SILVA FENNICA
Silva Fennica 40(2) research articles
www.metla.fi/silvafennica · ISSN 0037-5330
The Finnish Society of Forest Science · The Finnish Forest Research Institute

A New Heuristic Method for Solving Spatially Constrained Forest Planning Problems Based on Mitigation of Infeasibilities Radiating Outward from a Forced Choice

Pete Bettinger and Jianping Zhu

Bettinger, P. & Zhu, J. 2006. A new heuristic method for solving spatially constrained forest planning problems based on mitigation of infeasibilities radiating outward from a forced choice. Silva Fennica 40(2): 315–333.

A new heuristic method to mitigate infeasibilities when a choice is forced into a solution was developed to solve spatially constrained forest planning problems. One unique aspect of the heuristic is the introduction of unchosen decision choices into a solution regardless of the resulting infeasibilities, which are then mitigated by selecting next-best choices for those spatial units that are affected, but in a radiating manner away from the initial choice. As subsequent changes are made to correct the affected spatial units, more infeasibilities may occur, and these are corrected as well in an outward manner from the initial choice. A single iteration of the model may involve a number of changes to the status of the decision variables, making this an *n*-opt heuristic process. The second unique aspect of the search process is the periodic reversion of the search to a saved (in computer memory) best solution. Tests have shown that the reversion is needed to ensure better solutions are located. This new heuristic produced solutions to spatial problems that are of equal or comparable in quality to traditional integer programming solutions, and solutions that are better than those produced by two other basic heuristics. Three small hypothetical forest examples illustrate the performance of the heuristic against standard versions of threshold accepting and tabu search. In each of the three examples, the variation in solutions generated from random starting points is smaller with the new heuristic, and the difference in solution values between the new heuristic and the other two heuristics is significant ($p < 0.05$) when using an analysis of variance. However, what remains to be seen is whether the new method can be applied successfully to the broader range of operations research problems in forestry and other fields.

Keywords integer decision variables, integer programming, forest management
Authors' address Warnell School of Forest Resources, University of Georgia, Athens, GA 30602, USA **E-mail** pbettinger@forestry.uga.edu
Received 9 January 2006 **Revised** 14 March 2006 **Accepted** 20 March 2006
Available at http://www.metla.fi/silvafennica/full/sf40/sf402315.pdf

1 Introduction

Spatial forest planning problems, such as those that involve the location and timing of harvest units, generally require integer decision variables, and can be difficult planning problems to solve. Recently the interest in using or developing spatial forest planning techniques has increased (Bettinger and Chung 2004), motivated by changes to voluntary and regulatory forest-based programs, and facilitated by advancements in computer technology (Bettinger and Sessions 2003). Within the field of forest planning, green-up or adjacency constraints are among the most widely addressed spatial constraints. A common spatial forest planning problem would be to schedule forest harvest activities over time and space so that the openings remain below a certain size (i.e., the adjacency issue), and so that the openings are not near one another within a specified time frame (i.e., the green-up issue).

A significant amount of work has been published in the past 15 years regarding the use of spatial harvest scheduling constraints, from the early work of Torres-Rojo and Brodie (1990), Weintraub et al. (1994), and Yoshimoto and Brodie (1994) to the advancements provided by Snyder and ReVelle (1997), Murray (1999), and McDill and Braze (2000). The list of associated work is too long to include a summary of all of the relevant here, but advancements have proceeded generally along two lines: the development of adjacency constraint formulations for exact algorithms, and the development of heuristics. Spatial forest planning has spanned the fields of forest transportation (e.g., Weintraub et al. 1995), wildlife management (e.g., Bettinger et al. 1997), aquatic system management (Bettinger et al. 1998), biological diversity (Kangas and Pukkala 1996), and clearcut size distributions (Murray et al. 2004).

A number of mathematical programming techniques have been shown to be useful for spatial forest planning problems, including traditional mathematical programming (Weintraub and Navon 1976), dynamic programming (Hoganson and Borges 1998), Monte Carlo integer programming (O'Hara et al. 1989, Nelson and Brodie 1990, Clements et al. 1990), simulated annealing (Lockwood and Moore 1992, Dahlin and Sallnas 1993, Van Deusen 1999), tabu search (Bettinger et al. 2002, Boston and Bettinger 1999, Murray and Church 1995), threshold accepting (Bettinger et al. 2003), and genetic algorithms (Mullen and Butler 1999). Other heuristics, such as the sequential approach presented in Pukkala and Kangas (1993), the sequential quenching and tempering approach of Falcão and Borges (2002), and hybrid heuristics (Boston and Bettinger 2002, Borges et al. 1999) have also shown promise for forest planning problems. Most of these, with the exception of mathematical programming and dynamic programming, illustrate the development and usefulness of heuristics in forest planning. Much of the work related to the development of adjacency constraint formulations focuses on traditional mathematical programming as the solu-

Table 1. A brief categorization of search processes used in forest planning.

Search technique	Type of change to a developing forest plan	Number of units whose status is changed
Traditional exact methods (LP, IP)	Deterministic	Many
Dynamic programming	Deterministic	Many
Monte Carlo simulation	Random	Many
Simulated annealing	Random	One
Threshold accepting	Random	One
Tabu search	Deterministic	One
Genetic algorithms	Random	Many
Sequential search [a)]	Deterministic	One
Sequential quenching and tempering [b)]	Both	Many

[a)] Pukkala and Kangas (1993)
[b)] Falcão and Borges (2002)

tion technique (e.g., McDill and Braze 2000).

In general, when applied to forest planning problems, these search processes can be categorized by the type of change that is made to a developing forest plan with each "iteration" of the process (Table 1), and the number of units (stands, polygons) whose status (prescription, harvest timing) changes. Although these generalizations may be simplistic, they are characteristic of the basic processes when implemented in a forest planning environment. Bettinger (2005) provides an concise overview of the operation of the basic heuristic processes. Of course, a number of enhancements have been demonstrated as useful improvements to the basic search processes, such as the use of 2-opt moves (Bettinger et al. 1999, Heinonen and Pukkala 2004) or strategic oscillation (Richards and Gunn 2000).

In almost every example of heuristics applied to forest planning problems, potential infeasibilities associated with changes to a developing forest plan are either avoided (keeping the search process inside the feasible region of the solution space), recognized with penalties applied to the objective function, or mitigated (removing the infeasibilities by selecting alternative choices for the affected units). The mitigation processes have usually consisted of unscheduling or rescheduling (maintaining feasibility) a unit (stand) from a harvest schedule (e.g., Boston and Bettinger 2002). Many heuristics also require a number of parameters to need evaluation prior to implementation. For example, these might consist of the tabu state for tabu search, or the cooling rate for simulated annealing.

Here, we introduce a new heuristic that requires only two parameters, one that ignores potential infeasibilities when making changes to a developing forest plan (although correcting them later), and one that seems to produce very good solutions to difficult forest planning problems. The development of radiating waves in standing water by the impact of falling raindrops inspired this new method of addressing spatial forest planning problems. In short, when a choice is made for a developing forest plan, and infeasibilities occur, the mitigation process allows further infeasibilities to occur as the corrections are made in a radiating fashion away from the initial choice. This is a process that has yet to be explored in spatial harvest scheduling and forest planning and represents an improvement in solving some types of spatial forest planning problems.

2 Methods

The scheduling process we designed begins with the development of a randomly defined, feasible solution to the problem (Fig. 1). This is a random search process much like that used in Monte Carlo simulation. Next, as the forest plan is developing, a choice (management unit and harvest period) not currently in the solution is selected at random. The choice is then forced into the solution regardless of any potential constraint violations. A list of the management units that are affected (i.e., that then result in an infeasible solution) is then compiled. From this list, the management unit physically nearest the original selected management unit (based here on the centroids of each unit) is identified, and the next best alternative (harvest period) for this unit, that does not result in a constraint violation with the originally selected unit, is chosen and forced into the solution. Any management units that are subsequently affected by this change to the solution are added to the list of affected units. The list of affected units is then again consulted, and the unit physically nearest the originally selected unit is once again selected. The next best alternative for this third unit is chosen, as long as it does not result in a constraint violation with units previously examined, including the original unit and any other management units physically nearer the originally selected unit than the affected unit now being considered. This radiating, spatially sprawling (Fig. 2) adjustment process continues until all infeasibilities have been corrected. Thus ends one iteration of the search process. The process continues for a user-defined number of iterations, and at a user-defined interval, the current solution to the problem reverts back to the previously saved best solution.

