopardizing the Pareto-optimality of the individual solutions. If, however, one or more of the individual IP sub-problems are intractable, then increasing the optimality tolerance gap is likely to be the only workable solution.

Conclusions

The multi-criteria optimization techniques discussed in this paper provide useful alternatives to goal programming or other multi-criteria approaches when the decision maker does not have a prior understanding of the potentials for and trade-offs between the conflicting objectives and therefore cannot readily specify preferences or a list of targets for the objectives. This situation occurs frequently in forest planning. Target values or preferences for criteria that describe wildlife habitat goals, such as the overall area to maintain in mature forest patches or the amount of edges within a given landscape, are often hard to specify a priori. By providing exact information on the nature of the trade-offs between such conflicting criteria, the methods discussed above would help the DM select the best compromise solution and give him or her more insight into the problem. We believe that this is the primary value of the "frontier" methods described here. In those situations where the DM is confident about what the targets should be, these computationally expensive methods may not be appropriate.

The following conclusions are suggested by the theoretical discussion and the analysis of the test problem in this paper: (1) If a complete set of efficient solutions is desired, the discrete nature of the harvest scheduling problem rules out the Weighted Objective Function Method as a useful approach because many efficient solutions may be missed; (2) In the bi-criteria case, the ε-Constraining, the Tchebycheff Decomposition, the Alpha-Delta, and the Triangles Methods are all capable of identifying a very good set of solutions; (3) The Alpha-Delta, and the Triangles Methods performed the best in terms of solution times for the test problem; (4) There were infrequent and minor differences in how the different algorithms performed in terms of solution optimality, but when differences occurred, the ε-Constraining and the Triangles Method consistently performed better than the other methods; (5) In our experience, the Alpha-Delta Method is the easiest to translate into computer code; (6) Although each of the methods can be generalized to the tri-criteria case, the Alpha-Delta Method appears to generalize the most easily.

Rigorous additional experimentation would be needed to determine whether these results would apply to a wide range of forest planning problems of various scales and various structures. The primary avenue of future research, however, points to the development of algorithms that would efficiently tackle the general, n-criteria case for discrete formulations such as the spatially-explicit harvest scheduling problems. As multiple-use forest management becomes more important for society, multi-objective optimization techniques, such as the "frontier" methods discussed here, will probably receive more attention in the future. Furthermore, their potential will rapidly expand as the performance of both optimization software and computer hardware improves. Currently, however, these techniques can only be applied at a small-scale, pilot-study level. These pilot studies, however, are very important in testing and fine-tuning these models before applying them to on-the-ground forest planning with real constituents. Analyzing small models can provide valuable insights for the forest planner about the trade-offs in similar but computationally less tractable, large-scale problems. It is also likely that the interactive utilization of the “frontier” methods has a lot of potential for multi-criteria forest planning, as the involvement of various stakeholders in the decision making process will become increasingly important.

References:

Arp, P.A., and D.R. Lavigne. 1982. Planning with goal programming: a case study for multiple-use of forested land. The Forestry Chronicle. October:225-232.

Arthaud, G. J., and 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. 26:2191-2200.

Bettinger, P., J. Sessions, and K. Boston. 1997. Using Tabu Search to schedule timber harvests subject to spatial wildlife goals for big game. Eco. Mod. 94: 111-123.

Bowman, V.J. Jr. 1975. On the relationship of Tchebycheff norm and the efficient frontier of multiple-criteria objectives. Notes in Econ. and Math. Systems. Vol. 130:76-85.

Carter, D.R., M. Vogiatzis, C.B. Moss, and L.G. Arvanitis. 1997. Ecosystem management or infeasible guidelines ? Implications of adjacency restrictions for wildlife habitat and timber production. Can. J. For. Res. 27 :1302-1310.

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

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

Church, R.L., 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.

Cohon, J.L., R.L. Church, and D.P. Sheer. 1979. Generating Multiobjective Trade-Offs : An Algorithm for Bicriterion Problems. Water Resour. Res. 15(5): 1001-1010.

Cox, E. S., and J. Sullivan. 1995. Harvest scheduling with spatial wildlife constraints : an empirical examination of tradeoffs. J. of Envir. Mgmt. 43:333-348.

