Open Access

Informed Development of Meta Heuristics for Spatial Forest Planning Problems

Rongxia Li[1], Pete Bettinger[*,2] and Kevin Boston[3]

[1]*School of Forest Resources, Forest Bioproduct Research Initiative, University of Maine, Orono, ME 04469, USA*

[2]*Warnell School of Forestry and Natural Resources, University of Georgia, Athens, GA 30602, USA*

[3]*Department of Forest Engineering, Resources & Management, Oregon State University, Corvallis, OR 97331, USA*

Abstract: In this research application paper, the usefulness of an intelligent mechanism (a cubic spline smoothing technique) for determining when to switch from one algorithm to another within a meta heuristic search process is explored. We concentrated on a typical planning problem for a southern United States forestry company where the net present value of management activities is maximized subject to wood flow and harvest adjacency constraints. We found that more than 75% of the 3-algorithm meta heuristics examined produced consistently better solutions than the best standard heuristic (threshold accepting) in terms of mean and maximum solution values. However, a 2-algorithm meta heuristic (threshold accepting + tabu search) performed the best in terms of the average solution value and the absolute maximum solution value, improving solution quality 1.4% over the best standard heuristic solution value. Results also indicate meta heuristics which began a search with a relatively fast, stochastic search process (simulated annealing or threshold accepting) and end a search with a relatively slow, deterministic search process (e.g., tabu search) produced better solutions than other model configurations for the problem examined. Further, results suggest that the time to switch from one heuristic to another should be based on when the improvement in solution quality stagnates. Without recognizing this point, a search process may switch prematurely or be computationally wasteful.

Keywords: Operations research, forest planning, combinatorial optimization.

INTRODUCTION

Spatial forest planning has gained wide acceptance over the past decade, as people have gradually recognized the importance of tactical planning, and as forest sustainability concepts have been more closely integrated with planning processes [1]. Knowing the exact location of management activities can help forest managers better understand forest planning problems, account for spatial restrictions and wildlife habitat concerns, and thus allow them to make appropriate decisions. Many forest regulations and voluntary guidelines require or suggest that harvesting activities follow certain rules regarding clearcut sizes and landscape patterns [1]. Therefore, involving spatial components in forest planning processes helps planners more closely model operational issues. However, it is widely acknowledged that spatial forest planning problems can be difficult to solve [2], especially for those with green-up or adjacency constraints that control the timing and juxtaposition of harvests, since they are combinatorial in nature [1, 3-5]. Using exact mathematical methods (mixed integer programming and integer programming) to solve large spatial forest planning problems can be difficult, and can involve an excessively long computational times.

*Address correspondence to this author at the Warnell School of Forestry and Natural Resources, University of Georgia, Athens, GA 30602, USA; Tel: 706-542-1187; Fax: 706-542-8356;
E-mail: pbettinger@warnell.uga.edu

For these reasons, heuristic methods have been explored for addressing spatial forest planning problems, and they have been accepted as a practical approach to generate near-optimum solutions in a reasonable amount of time. The most commonly used heuristic methods in the field of forestry include simulated annealing [2, 4, 6-9], tabu search [10-12], genetic algorithms [12-14], threshold accepting [15, 16], and Monte Carlo random search [17]. Some of the heuristic methods have been enhanced to further explore the solution space and possibly improve the quality of solution values. For example, Richards and Gunn [18] designed an oscillating reactive tabu search and found it can improve solution values by about 20%. Bettinger *et al*. [5] developed 2-opt tabu search and also obtained better results over standard tabu search. Other forestry research efforts have shown that combining two algorithms may allow one to locate better solutions [12, 15]. However, this combination has generally been limited to two heuristic algorithms, and the decision criteria for switching processes has been relatively rote and determined by the experiences and expertise of the researcher (i.e., change after x number of iterations). While our assessment includes only the heuristic techniques commonly used in forest management and planning, applications and demonstrations of soft computing tools for decision support in fields other than forest management may inspire researchers to adapt the concepts to forestry problems. Advances in other areas of operations research include the use of neural network algorithms coupled with fuzzy reasoning [19] and fuzzy iteration methodologies [20] for addressing hydrologic concerns, ant colony optimization for risk management [21], and evolutionary techniques, which includes particle swarm optimization and others, for a variety of issues [22].

In this study, we examine the search behavior of four heuristic algorithms: simulated annealing (SA), threshold accepting (TA), tabu search (Tabu), and the raindrop method [23]. We then combine these heuristics into twelve 2-algorithm meta heuristics, and track search progress using a statistical measure that is continuously updated as the search proceeds. The statistical measure provides a queue for switching from one heuristic search process to another. Results from these tests are compared against those obtained from the four basic heuristic algorithms. Further, we systematically evaluate twenty-four 3-algorithm meta heuristics and compare their solution values and computing times against the four basic heuristic algorithms and the 2-algorithm meta heuristics. A 2-algorithm meta heuristic allows one heuristic (e.g., simulated annealing) to operate first, then switches to another (e.g., tabu search) at a point when the solution process has matured. A 3-algorithm meta heuristic uses a series of three standard heuristics (e.g., simulated annealing, threshold accepting, and tabu search) to located near-optimal solutions to the problem at hand. We hope our results and discussion can give a useful insight into intelligent and informed meta heuristic development in the forest planning field.