The scheduling process is different than previous heuristic processes in a number of ways. First, while the selection of management units and harvest periods to enter into a solution is random, much the same as Monte Carlo simulation, threshold accepting, and simulated annealing, a single

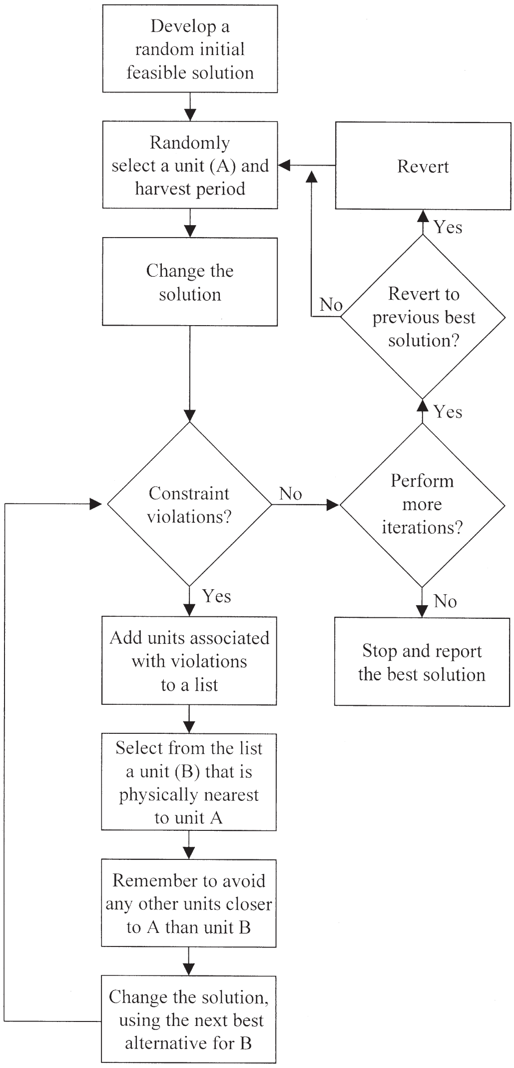
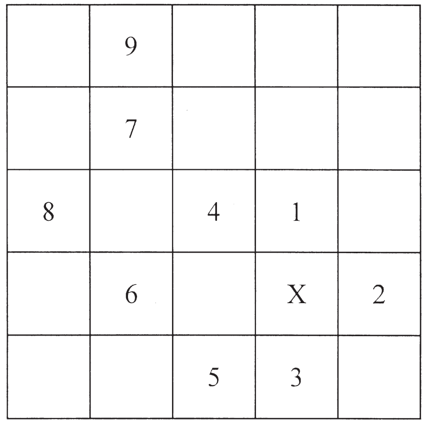


Fig. 1. A flow chart of the new heuristic search process for spatially constrained forest planning problems.



X = initial unit chosen to enter the solution
1–9 = subsequent affected units (in order, by distance from the original unit chosen) whose status is changed in the solution

Fig. 2. A conceptual model of the spatially sprawling infeasibilities that occur when a management unit and harvest period (X) is forced into a solution, and the resulting order of infeasibilities that must be corrected to complete one iteration of the new method.

iteration of this new process also includes subsequent deterministic changes to other affected management units. And unlike many of the basic implementations of other heuristics, any constraint (adjacency) violations that occur require an adjustment of the affected management unit's harvest timing such that the next best harvest timing is selected (regardless of the resulting additional infeasibilities). These changes are made in a geographical manner, and are applied first to the management units physically nearest to the originally selected unit. Also, this search process avoids the use of parameters typically employed in other heuristics. Thus this model lacks the moderately cryptic user-defined parameters such as the initial temperature (or threshold), the annealing rate (or threshold change value), the tabu state, or the mutation rate used in other heuristics, that cause users to evaluate their significance to the search process. The only two criteria that a user must define are the total number of iterations to model (as in all other heuristics that lack an intelligent stopping criteria), and the number of iterations that allowed to pass before the heuristic reverts to the previously saved best overall solution.

2.1 Forest Planning Problem Formulation

To illustrate the use of the new spatial forest planning process, a forest management problem is defined and solved using three small spatial databases that represent potential harvest volumes from three areas of the United States (U.S.). The objective of the forest planning problem is to maximize timber harvest over three time periods. The heuristic model was designed such that the objective function was represented as:

$$\text{minimize} \sum_{t=1}^{3}\left(H_t - T\right)^2 \qquad (1)$$

where
t = a time period
H_t = the total harvest volume (tons) during a time period
T = a target harvest volume

This is an even-flow (of harvest volume) objective, which seeks to locate a solution with the highest and most even harvest volume for each of the three time periods. Attempting to achieve perfect even-flow across a time horizon is relatively easy within a linear programming environment, where decision variables are continuous in nature. However, when using integer variables, the initial structure of a forest (polygon size distribution and forest age class distribution) may preclude an exact, and relatively high, harvest rate from being obtained.

To accumulate scheduled harvest volumes, accounting rows were used to sum the scheduled harvest volume during each time period:

$$\sum_{i=1}^{n}\left(X_{it}V_{it}A_i\right) - H_t = 0 \qquad \forall t \qquad (2)$$

where
i = a management unit
n = total number of management units
X_{it} = a binary variable indicating whether (1) or not (0) unit i is scheduled for harvest during time period t
V_{it} = the volume per unit area available for unit i during time period t
A_i = the size of a management unit

Unit restriction adjacency constraints (Murray 1999) are included in this problem formulation, where a unit sharing an edge (side) with another can not be harvested in the same time period.

$$X_{it} + X_{jt} \leq 1 \qquad \forall\, t, i, j \in \{N_i\} \qquad (3)$$

where
j = a management unit
N_i = the set of all management units adjacent to unit i

When addressing adjacency constraints in forest harvest scheduling problems, the unit restriction model is more restrictive on final solution values than the area restriction model (Murray 1999). While both models can be solved with exact methods, the area restriction model allows more flexibility in the harvest timing of individual stands, thus the final solution values are generally higher (in a maximization problem) than those provided by the unit restriction model. However, one aspect of this study is to compare the heuristic solution values to solution values provided by exact methods, and since the unit restriction model is more easily formulated with exact methods than the area restriction model, we chose to solve it and provide it as a way to validate the results. In addition, Murray (1999) notes that defining the potential contiguous areas for the area restriction model of adjacency and green-up may be difficult even for small forest planning problems.

Of the various types of adjacency constraint formulations (e.g., pairwise, Type I nondominated, new ordinary adjacency matrix), these are pairwise constraints. While Type I nondominated constraints have been shown to result in significantly lower solution times, and new ordinary adjacency matrix have also been shown to perform better in problems containing mainly immature forests, pairwise constraints have been shown to perform better in forest planning problems containing overmature and old-growth forests (McDill and Braze 2000). In addition, McDill and Braze (2000) state that the more mature the forest, the harder the problem to solve. This is important when using the branch and bound algorithm associated with the integer programming solver (LINDO), because a smaller proportion of branches are the trimmed from the branch and bound tree, thus the decision tree that is explored is larger than with other types of forest configurations (McDill and Braze 2001). Each of our example forest problems contain mature forests (i.e., those ready for harvest). In addition, no tolerance gap was used in the generation of integer programming solutions, although McDill and Braze (2001) show that near-optimal solutions can be obtained quickly by using a 0.1–2.0% gap.