Davis, L. S., and G. Lui. 1991. Integrated forest planning across multiple ownerships and decision makers. For. Sci. 37:200-226.

Eswaran, P.K., A. Ravindran, and Moskowitz. 1989. Algorithms for nonlinear integer bicriterion problems. J. Optimization Theory and Appl. 63:261-279.

Fedkiw, J. 1997. The Forest Service's pathway towards ecosystem management. J. For. 95(1):30-34.

Field, D.B. 1973. Goal programming for forest management. For. Sci. 19(2):125-135.

Field, R.C., P.E. Dress, and J.C. Fortson. 1980. Complementary linear and goal programming procedures for timber harvest scheduling. For. Sci. 26(1):121-133.

Franklin, J. F., and R. T. Forman. 1987. Creating landscape patterns by forest cutting: ecological consequences and principles. Land. Ecol. 1 :5-18.

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

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 Transactions on Systems, Man, and Cybernetics 1 :296-297.

Harris, L. D. 1984. The Fragmented Forest: Island Biogeography Theory and the Preservation of Biotic Diversity. The University of Chicago Press, Chicago, IL. 211 p.

Hof, J., and L. Joyce. 1993. A mixed integer linear programming approach for spatially optimizing wildlife and timber in managed forest ecosystems. For. Sci. 39(4):816-834.

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. 24:1260-1265.

Hotvedt, J. E. 1983. Application of linear goal programming to forest harvest scheduling. South. J. Agric. Econ. 15:103-108.

ILOG S. A. 2002. ILOG CPLEX 8.1 Reference Manual. ILOG Optimization Documentation.

Jones, J. G., B. J. Meneghin, and M. W. Kirby. 1979. Formulating adjacency constraints in linear optimization models for scheduling projects in tactical planning. For. Sci. 37(5):1283-1297.

Kao, C., and J. D. Brodie. 1979. Goal programming for reconciling economic, even-flow, and regulation objectives in forest harvest scheduling. Can. J. For. Res. 9:525-531.

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. 5:485-491.

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

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. Trans. N. Amer. Wildl. and Natur. Resour. Conf. 46:488-500.

Mendoza, G.A. 1897. Goal programming formulations and extensions, an overview and analysis. Can. J. For. Res. 17:575-581.

Miettinen, K.M. 1999. Non-linear Multiobjective Optimization. Kluwer Academic Publishers, Boston/London/Dordrecht. 298 p.

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

Murray, A.T., and R. L. Church. 1996b. Constructing and selecting adjacency constraints. INFOR 34(3):232-248.

Murray, A.T. 1999. Spatial restrictions in harvest scheduling. For. Sci. 45(1):45-52.

Pukkala, T. (ed.). 2002. Multi-objective Forest Planning. Kluwer Academic Publishers, Dordrecht /Boston/London. 207 p.

Ravindran, A. 2003. Multiple Criteria Optimization. Lecture Notes. Penn State University (unpublished).

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

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

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

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

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.

Rustagi, K. P, and B. B. Bare. 1987. Resolving multiple goal conflicts with interactive goal programming. Can J. For. Res. 17:1401-1407.

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

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

Snyder, S. and C. ReVelle. 1996b. The grid packing problem: Selecting a harvest pattern in an area with forbidden regions. For. Sci. 42(1):27-34.

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

Snyder, S. and C. ReVelle. 1997b. Dynamic selection of harvests with adjacency restrictions: The SHARe Model. For. Sci. 43(2):213-222.

Thompson, E.F., B.G. Halterman, T.J. Lyon, and R.L. Miller. 1973. Integrating timber and wildlife management planning. The Forestry Chronicle. Dec (1973):247-250.

T'kindt, V. and J. Billaut. 2002. Multicriteria Scheduling : Theory, Models and Algorithms. Springer, New York. 303 p.

Wendell, R.E., and D.N. Lee. 1977. Efficiency in multiple objective optimization problems. Math. Programming 12(3):406-414.

Williams, J. C. 1998. Delineating protected wildlife corridors with multi-objective programming. Envir. Mod. and Assess. 3:77-86.

Wolsey, L.A. 1998. Integer Programming. Wiley-Interscience, New York. 266p.