MATERIALS AND METHODOLOGY

We begin the discussion of our methods by describing the forest planning problem that we use to assess the combined meta heuristics. We then discuss the search techniques used to build a forest plan, along with the parameterization for each. A discussion of some preliminary work with combining algorithms, using break-point analysis, is then presented. Break-point analysis is then employed within the search processes to help us understand and evaluate the informed development of meta heuristics.

Problem Formulation

A vector-based geographic information system (GIS) database containing 1,123 polygons of stands covering 37,626 ha was used in the forest planning exercise. Forest stand polygons in this dataset were based on an actual southern United States forest land ownership. We modified polygon boundaries to ensure that the area of each stand polygon ranged from 24.3 ha to 48.6 ha (60 to 120 acres), because we assumed later that the maximum clearcut size was 48.6 ha. The initial forest age class distribution over the entire forest land was simulated as a uniformly distributed, near-normal forest with tree ages ranging from 1 to 30 years. This step was necessary since we were unable, through a confidentiality agreement, to disclose and use the original forest inventory provided by the cooperating company. As a result, approximately 1,300 hectares were assigned to each forest age class.

The spatial forest planning problem in this study was formulated with the objective of maximizing the net present value of planned management activities. We assumed that timber products were the only financially valued outcome. The planning horizon is 15 years long with 1-year planning periods. For simplicity, we also assumed that the only treatment on the forestland was to clearcut forested stands. Four constraints were considered. First, a unit restriction (URM) adjacency constraint [24] was assumed, under which any two contiguous stands are not allowed to be treated in the same planning period. Wood-flow constraints, which ensured sustainable and stable yields over the 15 year planning horizon were also assumed. In other words, the harvested volume in each time period should not deviate too far from the average harvested volume (maximum ±20% deviation in this case). An ending inventory constraint was also assumed, which prevents the depletion of timber stands at the end of planning horizon. Here, an amount equal to at least 90% of the original timber volume was required to remain at the end of the time horizon. Finally, a minimum harvest age constraint was assumed, where trees less than 20 years old are not considered for harvest. In sum, this is similar to a typical planning problem for a southern United States forestry company.

The problem formulation is described as the following:

Maximize

$$\sum_{t=1}^{T}\sum_{i=1}^{N}\left(X_{it}\left(\left(V_{it.saw}P_{saw}+V_{it.cns}P_{cns}+V_{it.pulp}P_{pulp}\right)-A_iC_r\right)/\left(1+d\right)^{t-0.5}\right) - \left(\frac{C_aT_A\left(\left(1+d\right)^t-1\right)}{i\left(1+d\right)^t}\right) \quad (1)$$

subject to

$$X_{it}+X_{jt}\le 1 \quad \forall i,j\in N_i \quad (2)$$

$$\begin{aligned} &0.8*\sum_{i=1}^{N}X_{it}V_{it} \le (\sum_{t=1}^{T}\sum_{i=1}^{N}X_{it}V_{it})/T \\ &\text{if } \sum_{i=1}^{N}X_{it}V_{it} > (\sum_{t=1}^{T}\sum_{i=1}^{N}X_{it}V_{it})/T \quad \forall\ t \\ &\sum_{i=1}^{N}X_{it}V_{it} \ge 0.8*(\sum_{t=1}^{T}\sum_{i=1}^{N}X_{it}V_{it})/T \\ &\text{if } \sum_{i=1}^{N}X_{it}V_{it} < (\sum_{t=1}^{T}\sum_{i=1}^{N}X_{it}V_{it})/T \quad \forall\ t \end{aligned} \quad (3)$$

$$\sum_{i=1}^{N}V_{il}\ge 0.9*\sum_{i=1}^{N}V_{i0} \quad (4)$$

$$\sum_{t=1}^{T}X_{it}\le 1 \qquad \forall\, i, X_{it}\in(0,1) \quad (5)$$

if $Age_{it}\ge 20$, then $X_{it}\in(0,1)$, otherwise $X_{it}=0$ (6)

where:

A_i = the area of management unit i

Age_{it} = the age of management unit i at time t period

C_a = the annual management cost (\$/unit land area)

C_r = the regeneration cost (\$/unit land area)

d = the discount rate assumed

i, j = an arbitrary harvested unit

N = the total number of harvest units

N_i = the set of all harvest units adjacent to unit i

P_{cns} = the stumpage price for chip-n-saw timber

P_{pulp} = the stumpage price for pulpwood

P_{saw} = the stumpage price for sawtimber

t = the period in which harvest activities occur

T = the total number of time periods in the planning horizon

T_A = the total planning area

V_{i0} = the total timber volume in the stands before any harvest activities

V_{il} = the timber volume left on the stands after the planning horizon

V_{it} = the timber volume harvested in time period t, from management unit i

$V_{it.cns}$ = the chip-n-saw volume harvested in time period t, from management unit i

$V_{it.pulp}$ = the pulpwood volume harvested in time period t, from management unit i

$V_{it.saw}$ = the sawtimber volume harvested in time period t, from management unit i

$$X_{it} = \begin{cases} 1 & \text{if management unit } i \text{ is treated in time period } t \\ 0 & \text{otherwise} \end{cases}$$