In this management problem, the number of choices available to each management unit is the number of time periods plus 1 (one choice for each potential clearcut period, and one additional choice for not cutting a unit). The number of units that can be clearcut during any one time period is unlimited, as the objective function (maximization of even-flow) sorts this out during the development of a forest plan.

2.2 Hypothetical Forests to Which the Model Is Applied

To assess the effectiveness of the new heuristic for solving spatially constrained forest planning problems, three hypothetical forests are used. First, a 625-unit hypothetical southern U.S. forest was designed as a 25 by 25 unit grid, with each grid cell representing 10 ha. Forest ages (0 to 30 years) were randomly assigned to each cell. Forest volumes were assigned to each age based on slash pine (*Pinus elliotii* Engelm.) volumes provided by Bailey et al. (1982) for stands with uniform stocking and 15% fusiform rust (*Cronartium fusiforme* Hedgcock & Hunt ex Cummins) stem cankers. The time horizon is assumed to be 15 years in this example, thus each time period is 5 years long.

Second, a 74-unit, 1012 ha forest commonly referred to as the Daniel Pickett Forest (Davis et al. 2001) is used to provide realistic polygons (Fig. 3). Potential board foot volumes for typical western U.S. forests are assigned to these polygons. The average size of polygons in this forest average 13.6 ha. The time horizon assumed in this example is 30 years, thus each time period represents one decade (10 years).

A 40-unit, 631 ha forest with parcels of 9, 18, and 36 ha was developed as the final standard problem to assess the usefulness of the new heuristic (Fig. 4). Potential northern U.S. hardwood yields for white oak (*Quercus alba* L.), red oak (*Quercus rubra* L.), and other red oaks (e.g., *Quercus palustris* Muenchh.) were assigned to each stand randomly. Yields were developed from estimated volumes for oak-hickory forest types in

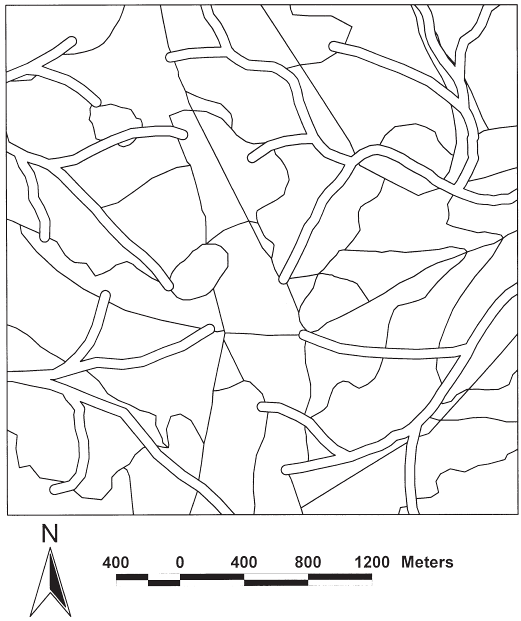


Fig. 3. The spatial arrangement of polygons that represent the 1012 ha Daniel Pickett forest (western U.S.) example.

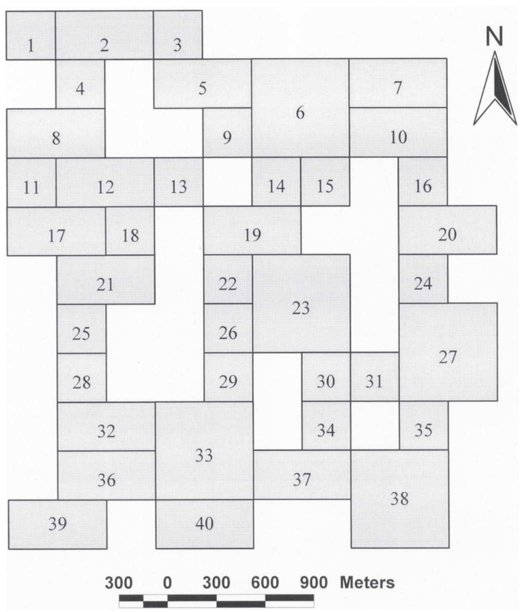


Fig. 4. The spatial arrangement of polygons that represent the 40-unit, 635 ha northern forest example.

Wisconsin (Essex and Hahn 1976). The time horizon assumed in this example is also 30 years, and thus each time period represents one decade.

Data regarding each of the hypothetical forest databases is available via the Internet (Bettinger 2004). Databases are both tabular (area, potential volumes, adjacency list, centroids of stands) and spatial (ArcView GIS databases for the 73-unit Daniel Pickett forest and the 40-unit northern U.S. forest example).

2.3 Validation

There are four general methods for assessing and validating the performance of a heuristic search process:

1) Compare solution values against a relaxed linear programming (LP) solution
2) Compare solution values against an integer programming (IP) solution (preferred)
3) Compare solution values against an estimated global optimum developed using extreme value theory
4) Compare solution values against those produced by other heuristics

While it may make sense to compare the new heuristic against others (which we have done), the best method for comparing the performance of a heuristic is against a very similar, and known, global optimum solution from integer programming (which we have also done). However, the objective function to the problem described above (Eq. 1) contains a non-linear term which can not be accommodated in the integer programming problem formulation. This non-linear term is needed in the heuristic to differentiate between similar solutions, one of which may have a more uneven volume stream. For example, assume two solutions have the following set of harvest volumes over three time periods:

Set A {1000, 1000, 1020}
Set B {950, 1000, 1070}

Each of the two solutions has the same total harvest volume over the time horizon (3020 units), although set A has harvest volumes that are obviously more even. If we were to assume a target volume of 1100 units per time period, each of the two sets is also 280 units away, in total, from the target. To recognize that set A is more even than set B, we squared the periodic deviations from the target volume in Eq. 1. Thus set A is given a value of 26 400, while set B receives a value of 33 400. Set A is thus preferred if one were to attempt to minimize the squared deviations from the target harvest level. Thus the non-linear term was needed to help us make this decision.

While the objective function (Eq. 1) for the heuristic contained a non-linear term, the integer programming problem was formulated so that the objective function simply maximized total harvest volume over the three time periods:

$$\text{minimize} \sum_{t=1}^{3} Z_t \tag{4}$$

where

$$Z_t = T - H_t \tag{5}$$

To ensure that the harvest levels were as high as possible, and as close as possible, pseudo even-flow constraints were included:

$$H_1 - H_2 \geq -Z \tag{7}$$
$$H_2 - H_1 \geq -Z \tag{8}$$
$$H_1 - H_3 \geq -Z \tag{9}$$
$$H_3 - H_1 \geq -Z \tag{10}$$
$$H_2 - H_3 \geq -Z \tag{11}$$
$$H_3 - H_2 \geq -Z \tag{12}$$

The value of Z allows some flexibility in the harvest levels, since an exact even-flow harvest level for each time period was impossible to find within a reasonable (12 hour) time period. In fact, as we alluded to earlier, an exact even-flow solution may be impossible to locate when using integer variables. These constraints were not used in the heuristic process, and making the Z values very small produces a different solution (generally more even in volume than the heuristic solution, but with lower total volumes). Therefore, to make the comparisons of the integer solutions and the heuristic solutions fair, the Z value was set to the largest deviation amongst periodic harvest levels found in the best solution generated by the new heuristic for each example forest problem. For the 625-unit southern U.S. example problem, we used a Z value of 66 300 tons. For the 73-unit Daniel

Pickett (western U.S. example) problem we used a Z value of 550 MBF. For the 40-unit northern U.S. forest example, we used a Z value of 133 m^3. The resulting integer programming models were solved using Industrial LINDO/PC release 6.1 (LINDO Systems, Inc. 2002).