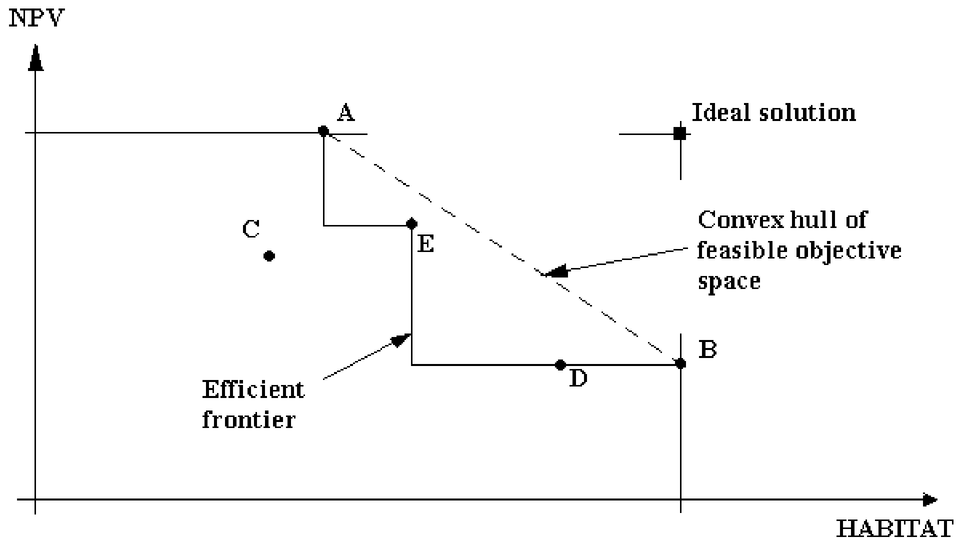

Figure 1. Multi-criteria optimization terminology.

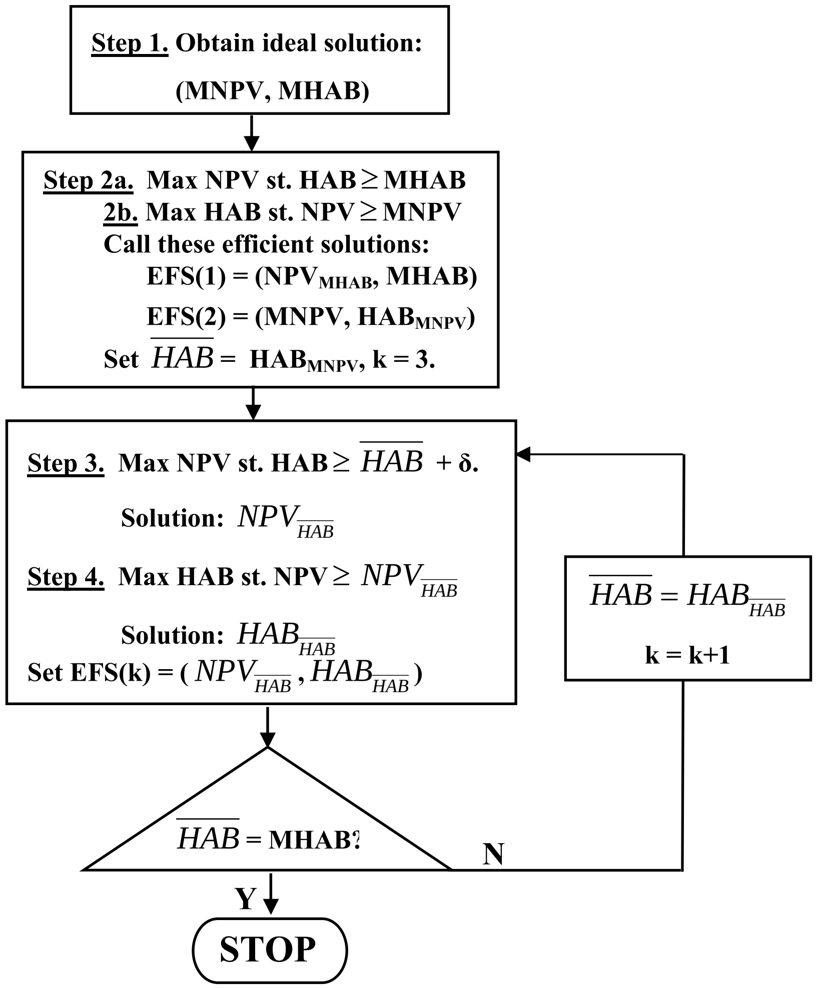

Figure 2. The ε-Constraining Algorithm.