The decision variables are the X_{it} variables, and all others are either indices, parameters, or coefficients related to the planning model. This problem has been reported previously in a doctoral dissertation [25]. Equation 2 refers to the URM adjacency constraint for harvested areas. Here, only one of two adjacent timber stands can be scheduled for harvest in the same time period. The wood flow constraints (equation 3) control the amount of wood scheduled in each time period. Essentially, this set of constraints limits deviations in scheduled harvested volume in any time period (t), and requires that the deviation be no more or less than 20% from the other time periods. Equation 4 represents an ending-inventory constraint, where the remaining standing volume at the end of the time period be at least 90% of the standing volume that was available at the beginning of the time horizon. Equation 5 indicates that a stand can only be harvested once during the planning horizon. Finally, equation 6 represents the logic used to implement the minimum harvest age constraint. We used a growth and yield model developed for southern United States pine stands by the Plantation Management Research Cooperative (Warnell School of Forest and Natural Resources, University of Georgia). Prices for timber stumpage were obtained from Timber-Mart-South (fourth quarter of calendar year 2006), and were \$36.58 (U.S. dollars) per ton for potential sawtimber products, \$20.40 per ton for potential chip-n-saw products, and \$6.68 per ton for potential pulpwood products. The regeneration costs are \$606.14 per ha (site preparation, planting, seedling costs, and herbaceous weed control). The annual management cost is assumed to be \$11.12 per ha.

Heuristic Algorithms

The heuristics examined in this study include simulated annealing, threshold accepting, tabu search, and the raindrop method. All but the latter have been used extensively in forestry harvest scheduling research and practice. The raindrop method is a recently introduced heuristic that has been shown to very effectively solve certain kinds of forest harvest scheduling problems [23]. Each of these heuristics is briefly described below.

Simulated Annealing

Simulated annealing (SA) is inspired by the process of annealing of metals, and was first described in 1953 [26]. As a search process, simulated annealing began to be used in a widespread manner in the early 1980s [27]. A number of papers have shown the usefulness of simulated annealing in forest harvest scheduling [2, 4, 6, 9]. The search process is initiated with a high temperature parameter, and as the search proceeds the temperature cools off. The temperature acts, in part, as a threshold, and the search process terminates when the stopping criterion is met (e.g., the temperature gets too low). As the temperature declines, the search moves about in a random fashion around a limited group of good candidate solutions. If a change to a solution results in an improved solution, the change remains in the solution set. If a new solution does not result in an improvement, whether this new solution should be accepted or not depends on the resulting solution quality and a probability defined by the following equation:

$$P(T) = e^{-|S_c - S_p|/T_z} \tag{7}$$

where:

S_c = the current solution value

S_p = the previous solution value

T_z = the temperature at iteration z

$P(T)$ = probability critical value

The calculated value of $P(T)$ is then compared to a randomly drawn number between 0 and 1. The process accepts the change to the new, yet inferior solution if $P(T)$ is greater than the randomly drawn number. In essence, an inferior solution is likely to be accepted at a high temperature level (at the beginning of the search) and likely to be refused at a low temperature level (near the end of the search), since at a higher temperature the critical probability value is larger.

An essential component of a simulated annealing algorithm is the cooling schedule that it employs. The parameters required for a cooling schedule include the initial temperature (T_0) and the cooling function, so that $T_{z+1} = f(T_z)$. The cooling function can be very complicated, and may involve a self-adapting process at each temperature during the searching process. For simplicity, we chose to use a fixed cooling schedule in present study, which only includes an initial temperature and a cooling rate.

Threshold Accepting

Threshold accepting (TA) was introduced in 1990 [28] and applied to forest planning problems about a decade later [15, 16]. Threshold accepting is similar to simulated annealing, except there is no annealing criterion to compute. A small, random change to a solution is proposed, and if there is an improvement in solution quality, the proposed change is incorporated into the solution. The difference between this and simulated annealing lies in how one deals with changes to a solution that do not improve the quality of a forest plan. Instead of using a probability to determine whether the change is acceptable, any solution that is worse than the current solution value by more than the amount of a threshold value is rejected. The threshold value is initially large, allowing the search process to move relatively freely throughout the solution space. The threshold gets smaller as the search progresses, usually either using a geometric rate of change (new threshold = 0.995 *x* old threshold) or using a constant rate of change (new threshold = old threshold - *Y* dollars). At some point, when the threshold is very small, the search process terminates. Threshold accepting is viewed as a fast and effective heuristic that is easier to conceptually understand than simulated annealing.

Tabu Search

Tabu search was introduced in 1989 [29, 30], and has been applied to a number of forestry problems [5, 10, 31, 32], thus along with simulated annealing, tabu search is one of the most frequently used techniques in forest planning. Unlike other heuristic techniques, tabu search largely involves a deterministic component in its search activity, which may result in high computation costs. However, the deterministic search process may provide very good solutions that are close to the global optima if the search process can avoid becoming trapped in local optima. A basic tabu search process can be summarized as having two aspects: 1) a neighborhood local search, which has the goal of locating the best feasible solution in the nearby neighborhood (solution space) of the current solution; and 2) an improvement mechanism which attempts to use tabu tenure to avoid being trapped in the local optima. Other variations of tabu search may use diversification techniques to explore further the feasible solution space, or intensification techniques to exploit deeper areas surrounding elite solutions. However, in this study only the basic 1-opt tabu search is developed and tested.