In addition to comparing solutions generated by the new heuristic to the integer programming solutions, we also compare the new heuristic's solutions to solutions generated by standard threshold accepting and tabu search methods. Threshold accepting was initially developed by Dueck and Scheuer (1990), and has been applied to forest planning problems by Bettinger et al. (2002, 2003). Rather than a linear change to the threshold, which is commonly used, a geometric change was used (0.995 × previous threshold). Through numerous tests, we have found that this type of "threshold change" moves the search process to better areas of the solution space more quickly, and results in higher quality solutions. Tabu search was initially described by Glover (1989), and has been applied to numerous forest planning problems (e.g., Batten et al. 2005, Bettinger et al. 1997, 2002). Standard 1-opt tabu search was used for the comparison with the new method, although others have shown that intensification (Bettinger et al. 1999) or diversification (Richards and Gunn 2000) strategies may be necessary. Each of these three heuristics (the new method, threshold accepting, tabu search) was developed within the HATT environment (Bettinger 2005), although only tabu search and threshold accepting are currently available to the public through an Internet site (Bettinger 2004).

Fifty solutions were generated using the three heuristics to compare the quality of their results. The target volumes that were used in this assessment were derived from relaxed linear programming solutions to each forest planning problem. They were: a) 2 972 462 tons for the 625-unit southern U.S. forest example, b) 9134.6 m^3 for the 40-unit northern U.S. forest example, and c) 34 467 MBF (thousand board feet) for the 73-unit western U.S. forest example.

Parameterization is required of each of the three heuristics examined. We made a concerted effort to select parameters for each model that would allow an objective comparison of each. We ran numerous trial runs of tabu search, and found that the tabu state should be about 150 iterations for the 73-unit western U.S. example, 600 iterations for the 625-unit southern U.S. example, and 80 iterations for the 40-unit northern U.S. example. We ran the tabu search heuristic for 100 000 iterations on the 73-unit western U.S. forest example and the 625-unit southern U.S. forest example. The tabu search heuristic was allowed to run for

Table 2. Preliminary results from the new heuristic method using alternative reversion rates (20 solutions generated using each reversion rate).

Reversion rate	Objective function values [a)]: Minimum (best)	Maximum (worst)	Average	Standard deviation
0	454 340 553 112	572 570 439 512	518 646 161 762	36 977 836 837
1	78 132 994 612	103 806 340 812	90 916 021 675	6 667 014 976
2	77 821 915 052	95 078 429 472	85 285 369 752	4 994 950 346
3	75 126 570 552	94 409 573 572	84 634 418 204	5 512 224 349
4	70 295 758 932	91 337 174 632	81 576 452 568	5 667 170 231
5	76 409 475 212	95 462 209 292	85 035 377 346	5 609 720 952
6	77 640 865 772	98 719 285 752	87 459 908 985	5 903 718 797
7	72 868 783 992	97 472 457 032	87 774 263 412	6 615 753 840
8	78 279 990 572	106 577 261 092	91 052 723 877	7 915 577 284
9	75 954 907 852	108 422 576 972	90 592 355 860	8 965 679 652
10	83 169 836 552	108 708 774 252	94 537 143 988	6 940 360 532
20	93 086 521 952	119 977 882 872	107 796 307 146	6 930 477 957

a) Using Eq. 1, and applied to the 625-unit southern U.S. forest example.

10 000 iterations on the 40-unit northern U.S. forest example.

The threshold accepting heuristic required an examination of the initial threshold, the number of iterations per threshold, the rate at which the threshold changes, and the number of unsuccessful (either infeasible or outside the threshold) choices selected before the threshold changed. Numerous trials were run for each example forest. For the 73-unit western U.S. example, we decided to use an initial threshold of 10 000 000, and a rate of change of 0.995. The number of iterations per threshold was 1000, and the number of unsuccessful iterations prior to changing a threshold was 1000. For the 625-unit southern U.S. example, we decided to use an initial threshold of 110 000 000 000, and a rate of change of 0.995. The number of iterations per threshold was 10 000, and the number of unsuccessful iterations prior to changing a threshold was 1000. For the 40-unit northern U.S. example, we decided to use an initial threshold of 10 000 000, and a rate of change of 0.995. The number of iterations per threshold was 1000, and the number of unsuccessful iterations prior to changing a threshold was 500.

The new heuristic process was first subjected to preliminary analyses where the reversion rate varied from 0 iterations (no reversion to the best solution) to 20 iterations (Table 2). The heuristic was applied to the 625-unit southern U.S. forest example and allowed to run for 20 000 iterations. Twenty solutions were generated using each reversion rate tested. From this examination we found that reverting back to the best solution every four iterations of the model seemed to provide the best results. Here, not only was the best solution located, but the standard deviation was the amongst the lowest, and the average and worst solution was better than those provided when using the other reversion rates. What this shows is that the spatially-sprawling method, when used by itself (0 iteration reversion), does not produce good solutions to planning problems. The heuristic is essentially allowed to make numerous random and deterministic changes to a developing forest plan, yet there are no provisions to force the heuristic to find the better areas of the solution space. The reversion helps, but when too few and too many iterations pass before reverting back to the best solution, inferior solutions are developed. The best situation seems to be to let the new method perform about 4 randomly located, spatially-sprawling modifications to a solution. If a better solution is not found along the way, it is best to revert back to the best solution and try again. We subsequently used 4 iterations as the reversion rate for the remaining analysis of the new method, and ran the heuristic for 100 000 iterations on the 73-unit western U.S. forest example and the 625-unit southern U.S. forest example. The heuristic was allowed to run for 10 000 iterations on the 40-unit northern U.S. forest example.

3 Results

The integer programming version of the model produced optimal solutions to the 40-unit and 73-unit examples relatively quickly (<30 seconds). The 625-unit southern U.S. forest example was allowed to run in LINDO for about 20 hours. In none of the examples were the new method's solutions as good as the corresponding integer solution (Table 3). However, the best solution for the 40-unit problem produced with the new heuristic contained volumes that were within 1.1% per decade of the integer programming solution. The best solution for the 73-unit problem contained volumes that were within 0.7% per decade of the integer programming solution. The best solution for the 625-unit problem contained volumes that were within 0.4% per period of the integer programming solution. What should be kept in mind is that the new heuristic method was only run for a limited number of iterations to provide an objective comparison of the three heuristic methods. As an additional test of the robustness of the new heuristic, it was run a limited number of times (15) on the 40-unit and 73-unit example forest problems with 1 million iterations. For the 40-unit problem, 3 of the 15 solutions were slightly better than the solution noted in Table 3, but not as good as the integer programming solution. For the 73-unit problem, 5 of the 15 solutions were as good as the integer solution.

When examining the 50 solutions generated by each heuristic for the 73-unit western U.S.

Table 3. Comparison of integer programming solution and heuristic solutions.

	Objective function value [a)]	Harvest volumes (tons)		
		Period 1	Period 2	Period 3
625-unit southern U.S. forest example				
Integer solution	64 859 941 092	2 796 070	2 834 340	2 851 350
Best heuristic solution	68 823 857 652	2 789 800	2 824 390	2 856 130
73-unit western U.S. forest example				
Integer solution	5 500 391	33 049.5	32 933.6	33 399.4
Best heuristic solution	5 556 343	32 827.8	33 169.8	33 377.7
40-unit northern U.S. forest example				
Integer solution	98 439	8 981.5	8 903.6	8 987.5
Best heuristic solution	102 653	8 879.3	8 984.2	9 012.6

[a)] Using Eq. 1

Table 4. A comparison of 100 heuristic solutions developed for the 73-unit western U.S. forest example, using the new method, threshold accepting, and tabu search.