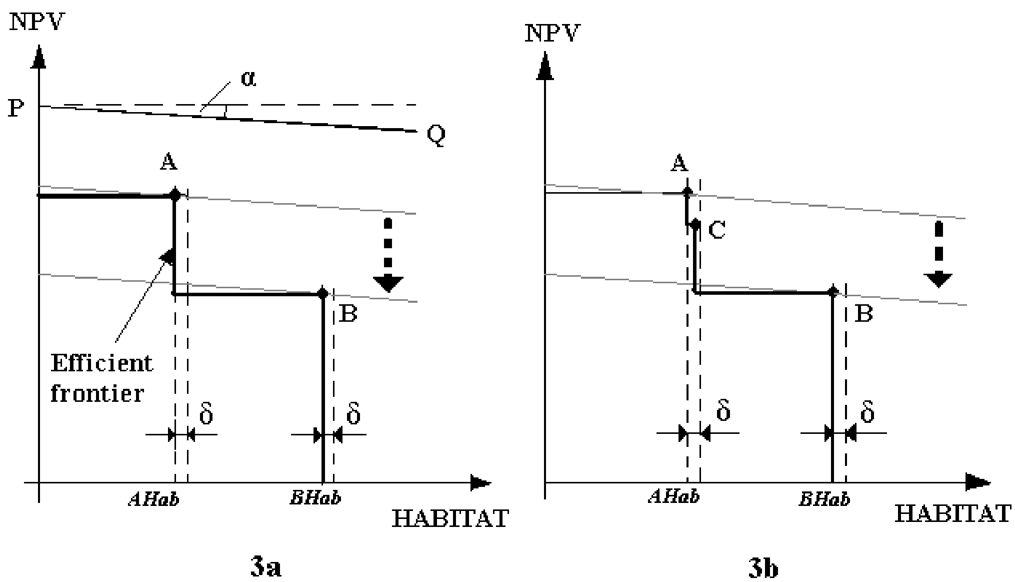

Figure 3. The Alpha-Delta Method.

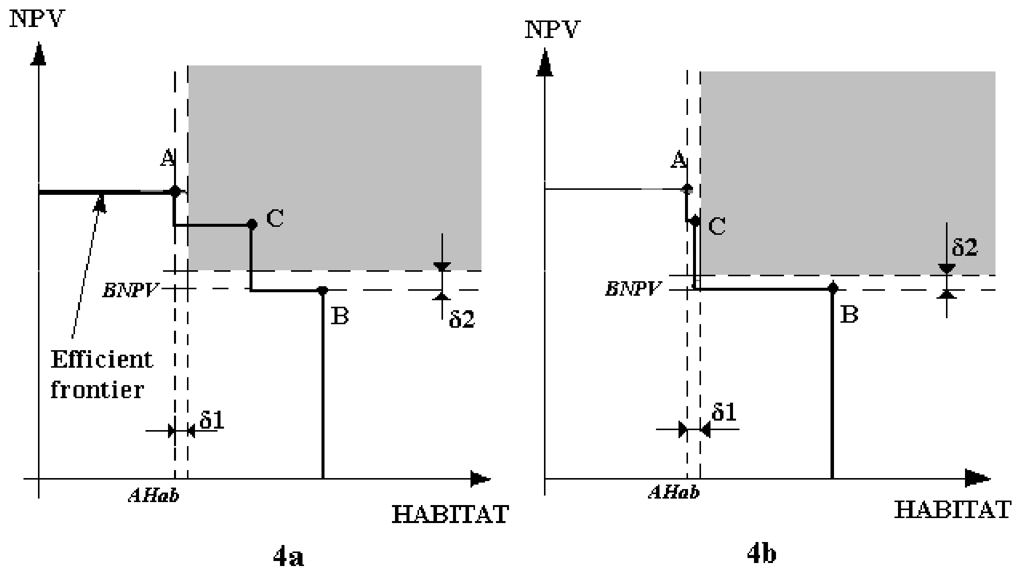

Figure 4. The Triangles Method.