The neighborhood developed examines all of the potential changes to the current solution, and the best of these is selected. This change could either increase the quality of a forest plan or decrease it. In the latter case, a change that reduces the quality the least is chosen, no matter how much reduction in solution quality occurs. The improvement mechanism is called the aspiration criteria. Within tabu search, a choice is generally forbidden (taboo) if it has been made recently (within the last *x* number of iterations, where *x* is the tabu tenure). However, if a choice is considered tabu, yet will result in the highest quality solution found thus far during the search, the change is accepted, and the tabu tenure is thus over-ridden. The search process develops a new neighborhood of potential changes to a solution with each iteration, and terminates when a pre-defined number of iterations have been used. A high computational cost may be incurred if the size of the neighborhood is large, which suggests that many potential changes must be examined before one is chosen. In simulated annealing and threshold accepting, only one (or a few, if constraint violations occur) potential changes are assessed with each iteration of a search process.

Raindrop Method

The raindrop method was first developed in 2006 [23] and is aimed at mitigating forest harvesting adjacency constraint violations (clearcuts that should not be placed next to one another) by radiating changes in a manner away from an initial forced choice. It is similar to the other heuristics in that it begins with a solution and allows random changes to be made. In most other heuristics described in the forestry literature, if this change causes any infeasibility (e.g., violating the adjacency constraints), the choice will be discarded and the heuristic will consider other choices. Instead, the raindrop method keeps this change (a forced choice), but records all the constraint violations, then attempts to mitigate the violations. Activities assigned to other land units are altered deterministically, perhaps creating more constraint violations that are further away (spatially) from the original forced choice. These additional constraint violations are recorded as well, and the process continues until all the constraint violations are mitigated. Decisions planned for land units physically closest to the forced choice are altered first, and constraint violations that might arise in a direction back to the forced choice in the original land unit are avoided. This change-violate-fix sequence continues until no constraint violations remain, thus ending one iteration of the model. After a certain number of iterations, the process reverts to the previous best solution if no improvement has been found. Therefore, the only two parameters required in raindrop method are the total number of iterations and the reversion rate. This method has been shown to produce consistently higher quality forest plans when the URM model of harvest adjacency is assumed in forest plans [31].

Parameterization of Heuristic Models

In this study, we used empirical methods to find the best parameter values for each algorithm by searching a wide range of possible parameter values and locating a narrow parameter interval. We determined that all values falling in this interval would more likely lead to steady and high-valued solutions. Although we can not guarantee the parameter values used are exactly the best choice for the problem, we are confident they are reasonable values and would allow us to generate high quality solutions.

For simulated annealing, an initial temperature value was tested that ranged from 10,000 to 5,000,000 degrees with an interval of 10,000. A cooling rate was tested using five different reduction values: 0.9999, 0.9995, 0.999, 0.995, 0.99. Similarly for threshold accepting, an initial threshold was tested that ranged from 10,000 to 1,000,000 US dollars with an interval of 10,000, and five different reduction values were assessed: 0.9999, 0.9995, 0.999, 0.995, 0.99. For tabu search, the tabu tenure we tested ranged from 100 to 20,000 iterations with an interval of 100. For the raindrop method, the reversion rate we tested ranged from 5 to 1000 iterations using an interval of 5. Except for the raindrop method, the other methods were terminated when solutions did not improve after a certain number of iterations. The raindrop method used the total number of iterations as a stopping criterion.

For simulated annealing, solution quality decreased as the cooling rate value assumed decreased. With geometric cooling rates of 0.99 and 0.995, solution values appeared to be lower in quality as compared to solutions with assumed cooling rate values of 0.999, 0.9995 and 0.9999. This suggested that we were not allowing free movement in the initial period of the search. Therefore, in our implementation of simulated annealing, we assumed a 0.9995 cooling rate value. When viewing more detailed results, we observed that the initial temperature did not matter too much as long as it was above about 7,000 degrees. After this point, the quality of solutions was stable with respect to the initial temperature, and the cooling rate thus had the most influence on solution quality. For these reasons, we assumed an initial temperature of 10,000 degrees.

Our initial tests of threshold accepting showed similar results. Threshold change values of 0.9999, 0.9995 and 0.999 seemed to provide high quality solutions as compared to threshold change values of 0.995 and 0.99. As with simulated annealing, we assumed 0.9995 was the geometric threshold change rate. To determine which initial threshold should be used, we examined more closely the solutions generated with initial threshold values between 1,000 and 30,000 US dollars. The quality of solutions stabilized after an initial threshold of about 15,000, so we chose to assume an initial threshold of 20,000 US dollars for the remainder of this work.

For tabu search, we located the stable interval for the tabu tenure, which ranged from 4,500 to 5,500 iterations. This represented about 1/3 of the potential choices available in adjusting a solution from one iteration to the next. Solutions produced using the tabu tenure in this interval maintained a high quality level, therefore we chose 5,000 iterations, the median of this interval, as the tabu tenure used in this work. The process terminated when the solution made no improvement after 10,000 iterations.

Our initial tests of the raindrop method did not suggest any pattern in solution quality with increases in the value of reversion rate. Therefore, we assumed a value of 5 iterations for this parameter, similar to what has been suggested by others [31].

Preliminary Analysis

A better understanding of the searching pattern of each individual algorithm would provide us insightful perspectives and logical reasons regarding how to combine different algorithms. Before developing combined meta-heuristics, we applied each standard algorithm to the same study problem, and observed and analyzed their different solution-development behaviors. Although summarized here, a complete analysis of the following discussion can be found in Li's dissertation [25].