	New method	Threshold accepting	Tabu search
Minimum (best) [a)]	5 556 343	8 880 500	9 323 179
Maximum (worst) [a)]	15 491 044	39 050 808	33 907 707
Average [a)]	9 019 837	22 657 588	18 997 219
Standard deviation [a)]	2 432 331	6 481 078	5 439 394
Best solution volumes (MBF) [b)]			
Time period 1	32 828	32 433	32 417
Time period 2	33 170	32 779	32 663
Time period 3	33 378	33 090	33 102

[a)] Using Eq. 1
[b)] Target was 34,467 MBF per time period

forest example, we found that the new method produced better solutions than either threshold accepting or standard 1-opt tabu search (Table 4). The average solution generated using the new method was almost as good as the best solution generated from threshold accepting, and was better than the best solution generated with 1-opt tabu search. The spatial distribution of harvests (Figs. 5–7) allows one to visualize the resulting forest plans, and for this example problem show many similarities. Finally, the standard deviation of the solutions generated from the new method was relatively small, while the standard deviation of the solutions generated by the other two was large, as they resemble flat, normal distributions (Fig. 8). An analysis of variance test showed that the distribution of solutions produced by the new heuristic was significantly different ($p < 0.05$) than the distribution of solutions produced by the other heuristics.

From the 50 solutions generated by each heuristic for the 625-unit southern U.S. forest example, we found that the new method produced better solutions than either threshold accepting or standard 1-opt tabu search (Table 5). The worst solution from the new method was better than the

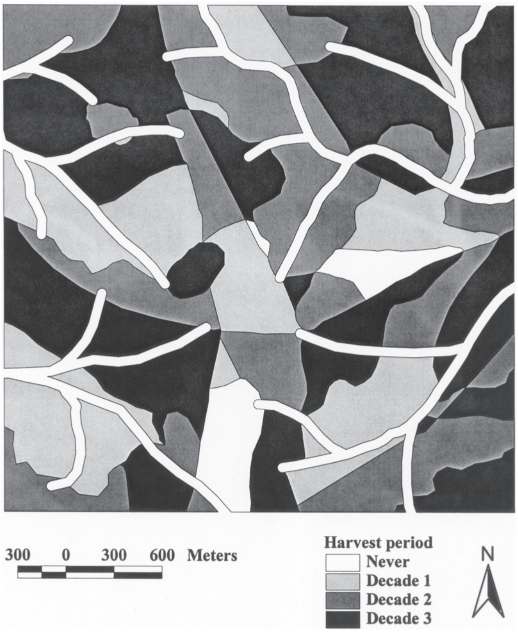


Fig. 5. The best solution generated by the new method for the 73-unit Daniel Pickett forest (western U.S.) example.

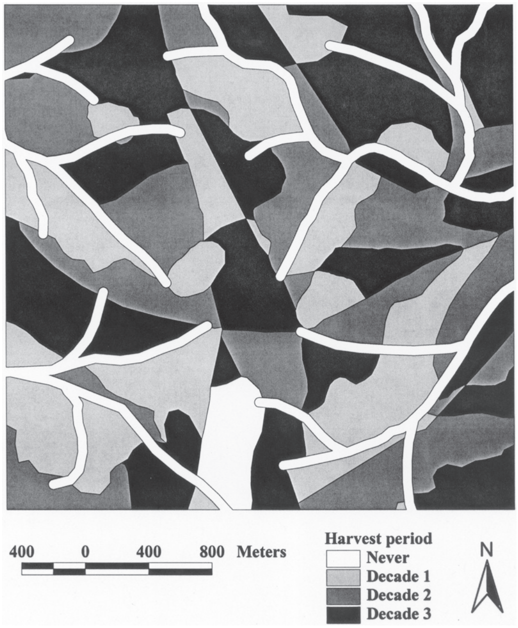


Fig. 6. The best threshold accepting solution for the 73-unit Daniel Pickett forest (western U.S.) example.

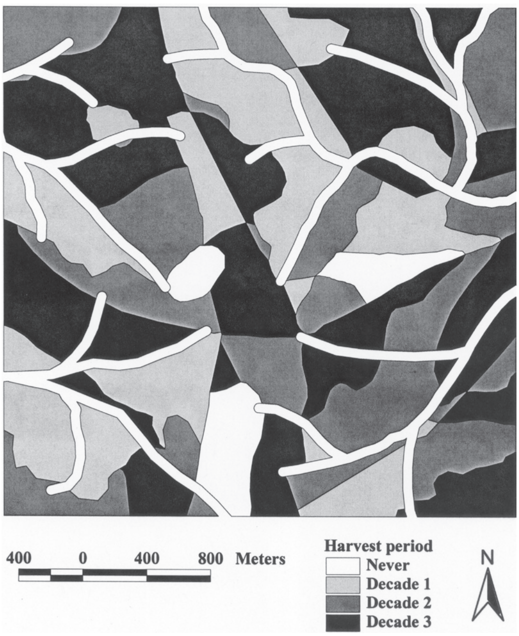


Fig. 7. The best tabu search solution for the 73-unit Daniel Pickett forest (western U.S.) example.

best solution developed from 1-opt tabu search. The average solution generated using the new method was almost as good as the best solution generated from threshold accepting. Finally, the standard deviation of the solutions generated from both the new method and threshold accepting were relatively small (Fig. 9), but an analysis of variance test indicated that the difference between the two distributions of solutions was significantly different ($p<0.05$).

Finally, when examining the 50 solutions generated by each heuristic for the 40-unit northern U.S. forest example, we once again found that the new method produced better solutions than either threshold accepting or standard 1-opt tabu search (Table 6). The new method and threshold accepting produced similar best solutions, while the average solution generated by the new method was almost as good as the best solution generated from 1-opt tabu search. The spatial distribution of harvests (Figs. 10–12) again allows one to visualize the resulting forest plans, and as with the 73-unit western U.S. forest example, show

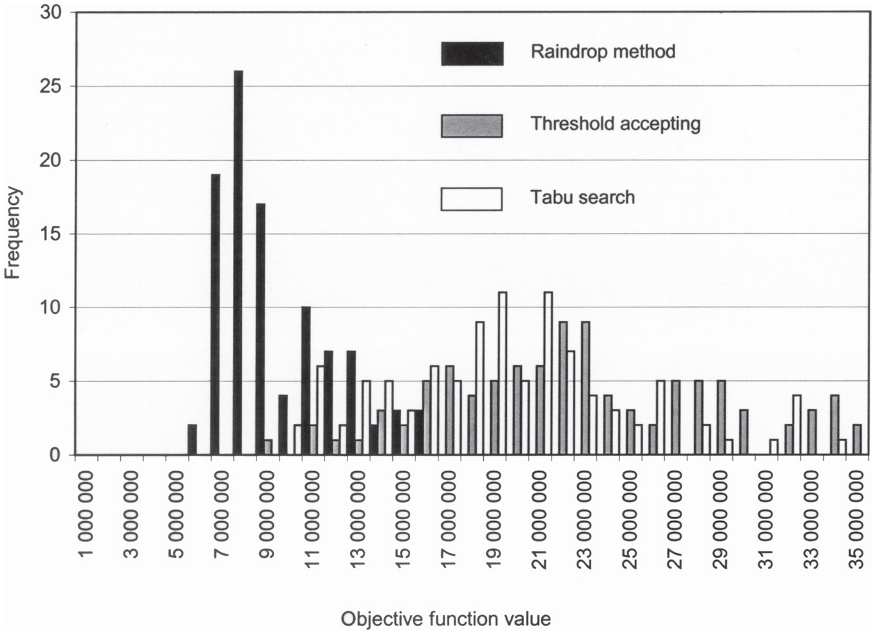


Fig. 8. The distribution of solutions generated for the new method, tabu search, and threshold accepting, using the 73-unit Daniel Pickett forest (western U.S.) example.