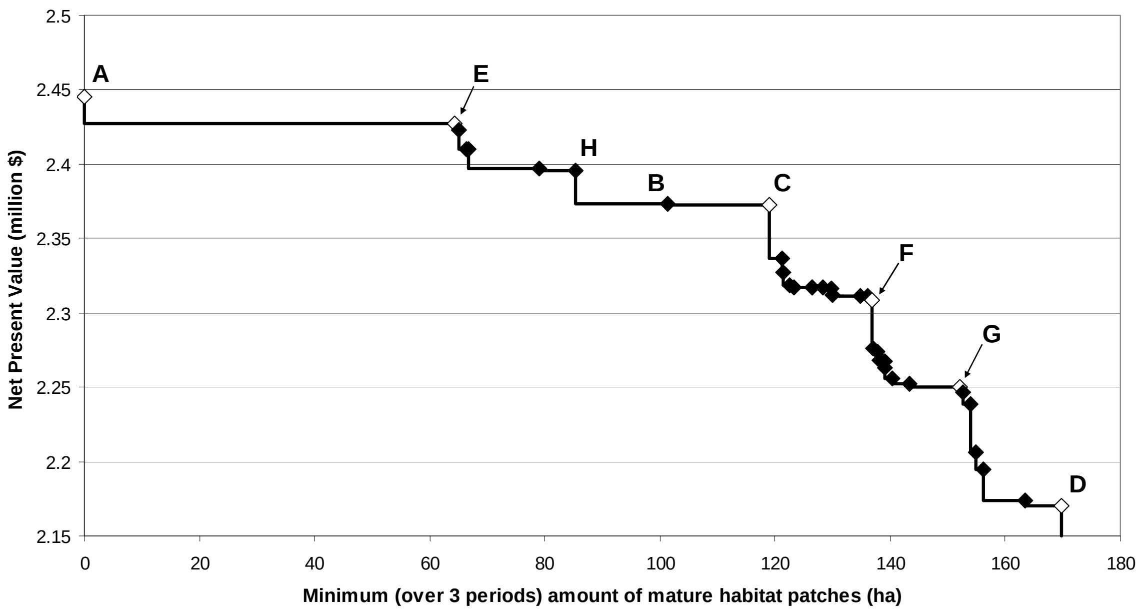

Figure 5. Efficient frontier of the bi-criteria harvest scheduling problem (supported solutions are shown with open markers).