Bettinger *et al*. [10] studied tabu search behavior for a forestry and wildlife problem, and described a typical search process by three phases: a hill-climbing phase, an adjustment phase, and a steady-state phase. We have also observed this behavior when using other algorithms when applied to forest planning problems. To further investigate the search patterns, we utilized break-point analysis techniques to detect significant changes in search behavior. Break-point analysis has been mostly used for analyzing economic, time series data. The foundation for estimating breaks in time series regression models was proposed in 1994 [33], extended to multiple breaks [34-36], and subsequently implemented as an algorithm [37]. The basic idea is to estimate break points by fitting multiple linear regression models simultaneously and minimizing the residual sum of squares. In our study, we tracked the searching path of a search algorithm by recording, at each iteration, the current solution value. We treated this search path as time series data, with iterations representing time slices, and current solution values as response values. We found two significant break points when using the four heuristics on our forestry problem. Based on these two break points, we divided the searching path into three intervals, which matched with three phases of their search behavior: a hill-climbing phase, an adjustment phase, and a steady-state phase [10]. In general, solution values increase very quickly in the first hill-climbing phase, slow down in the adjustment phase, eventually move to a steady-state phase.

Algorithm Integration

The main concentration of this work was placed on developing an intelligent mechanism for combining the different algorithms. In other words, we sought methods for automatically locating the integration points for switching the search from one algorithm to another during the search process. An integration point is the time (as denoted by the iteration number) during the search process where a second (or third) heuristic begins to operate, providing a different means for locating solutions of better quality. Simply using a fixed break point from the knowledge gained during the break point analysis as integration positions turned out to be a bad choice, because 1) the positions of the break points are constantly changing with different runs of the models; 2) the break point analysis was done after an entire solution was generated, even though we needed to decide during the generation of a solution where to stop one algorithm and start another one; and 3) each algorithm had its own internal mechanism which determines the searching path pattern.

Therefore, the phase separation was only meaningful for one algorithm, and thus there was no simple equivalence of the same phase between different algorithms. For example, the hill climbing phase of the raindrop method was longer, with respect to the best solution found, than the same hill-climbing phase observed with simulated annealing.

Since the purpose of combining different algorithms in a meta heuristic is to enhance the searching ability by taking advantage of the beneficial aspects of different algorithms, the best time to switch from one algorithm to another should be where the one algorithm's performance wanes. In order to quantify the subjective term "wane," we needed to know the relative solution improvement speed at each iteration. Because solution values increase and decrease constantly during any one search, it is moot to calculate the solution improvement speed by using the difference between the current solution value and the previous solution value. If one could generalize the searching path into a smooth line with a clear trend, ignoring all small deviations, one could derive the slope, or relative rate of change (i.e., the solution developing speed) for the process. A cubic spline smoothing technique [38] was utilized to address this task. Schoenberg [39] was one of the first to describe the smoothing technique, which was later described statistically [38]. Using this technique, cubic smoothing splines are fitted to each search path during the search process. In other words, while the heuristic algorithm searches for the best solution, cubic smoothing splines are simultaneously fitted to the current search path with a frequency of every 200 iterations. Since the fitted lines are smoothed, the first derivative (i.e., slope) can be obtained. Based on the value of the derivative, one could decide whether a switch should be made at that moment. A large positive derivative value indicates a strong and fast search process (a trend of increasing solution values), and a small positive derivative value indicates a slow and weak search process (a trend of minor increases in solution values). A negative derivative value suggests a trend of decreasing solution values. Derivative values that stay around zero for a certain number of iterations indicate that the search has stagnated. The events we considered to determine when to switch heuristic algorithms were when derivative values turned from positive to negative, and when derivative values became constant at zero for a number of iterations.

There are 12 possible two-heuristic combinations of the algorithms we studied (e.g., SA-TA, TA-SA, etc.). All permutations are examined because each heuristic has a different manner in which solutions are developed, since we were not convinced of the appropriate order (first or second). The four positions for linking them that we evaluated were: 1) the first point at which derivatives changed from positive to negative; 2) the second point at which derivatives changed from positive to negative (after having changed from negative to positive); 3) the third point at which derivatives changed from positive to negative; and 4) the point where the derivatives suggested search stagnation. We developed 50 solutions to the planning problem with each meta heuristic, using each of four integration positions. Each solution (run) began with a randomly-selected initial feasible solution. We calculated the mean and the standard deviation of the final solution values, the computing time, and the maximum solution value for each of the 50 solutions. An analysis of variance (ANOVA) was used to test whether any significant differences in the solution values occurred due to varying the integration positions, allowing us to identify the best integration position for each of twelve combinations of heuristics. Using the best integration position from the two-heuristic pairs, sets of three heuristics (3-algorithm meta heuristics) were then combined and applied to the same forestry planning problem. In sum, one heuristic is allowed to develop a high quality solution, then another is employed to fine-tune or adjust the solution, and finally a third is used to fine-tune or adjust the final solution. The point at which the meta heuristic switches from one to another is the focus of this work.

When simulated annealing or threshold accepting are the second or third algorithms in a meta heuristic, the initial temperature each assumed needed to be adjusted accordingly. In other words, through our empirical testing of these heuristics, we had located the appropriate initial temperature (SA) and threshold (TA) assuming they would be the first algorithm in a meta heuristic, not if they were the second or the third. When SA or TA are used as the second or third heuristic in a meta heuristic, we set the beginning temperature (SA) or threshold (TA) as a proportion of the original parameter assumption.