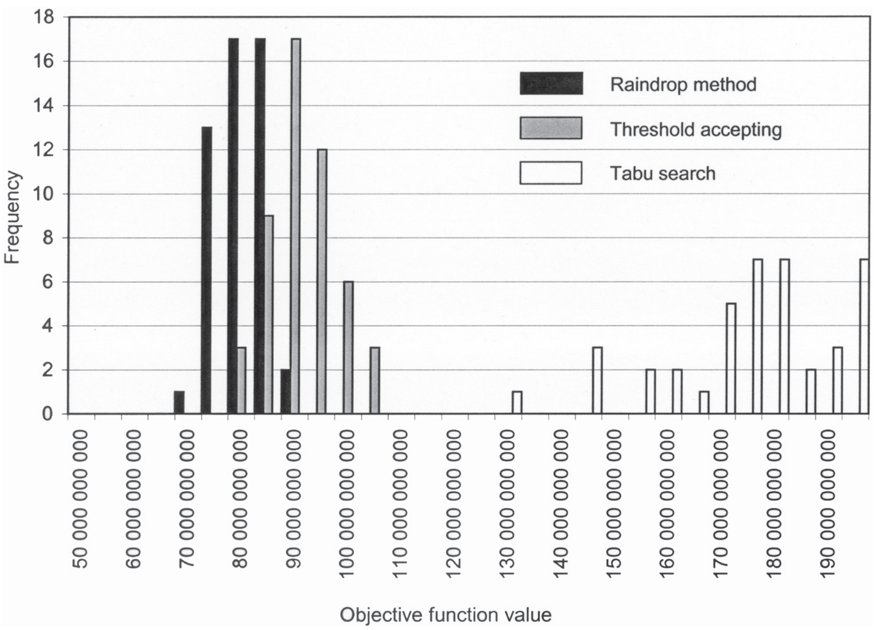


Fig. 9. The distribution of solutions generated for the new method, tabu search, and threshold accepting, using the 625-unit southern U.S. forest example.

Table 5. A comparison of 50 heuristic solutions developed for the 625-unit southern U.S. forest example, using the new method, threshold accepting, and tabu search.

	New method	Threshold accepting	Tabu search
Minimum (best) [a]	68 823 857 652	76 758 360 212	127 256 326 072
Maximum (worst) [a]	88 044 884 212	104 664 970 452	231 164 522 272
Average [a]	78 003 063 077	89 038 938 886	179 908 440 258
Standard deviation [a]	4 425 098 183	6 455 070 410	21 078 698 124
Best solution volumes (tons) [b]			
Time period 1	2 789 800	2 769 750	2 719 280
Time period 2	2 824 390	2 822 300	2 772 290
Time period 3	2 856 130	2 857 930	2 775 350

[a] Using Eq. 1
[b] Target was 2,972,462 tons per time period

Table 6. A comparison of 100 heuristic solutions developed for the 40-unit northern U.S. forest example, using the new method, threshold accepting, and tabu search.

	New method	Threshold accepting	Tabu search
Minimum (best) [a]	102 653	121 836	203 589
Maximum (worst) [a]	440 767	652 462	2 042 786
Average [a]	217 470	330 930	642 424
Standard deviation [a]	73 660	119 546	314 746
Best solution volumes (m^3) [b]			
Time period 1	8 879	8 858	8 878
Time period 2	8 984	9 007	8 875
Time period 3	9 013	8 965	8 869

[a] Using Eq. 1
[b] Target was 9,134.6 m^3 per time period

many similarities. Finally, the standard deviation of solution values generated from the new method and threshold accepting were relatively small, while the standard deviation of solution values generated by the 1-opt tabu search process was quite large (Fig. 13). However, as with the other two examples, an analysis of variance test showed that the distribution of solutions produced by the new heuristic was significantly different ($p<0.05$) than the distribution of solutions produced by the other heuristics.

All of the results provided here were developed using a personal computer equipped with a 2.4 MHz Pentium processor. The average time to generate a solution using the new heuristic was about 8 minutes for the 73-unit western U.S. forest example, about 21 minutes for the 625-unit southern U.S. forest example, and about 1 minute for the 40-unit northern U.S. forest example. By comparison, these values were slower than those for threshold accepting (20 seconds, 25 seconds, 8 seconds, respectively), and those for tabu search (7 minutes, 13.5 minutes, 40 seconds, respectively). One should keep in mind that the heuristic was developed using the Visual Basic 6.0 programming language, and that using other programming languages will likely shorten the time required to generate a solution.

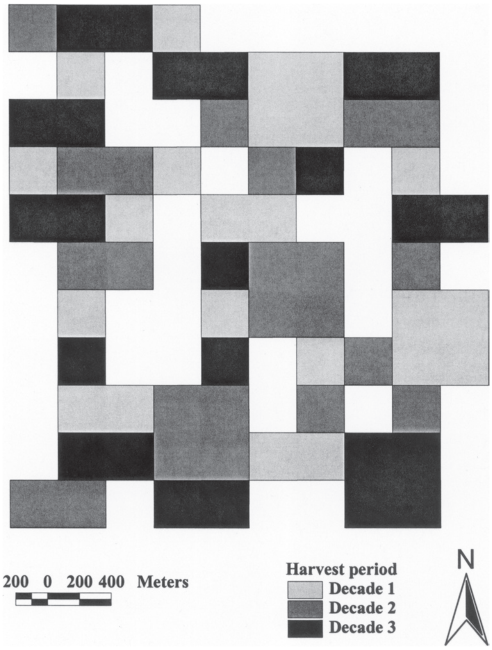


Fig. 10. The best solution generated by the new method for the 40-unit northern U.S. forest example.

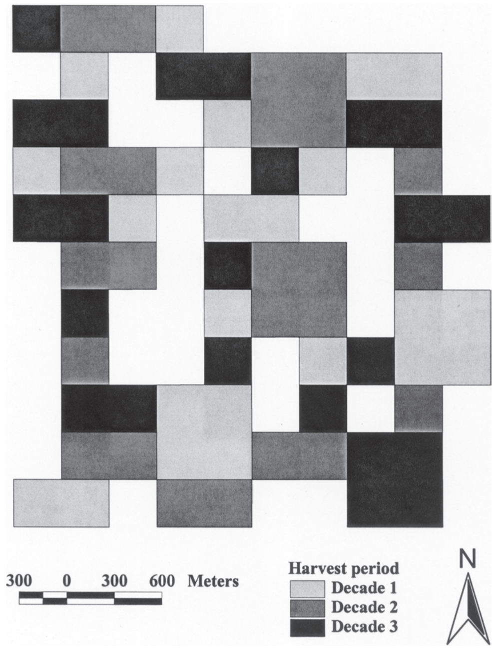


Fig. 11. The best threshold accepting solution for the 40-unit northern U.S. forest example.

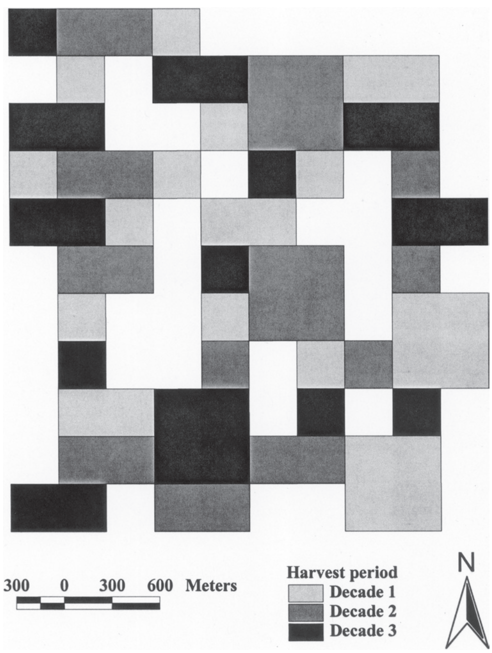


Fig. 12. The best tabu search solution for the 40-unit northern U.S. forest example.