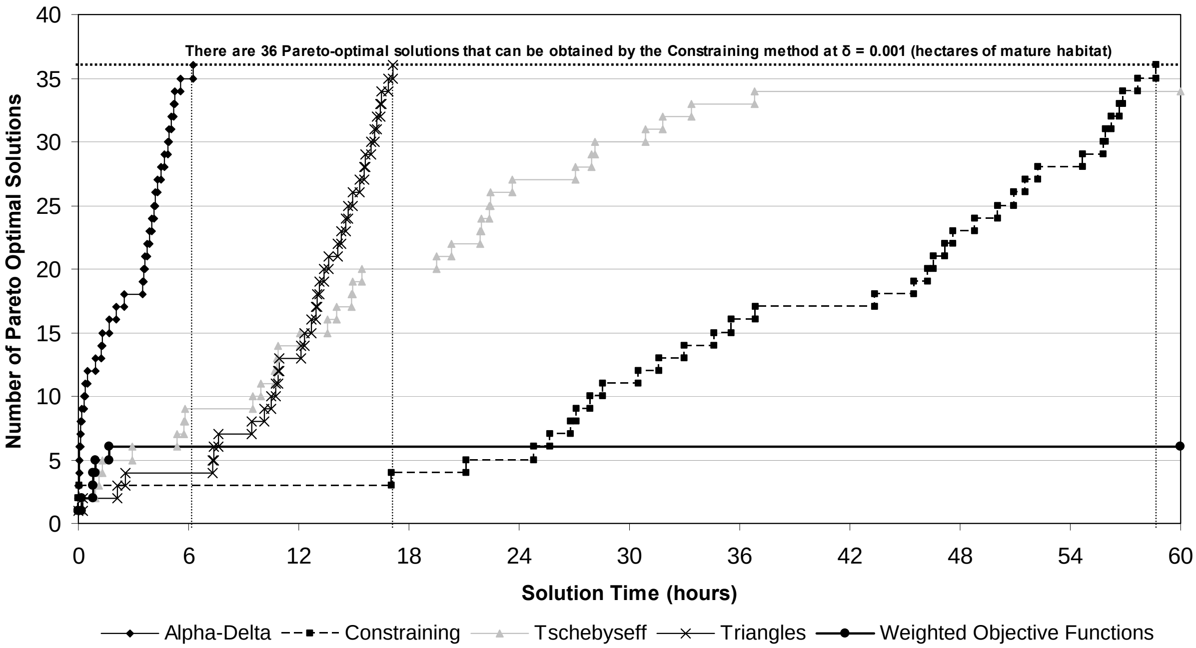

Figure 6. Cumulative solution times for each method.

Table 1. The set of efficient solutions

Efficient solutions			Constraining period	Efficient points missed	Management units in mature forest patches		
No.	NPV ($)	Habitat (ha)			Period 1	Period 2	Period 3
1	2,445,084	0	-		-	-	-
2	2,427,424	64.3817	3		13, 18, 23, 43	5, 14, 48	7, 44, 48
3	2,423,119	64.9906	2	W	13, 18, 23, 43	5, 14, 48	5, 44, 48
4	2,410,088	66.3003	3	Ts, W	13, 18, 23, 43	5, 6, 11, 48	5, 44, 48
5	2,409,663	66.7425	1	W	13, 18, 23, 43	5, 6, 11, 48	5, 6, 7, 44
6	2,397,375	79.0436	3	W	2, 6, 28	5, 6, 11, 14, 48	7, 14, 44, 48
7	2,395,672	85.2725	1	W	2, 6, 29	5, 6, 11, 14, 48	14, 16, 21, 36, 44, 48
8	2,373,040	101.2655	1	W	6, 13, 18, 28, 43	5, 6, 11, 43, 48	7, 16, 21, 36, 48
9	2,372,633	119.0577	1		2, 6, 18, 28, 43	5, 6, 11, 14, 43, 48	7, 16, 21, 36, 44, 48
10	2,336,542	121.2323	1	W	2, 13, 18, 28, 43	5, 11, 13, 14, 43, 48	7, 16, 21, 36, 44, 48
11	2,327,001	121.3514	1	W	2, 6, 23, 28, 43	6, 11, 14, 23, 34, 43, 50	16, 21, 22, 34, 36, 47, 50
12	2,318,702	122.6209	1	W	2, 6, 13, 28, 43	5, 6, 11, 13, 14, 43, 48	7, 16, 21, 36, 44, 48
13	2,317,408	123.2003	3	W	2, 6, 18, 23, 28, 43	6, 11, 14, 23, 34, 43, 50	16, 17, 21, 34, 36, 42, 47
14	2,317,016	126.3608	3	W	2, 6, 18, 23, 28, 43	6, 11, 14, 23, 34, 43, 50	16, 21, 22, 34, 36, 47, 50
15	2,317,008	126.5215	3	Ts, W	2, 6, 18, 23, 28, 43	6, 11, 14, 23, 34, 43, 50	16, 21, 34, 36, 42, 47, 50
16	2,316,923	128.2639	3	W	2, 6, 18, 23, 28, 43	6, 11, 14, 23, 34, 43, 50	16, 21, 22, 34, 36, 42, 50
17	2,316,392	129.7142	2	W	2, 6, 18, 23, 28, 43	6, 11, 14, 23, 34, 43, 50	16, 21, 22, 34, 36, 42, 47, 50
18	2,312,040	130.0187	3	W	2, 6, 18, 23, 28, 43	5, 6, 14, 23, 34, 43, 48	7, 14, 34, 42, 44, 47, 48
19	2,311,517	134.9016	1	W	2, 6, 18, 23, 28, 43	5, 6, 14, 23, 34, 43, 48	7, 16, 21, 34, 36, 42, 44, 47, 48
20	2,311,161	136.1711	1	W	2, 6, 13, 18, 23, 28, 43	5, 6, 11, 13, 14, 43, 48	7, 14, 16, 21, 36, 44, 48
21	2,308,774	136.8799	1		2, 6, 15, 18, 23, 28, 43	5, 6, 11, 14, 15, 43, 48	7, 14, 16, 21, 36, 44, 48
22	2,276,022	137.0762	1	W	2, 13, 18, 23, 28, 43	5, 11, 13, 14, 23, 43, 48	7, 14, 16, 21, 36, 44, 48