Validation

The preferred way to validate the performance of a heuristic is to locate the exact optimal solution to a problem, and compare it to the solutions provided by the heuristic. However, it is impractical to locate the exact optimum solution for the planning problem used in this manuscript, due to its size (1,123 management units, each with 16 potential choices) and due to the number of adjacency constraints necessary. Boston and Bettinger [4] listed other ways to validate heuristic solutions, including comparing heuristic solutions with solutions generated from other heuristic methods, locating the upper bound solution value through linear programming (using a relaxed version of the problem), and using the extreme value theory. In this study, we compared solutions produced by the 2-algorithm and 3-algorithm meta heuristics with solutions produced by standard heuristics. However, each of the heuristics used here has been validated for use in very similar forest planning problems [15, 23].

All of the standard heuristics and each of the meta heuristic algorithms were developed using the C# language under the Microsoft.Net platform. The break point analysis was performed using the statistical package 'strucchange' in the R software program [37].

RESULTS AND DISCUSSION

In examining the results, we first consider the four standard heuristics, then the 2-algorithm meta heuristics, and finally the 3-algorithm meta heuristics. Table **1** provides a summary of the solution quality and the computing time required for sets of 50 runs of each standard heuristic. Simulated annealing produced the highest mean solution value ($25.55 million), and threshold accepting produced the highest single maximum solution value ($25.73 million). The standard deviation for simulated annealing solutions was slightly lower than that for threshold accepting solutions. An ANOVA analysis indicated that there was no significant difference between these two methods in terms of solution quality, although their results were significantly different (p=0.05) than those provided by tabu search and the raindrop method. As for the average computing time needed for generating a single solution, simulated annealing and threshold accepting were also the two fastest heuristics (12.35 s per solution and 13.05 s per solution, respectively). Simulated annealing was slightly faster than threshold accepting and also had a tighter standard deviation for the computing time. The performance of tabu search suggested that this algorithm could produce good quality solution values, but it needed exceedingly longer time (158.57 s on average) to generate one solution compared with all other standard algorithms. These results were consistent with forest planning problems solved in Bettinger *et al.* [15]. The raindrop method did not seem to perform well on its own, and was deemed to be less efficient for the problem at hand (longer computing time and lower solution values) compared to simulated annealing and threshold accepting. While the raindrop method performs well in simpler forest planning problems [23], Zhu *et al.* [40] noted that the raindrop method may not perform well in forest planning problems containing wood flow constraints.

Table **2** provides results comparing the sets of 50 solutions generated while employing the four integration positions in association with the 2-algorithm meta heuristic models. Overall, improvements in the maximum solution values (over the standard heuristics) were observed using several models (e.g., SA-TA, TA-SA, Tabu-SA, and Tabu-TA). This suggests that either version of the SA-TA combination seems fruitful for problems such as the one solved here, as well models that begin with a slow deterministic algorithm (tabu search) and end with one of the fast algorithms (TA or SA). However, the TA-Tabu algorithm (fast start, slow finish) using the longest delay before integrating the algorithms (integration position 4) produced the single best solution and the highest average solution value. In a few of these 2-algorithm combinations, we could not determine a difference (p=0.05) in solution quality when considering the four integration positions (e.g., Tabu-SA, Tabu-TA, Raindrop-TA and Raindrop-Tabu). However, most of the 2-algorithm meta heuristic models showed strong significant differences among the different integration positions. Further, Tukey's multiple comparison was used to point out which of the integration position groups are different. We noticed that the difference mostly occurred between integration position 4 (stagnation) and other integration positions. From these results we observed that the mean solution values using integration position 4 were larger than those of other integration positions. We noted the best integration position for each two-heuristic pair, and used this in the development of each 3-algorithm meta heuristic.

When evaluating the 3-algorithm meta heuristics, we found that most of the combinations improved on the solution qualities obtained *via* the standard heuristics. A small set of meta heuristics (Table **3**) produced results which were significantly better (p=0.05) than other combinations of heuristics. Most of the better 3-algorithm meta heuristic combinations began with a relatively fast heuristic (SA or TA) to move quickly through the hill-climbing phase, then incorporated tabu search either in the adjustment or steady-state phases. The raindrop method was also employed in some of these meta heuristics for adjusting or fine-tuning the solution. In only one case high-quality solutions were obtained where the raindrop method was used during the hill-climbing phase, followed by a fast heuristic (TA) then a deterministic process (tabu search). Meta heuristics that had tabu search or the raindrop method employed during the hill-climbing phase were not as good as the other combinations tested. Compared with the best mean solution value from five standard algorithms (SA, $25.55 million), eighteen 3-algorithm meta heuristics (75%) produced higher mean solution values. Interestingly, while each of the top seven 3-algorithm meta heuristics produced results superior to most of the 2-algorithm meta heuristics, one 2-algorithm meta heuristic (TA-Tabu) produced better results. This limits the suggested usefulness of the notion that utilizing the search behavior of three heuristics could provide better solutions to the problem at hand.