4 Discussion

The process presented here, to the best of our knowledge, has not been presented previously in the literature. The method of randomly selecting a decision choice is not new, it is the main decision process in simulated annealing, threshold accepting, Monte Carlo simulation, genetic algorithms, and other heuristics. How the resulting infeasibilities are mitigated is new, however. Here, we compile a list of the affected management units, and based on their proximity to the original randomly selected management unit, the next best alternative for the affected unit(s) is chosen. This mitigation process could result in further infeasibilities, which are added to the list of affected management units, and the process continues until all infeasibilities have been eliminated. Other work, as noted earlier in this paper, either avoids infeasible solutions, mitigates the immediate infeasibilities (not resulting in new infeasibilities), or penalizes the infeasibilities (hoping that they would be removed later in the

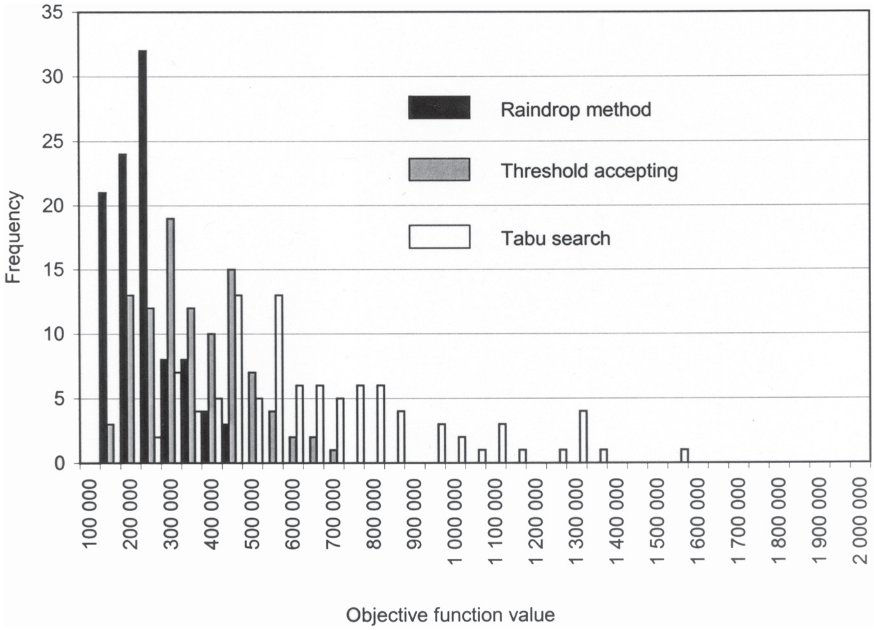


Fig. 13. The distribution of solutions generated for the new method, tabu search, and threshold accepting, using the 40-unit northern U.S. forest example.

search). Based on our results, this new method shows promise for further examination in forest planning problems.

Our new heuristic uses both random and deterministic changes to a developing forest plan, which is seldom found in a single iteration of a heuristic. In contrast to the sequential method presented in Falcão and Borges (2002), which also uses both random and deterministic changes to a developing forest plan, our heuristic makes the random change first, then uses deterministic changes to mitigate the infeasibilities. In Falcão and Borges (2002), deterministic changes are made first, then a set of random perturbations are made later. And while many units are assigned a new prescription during a single iteration when using the random changes, the changes are random, and all infeasibilities are penalized, not mitigated. Whereas in our new heuristic, during an iteration only the first choice is random, the others are deterministic, and feasibility is maintained at the end of the iteration.

The major advantage (i.e., the main benefit) of the new heuristic is that the number of parameters required by the user is limited to two: the total number of iterations to run the model, and the reversion rate. As Falcão and Borges (2002) have shown, and as we have shown here with threshold accepting and tabu search, some heuristics require significant parameterization prior to being used. Further, a second benefit is that the parameters used in the new heuristic are more intuitive than the parameters required to utilize simulated annealing (initial temperature, annealing rate), threshold accepting (initial threshold, threshold change rate), tabu search (tabu state), and genetic algorithms (mutation rate). This addresses the notion presented in Pukkala and Kangas (1993) that a system should be easy to use and understand. While some may argue that each of the other heuristics can be modified from their basic form to revert to the best solution, it is not a standard practice. In addition, assessing the impact of changing the status of decision choices, and correcting the resulting infeasibilities, is not a standard practice in other heuristics, where choices that lead to infeasibilities are either avoided, or the objective function is penalized.

However, by approaching the development of solutions in the manner we have suggested, the search process is allowed to move more forcefully through the solution space, as compared to typical 1-opt moves represented in simulated annealing, threshold accepting, and tabu search. The third advantage (and benefit) of the new heuristic is that the quality of solutions is better than other standard heuristics, and very close to integer programming results, implying high quality solutions are generated. A final advantage (the final main benefit) is that one can be assured that a set of solutions generated from random starting points have a tighter distribution than solutions generated similarly with the other standard heuristics studied, implying higher consistency in solution values.

The main disadvantage of the new heuristic is the additional programming logic required to identify and track the infeasibilities that result as a management unit and harvest period is forced into the solution, and as subsequent alterations to the solution are made to correct the infeasibilities that arise. This process of tracking infeasibilities is not usually incorporated into any other heuristic (e.g., simulated annealing, threshold accepting, or tabu search) unless for applying penalties to the objective function. In addition, the order of the mitigation of infeasibilities in the new heuristic is based the proximity of infeasible management unit choices to the original randomly selected management unit chosen at each iteration. This requires knowledge of the location of each management unit (i.e., the centroid), which is information that not commonly used in applications of other heuristics in forest planning.

Based on the results shown here, the utility of the new heuristic method for assisting with the development of spatial forest management plans show promise. Three additional areas of research associated with this search technique seem fruitful. First, we have only described the performance of the heuristic for one type of forest planning problem: maximization of even-flow of timber volume subject to unit restriction, one-period green-up, adjacency constraints. There are numerous other objectives that are common in forest planning, such as the maximization of net present value or some type of utility function. In addition, there are a multitude of constraints that could be associated with a forest planning problem, such as those that relate to wood flows of various products, cash flows, or habitat requirements. How well the process described here works with other forest planning problems is uncertain. One could speculate that as the number of constraints grows, the ability of the heuristic to mitigate the infeasibilities that arise may become cumbersome. Additional testing of the heuristic would be needed to fully assess this assumption, however.

Second, additional research may be needed to assess the selection of the initial management unit and harvest period (the choice forced into the solution) to reflect choosing a cluster of management units over a period of iterations, rather than a random selection of a management unit at each iteration. The clustering could be modeled spatially, irregardless of the underlying management units, or could be modeled with respect to those management unit and harvest choices that are more important to the planning problem. For example, perhaps the initial choices could be selected a cluster of older forested stands that are more likely to be harvested within the time frame of the resulting plan, rather than from a cluster of younger forested stands that may not contribute much to the objective. In addition, rather than selecting a single management unit and harvest period to begin each iteration of the model, multiple management units and harvest periods could be used to begin an iteration.

We reiterate that what we have described is a new heuristic that requires both the spatial-sprawling mitigation of infeasibilities and a periodic reversion to the best solution stored in memory. However, a third area of research could be aimed at incorporating the methods we have proposed here with other heuristics to form a hybrid search process. In addition, simply incorporating the reversion process into another type of heuristic may show promise for the generation of high quality forest plans. We leave these investigations, however, for the future and for other researchers.

5 Conclusions

We feel that the new heuristic we have presented here has significant potential for use in spatially constrained planning problems, whether arising from the field of forest management or from other disciplines. The radiating manner in which infeasibilities, based on their proximity to the a randomly selected management unit, are mitigated make this process unique among search processes currently used to address forest planning problems. The fact that the mitigation process could result in further infeasibilities, which must also be mitigated, until the solution is once again feasible, makes the search process unique. The ability to revert back to the best solution is a required parameter of the model, as no reversion seems to produce unacceptable final solutions. This, along with the number of iterations that the model is allowed to run, are the only considerations required by a user of the model, making it more simplistic than most other heuristics commonly used in forest planning.