23	2,273,635	137.785	1	W	2, 15, 18, 23, 28, 43	5, 11, 14, 15, 23, 43, 48	7, 14, 16, 21, 36, 44, 48
24	2,268,411	138.2501	1	W	2, 13, 15, 18, 28, 43	5, 11, 13, 14, 15, 43, 48	7, 14, 16, 21, 36, 44, 48
25	2,268,166	138.4648	1	W	2, 6, 13, 23, 28, 43	6, 11, 13, 14, 23, 34, 43, 50	16, 21, 22, 34, 36, 42, 47, 50
26	2,267,415	139.0545	1	W	2, 13, 15, 18, 28, 43	5, 11, 13, 14, 15, 43, 48	5, 7, 16, 21, 36, 44, 48
27	2,263,109	139.1736	1	W	2, 6, 15, 23, 28, 43	6, 11, 14, 15, 23, 34, 43, 50	6, 11, 16, 21, 34, 36, 42, 44
28	2,255,576	140.4431	1	W	2, 6, 13, 15, 28, 43	2, 5, 6, 11, 43, 48	5, 6, 7, 11, 16, 21, 36, 44
29	2,252,026	143.2915	2	W	2, 6, 13, 18, 23, 28, 43	5, 6, 14, 23, 34, 43, 48	7, 16, 21, 34, 36, 42, 44, 47, 48
30	2,250,014	152.015	1		2, 6, 13, 18, 23, 28, 43	5, 6, 14, 23, 34, 43, 48, 50	7, 16, 21, 34, 36, 42, 44, 47, 48
31	2,246,204	152.7238	1	W	2, 6, 15, 18, 23, 28, 43, 48	5, 6, 14, 15, 23, 34, 43, 48	14, 16, 21, 34, 36, 42, 44, 47
32	2,238,791	153.9933	1	W	2, 6, 13, 15, 18, 28, 43	5, 6, 11, 13, 14, 15, 43, 48	6, 7, 11, 16, 21, 36, 44, 48
33	2,205,934	154.8984	1	W	2, 13, 15, 18, 23, 28, 43	2, 5, 14, 23, 34, 43, 48	5, 7, 16, 21, 34, 36, 42, 44, 47
34	2,194,305	156.287	1	W	2, 6, 13, 15, 23, 28, 43	2, 5, 6, 23, 34, 43, 48	5, 6, 7, 16, 21, 34, 36, 42, 44
35	2,174,106	163.5352	2	W	2, 6, 13, 15, 18, 23, 28, 43	2, 5, 6, 23, 34, 43, 48	5, 6, 7, 16, 21, 34, 36, 42, 44, 47
36	2,170,416	169.8372	1		2, 6, 13, 15, 18, 23, 28, 43	5, 6, 14, 15, 23, 34, 43, 48, 50	7, 14, 16, 21, 34, 36, 42, 44, 48

Notes: The 'W's and 'Ts's stand for those efficient points that were missed by the Weighted Objective Function and the Tchebycheff methods respectively.

Table 2. Differences in solution optimality between the various methods.

ε-Constraining & Triangles			Alpha-Delta		Difference	Tschebyseff		Difference
No.	NPV ($)	Habitat (ha)	NPV ($)	Habitat (ha)	in NPV (%)	NPV ($)	Habitat (ha)	in NPV (%)
2	2,427,424	64.3817	2,426,808	64.3817	0.0254	2,426,828	64.3817	0.0246
3	2,423,119	64.9906	2,423,028	64.9906	0.0038	2,422,636	64.9906	0.0199
4	2,410,088	66.3003	2,409,557	66.3003	0.0220	Efficient Point Missed		
5	2,409,663	66.7425	2,409,088	66.7425	0.0239	2,409,608	66.7425	0.0023
36	2,170,416	169.8372	2,170,320	169.8372	0.0044	2,170,416	169.8372	0.0000