The standard deviation of solution values for most 3-algorithm meta heuristics were around $0.08 or $0.09 million, with a few exceptions. Compared with standard

Table 1. A Summary of Solution Quality and Solution Speed for a Sample of 50 Runs of Four Standard Heuristics

Algorithms	Solution Quality				Computing Time	
	Mean (Million $US)	Standard Deviation (Million $US)	Maximum (Million $US)	ANOVA Groups	Mean (s)	Standard Deviation (s)
SA	25.55	0.08	25.72	A	12.35	0.11
TA	25.51	0.10	25.73	A	13.05	0.28
Tabu	24.79	0.27	25.47	B	158.57	49.92
Raindrop	21.76	0.25	22.41	C	63.77	3.89

SA = Simulated annealing, TA = Threshold accepting, Tabu = Tabu search, Raindrop = Raindrop method.

Table 2. A Summary of Solution Quality and Solution Speed for 50 Runs of Twelve 2-Algorithm Heuristics Using Four Different Integration Positions

Link Type	Integration Position	Solution Quality			Multiple Comparison Result		Computing time	
		Mean (Million $)	Standard Deviation (Million $)	Maximum (Million $)	*p*-Value	Groups[a]	Mean (s)	Standard Deviation (s)
SA-TA	1	25.56	0.13	25.78	0.002	AB	14.95	0.33
	2	25.55	0.13	25.78		B	14.98	0.14
	3	25.53	0.13	25.76		B	15.05	0.12
	4	25.62	0.08	25.78		A	52.46	2.24
TA-SA	1	25.61	0.08	25.75	0.000	AB	14.39	0.12
	2	25.57	0.09	25.72		BC	14.39	0.13
	3	25.57	0.07	25.73		C	14.40	0.12
	4	25.64	0.08	25.81		A	64.49	1.30
SA-Tabu	1	25.05	0.18	25.45	0.000	A	53.40	22.45
	2	24.99	0.22	25.34		A	53.23	17.93
	3	25.02	0.26	25.55		A	59.33	19.74
	4	25.77	0.10	25.97		B	70.61	8.17
Tabu-SA	1	25.60	0.09	25.79	0.692	A	15.27	0.25
	2	25.59	0.08	25.75		A	15.26	0.46
	3	25.58	0.07	25.76		A	15.64	1.01
	4	25.60	0.08	25.78		A	182.96	269.27
TA-Tabu	1	25.05	0.17	25.37	0.000	A	60.71	25.00
	2	25.09	0.17	25.36		A	63.13	22.12
	3	25.07	0.16	25.36		A	58.65	22.56
	4	25.89	0.10	26.09		B	92.71	12.84
Tabu-TA	1	25.54	0.13	25.77	0.123	A	15.04	0.31
	2	25.54	0.13	25.77		A	15.27	0.58
	3	25.50	0.13	25.73		A	15.57	0.95
	4	25.55	0.11	25.77		A	223.12	257.69
SA-Raindrop	1	23.50	0.31	24.06	0.000	A	75.18	3.88
	2	23.51	0.27	24.01		A	86.89	5.08
	3	23.52	0.32	24.07		A	98.88	3.58
	4	25.61	0.07	25.86		B	159.70	16.37
Raindrop-SA	1	25.49	0.08	25.65	0.039	A	482.09	5.24
	2	25.53	0.08	25.69		A	491.83	11.82
	3	25.50	0.07	25.65		A	488.89	10.45
	4	25.49	0.09	25.66		A	15.21	0.58
TA-Raindrop	1	23.47	0.24	23.84	0.000	A	73.92	4.09
	2	23.47	0.22	23.84		A	86.35	5.27
	3	23.57	0.22	23.95		A	98.21	3.66
	4	25.61	0.09	25.79		B	167.23	9.19
Raindrop-TA	1	25.47	0.12	25.68	0.267	A	483.12	5.22
	2	25.49	0.12	25.72		A	504.08	9.19
	3	25.49	0.14	25.75		A	525.90	8.42
	4	25.45	0.12	25.63		A	15.65	0.32
Tabu-Raindrop	1	21.67	0.27	22.44	0.000	A	76.06	5.70
	2	21.70	0.40	22.44		A	88.09	7.05
	3	21.81	0.39	22.78		A	100.99	4.70
	4	23.71	0.54	24.99		B	248.14	245.02
Raindrop-Tabu	1	24.33	0.41	25.13	0.402	A	563.25	36.94
	2	24.35	0.31	24.98		A	632.38	44.13
	3	24.30	0.41	25.09		A	564.99	34.19
	4	24.42	0.37	25.13		A	84.24	25.53

SA = simulated annealing, TA = threshold accepting, Tabu = tabu search, Raindrop = raindrop method.
[a] Significantly different groups within the "link type".