The four main benefits of the new method, which suggest a contribution to science, are: 1) that higher quality solutions are generated, as compared to two other standard heuristics, 2) that a tighter distribution of solution values is generated, as evidenced by a relatively low standard deviation amongst sets of solutions, 3) that a smaller set of model parameters, as compared to other heuristics, which implies that the model needs less parameterization, and 4) that simpler and more intuitive parameters are used, as compared to other heuristics. What remains to be seen is whether the model can be applied successfully to the broader range of operations research problems. These include the travelling salesman problem and other types of machine scheduling problems, which also require integer decision variables, and forest planning problems with other objectives or a more comprehensive suite of constraints.

References

Bailey, R.L., Pienaar, L.V., Shiver, B.D. & Rheney, J.W. 1982. Stand structure and yield of site-prepared slash pine plantations. The University of Georgia College of Agriculture Experiment Stations, Research Bulletin 291. 83 p.

Batten, W.H., Jr., Bettinger, P. & Zhu, J. 2005. The effects of spatial harvest scheduling constraints on the value of a medium-sized forest holding in the southeastern United States. Southern Journal of Applied Forestry 29: 185–193.

Bettinger, P. 2004. Landscape planning databases. Available at: http://warnell.forestry.uga.edu/Warnell/Bettinger/planning/index.htm. [Cited 6 Jan 2006].

— 2005. HATT: The heuristic algorithm teaching tool for advanced forest planning courses. *In* Bevers, M. & Barrett, T.M. (comps.). Systems analysis in forest resources: Proceedings of the 2003 symposium. USDA Forest Service, Pacific Northwest Research Station, Portland, OR, General Technical Report PNW-GTR-656. p. 61–70.

— Chung, W. 2004. The key literature of, and trends in, forest-level management planning in North America, 1950–2001. The International Forestry Review 6: 40–50.

— & Sessions, J. 2003. Spatial forest planning: to adopt, or not to adopt? Journal of Forestry 101(2): 24–29.

— , Sessions, J. & Boston, K. 1997. Using Tabu search to schedule timber harvests subject to spatial wildlife goals for big game. Ecological Modelling 94: 111–123.

— , Sessions, J. & Johnson, K.N. 1998. Ensuring the compatibility of aquatic habitat and commodity production goals in eastern Oregon with a tabu search procedure. Forest Science 44: 96–112.

— , Boston, K. & Sessions, J. 1999. Intensifying a heuristic forest harvest scheduling search procedure with 2-opt decision choices. Canadian Journal of Forest Research 29: 1784–1792.

— , Graetz, D., Boston, K., Sessions, J. & Chung, W. 2002. Eight heuristic planning techniques applied to three increasingly difficult wildlife planning problems. Silva Fennica 36: 561–584.

— , Johnson, D.L. & Johnson, K.N. 2003. Spatial forest plan development with ecological and economic goals. Ecological Modelling 169: 215–236.

Borges, J.G., Hoganson, H.M. & Rose, D.W. 1999. Combining a decomposition strategy with dynamic programming to solve spatially constrained forest management scheduling problems. Forest Science 45: 201–212.

Boston, K. & Bettinger, P. 1999. An analysis of Monte Carlo integer programming, simulated annealing, and tabu search heuristics for solving spatial harvest scheduling problems. Forest Science 45: 292–301.

— Bettinger, P. 2002. Combining tabu search and genetic algorithm heuristic techniques to solve spatial harvest scheduling problems. Forest Science 48: 35–46.

Clements, S.E., Dallain, P.L. & Jamnick, M.S. 1990. An operational, spatially constrained harvest scheduling model. Canadian Journal of Forest Research 20: 1438–1447.

Dahlin, B. & Sallnäs, O. 1993. Harvest scheduling under adjacency constraints – a case study from the Swedish sub-alpine region. Scandinavian Journal of Forest Research 8: 281–290.

Davis, L.S., Johnson, K.N., Bettinger, P.S. & Howard, T.E. 2001. Forest Management. Fourth edition. McGraw-Hill, Inc., New York. 804 p.

Dueck, G. & Scheuer, T. 1990. Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. Journal of Computational Physics 90: 161–175.

Essex, B.L. & Hahn, J.T. 1976. Empirical yield tables for Wisconsin. USDA Forest Service, North Central Forest Experiment Station, St. Paul, MN. General Technical Report NC-25. 22 p.

Falcão, A.O. & Borges, J. 2002. Combining random and systematic search heuristic procedures for solving spatially constrained forest management scheduling models. Forest Science 48: 608–621.

Glover, F. 1989. Tabu search – Part I. ORSA Journal on Computing 1: 190–206.

Heinonen, T. & Pukkala, T. 2004. A comparison of one- and two-compartment neighborhoods in heuristic search with spatial forest management goals. Silva Fennica 38: 319–332.

Hoganson, H. & Borges, J. 1998. Using dynamic programming and overlapping subproblems to address adjacency in large harvest scheduling problems. Forest Science 44: 526–538.

Kangas, J. & Pukkala, T. 1996. Operationalization of biological diversity as a decision objective in tactical forest planning. Canadian Journal of Forest Research 26: 103–111.

LINDO Systems, Inc. 2002. Industrial LINDO/PC release 6.1. LINDO Systems, Inc., Chicago, IL.

Lockwood, C. & Moore, T. 1992. Harvest scheduling with spatial constraints: a simulated annealing approach. Canadian Journal of Forest Research 23: 468–478.

McDill, M.E. & Braze, J. 2000. Comparing adjacency constraint formulations for randomly generated forest planning problems with four age-class distributions. Forest Science 46: 423–436.

— & Braze, J. 2001. Using the branch and bound algorithm to solve forest planning problems with adjacency constraints. Forest Science 47: 403–418.

— , M.E., Rebain, S.A. & Braze, J. 2002. Harvest scheduling with area-based adjacency constraints. Forest Science 48: 631–642.

Mullen, D.S. & Butler, R.M. 1999. The design of genetic algorithm based spatially constrained timber harvest scheduling model. *In* Vasievich, J.M. et al. (eds.). Proceedings of the Seventh Symposium on Systems Analysis in Forest Resources. USDA Forest Service North Central Experiment Station, St. Paul, MN, General Technical Report NC-205. p. 57–65.

Murray, A.T. 1999. Spatial restrictions in harvest scheduling. Forest Science 45: 45–52.

— & Church, R.L. 1995. Heuristic solution approaches to operational forest planning problems. OR Spektrum 17: 193–203.

— , Goycoolea, M. & Weintraub, A. 2004. Incorporating average and maximum area restrictions in harvest scheduling models. Canadian Journal of Forest Research 34: 456–464.

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

O'Hara, A.J., Faaland, B.H. & Bare, B.B. 1989. Spatially constrained timber harvest scheduling. Canadian Journal of Forest Research 19: 715–724.

Pukkala, T. & Kangas, J. 1993. A heuristic optimization method for forest planning and decision making. Scandinavian Journal of Forest Research 8: 560–570.

Snyder, S. & ReVelle, C. 1997. Dynamic selection of harvests with adjacency restrictions: the SHARe model. Forest Science 43: 213–222.

Torres-Rojo, J.M. & Brodie, J.D. 1990. Adjacency constraints in harvest scheduling: an aggregation heuristic. Canadian Journal of Forest Research 20: 978–986.

Van Deusen, P.C. 1999. Multiple solution harvest scheduling. Silva Fennica 33(3): 1–9.

Weintraub, A. & Navon, D. 1976. A forest management planning model integrating silvicultural and transportation activities. Management Science 22: 1299–1309.

— , Barahona, F. & Epstein, R. 1994. A column generation algorithm for solving general forest planning problems with adjacency constraints. Forest Science 40: 142–161.

— , Jones, G., Meacham, M., Magendzo, A., Magendzo, A. & Malchuk, D. 1995. Heuristic procedures for solving mixed-integer harvest scheduling – transportation planning models. Canadian Journal of Forest Research 25: 1618–1626.

Yoshimoto, A. & Brodie, J.D. 1994. Comparative analysis of algorithms to generate adjacency constraints. Canadian Journal of Forest Research 24: 1277–1288.

Total of 43 references