Table 3. A Summary of Solution Quality and Solution Speed for 50 Runs of Some of the 3-Algorithm Meta Heuristics

Model	Solution Quality						Computing Time	
	Mean (Million $US)	Standard Deviation (Million $US)	Maximum (Million $US)	Percentage Improved (%)[a]	*p*-Value[b]	ANOVA Groups	Mean (s)	Standard Deviation (s)
TA-Tabu-Raindrop	25.87	0.10	26.01	1.23	0.000	A	155.39	27.10
TA-Raindrop-Tabu	25.86	0.08	26.02	1.23	0.000	A	84.86	8.73
TA-Tabu-SA	25.85	0.09	26.04	1.18	0.000	AB	99.08	23.54
SA-TA-Tabu	25.85	0.09	25.98	1.16	0.000	AB	115.81	7.75
TA-SA-Tabu	25.85	0.08	26.03	1.16	0.000	AB	119.62	7.94
SA-Tabu-TA	25.84	0.08	26.02	1.12	0.000	AB	119.06	9.99
SA-Raindrop-Tabu	25.78	0.09	25.98	0.90	0.000	ABC	72.18	8.07
Raindrop-TA-Tabu	25.76	0.11	25.98	0.81	0.000	ABC	83.00	7.28
SA-Tabu-Raindrop	25.74	0.08	25.87	0.74	0.000	BCD	143.75	25.79
Tabu-Raindrop-SA	25.55	0.26	25.73	-0.01	0.538	FGHI	245.35	261.61
Raindrop-SA-TA	25.54	0.08	25.73	-0.03	0.687	FGHI	53.68	2.99
Tabu-Raindrop-TA	25.52	0.14	25.74	-0.13	0.926	GHI	265.77	289.30
Raindrop-Tabu-SA	25.49	0.09	25.68	-0.25	1.000	HI	206.37	259.12
Raindrop-Tabu-TA	25.47	0.14	25.72	-0.32	1.000	HI	282.72	289.87
Tabu-SA-Raindrop	25.46	0.48	25.74	-0.37	0.913	I	260.66	247.24

[a]Over the mean solution value generated with simulated annealing.
[b]Compared with solutions generated with simulated annealing.

algorithms TA ($0.10 million) and SA ($0.08 million), more than 85% of meta heuristics have the same variability of solution values as the standard TA and SA algorithms. But compared with standard tabu search ($0.27 million) and the raindrop method ($0.25 million), the standard deviations of most 3-algorithm meta heuristic results were much tighter. As for computing time, as a general trade-off, all 2-algorithm or 3-algorithm meta heuristics required much longer computing time to generate a good solution than did the standard heuristics. This extended computing time was expected, and not only included the running time for three different standard algorithms, but also included time for fitting smoothing splines to a search path and calculating the integration points for switching among algorithms. Some meta heuristics had very large standard deviation values for computing times, which was due to the trouble involved in finding the integration point for tabu search to switch to the next algorithm. In some of the runs of these models, the integration point appeared early in the search, but in other runs, it only occurred after a large number of iterations.

CONCLUSIONS

Rather than modifying a single standard heuristic, we chose to use standard heuristics in combination to understand if their respective search behaviors can be combined efficiently and effectively to solve a typical forest planning problem. For example, due to its deterministic component, standard 1-opt tabu search is an average performer in most forest planning problems, and it loses its effectiveness soon after a short hill-climbing phase. Therefore, the goal of this research was to determine whether the unique search behavior of multiple heuristics could be used to address forest planning problems. While our initial hypothesis was that an early transition from one heuristic to another would be beneficial in the search process, we found that the best integration point seems to be when the improvement in solution values of using one algorithm wanes (i.e., solution values stagnate). The only exception is when starting with tabu search, although as we have shown for this one problem, meta heuristics starting with tabu search are not as effective as the others. The best 2-algorithm meta heuristic combines a fast random search (TA) with a slower deterministic process (tabu search). The best 3-algorithm meta heuristic combines fast random search (TA) with a slower deterministic process (tabu search), and ended with a combined random-deterministic process (raindrop method). However, this 3-algorithm meta heuristic is relatively slow, when considering computing time, and the addition of the raindrop method does not seem to add to an increase in solution quality.

This work has shown that meta heuristics that combine the beneficial aspects of standard heuristics and how they behave in the three phases of a search, will generally produce consistently better solutions than standard heuristics alone. In general, a forest planning meta heuristic that begins with simulated annealing or threshold accepting, then utilizes tabu search or the raindrop method, seems to enable one to develop better solutions than when using the standard heuristics alone. In other words, starting with tabu search or the raindrop method is not as effective as starting with simulated annealing or threshold accepting. Ending with tabu search or raindrop method presents better results than ending with simulated annealing or threshold accepting. We demonstrate that determining when to switch, or integrate, algorithms can successfully be made based on the behavior of the search, rather than being made based on some *a priori* decision of the planner. While this transition seems to be when the quality of solutions generated by the prior heuristic stagnate, this knowledge can prevent premature switching of heuristic methods or prevent wasteful computational effort should the search extend well beyond a stagnation phase.

This study is limited in that only standard versions of the individual heuristic techniques were used to form meta heuristics. Improvements in tabu search performance, for example, have been reported by incorporating strategic oscillation [18] or 2-opt processes [5] into a search process. Further, we explored only the set of heuristic techniques which have been commonly used in forest management planning, and other techniques that have been reported through the broader operations research community may be of value in solving the type of forest planning problems described here. Therefore, one suggested improvement and future direction for this work may involve assessing a broader collection of heuristic techniques to combine and evaluate as meta heuristics. Another suggestion would be to focus on the top five or ten meta heuristics found here, and refine the individual heuristic techniques used to emulate the recent advances suggested earlier. However, we feel that our work represents an advance in exploring efficient and effective methods for locating near-optimal solutions to complex forest planning problems. Further refinement of the techniques used, along with the knowledge gained by intelligently understanding when to switch from one heuristic technique to another in a meta heuristic model, may lead to the development of search processes that can locate near-optimal solutions to combinatorial problems that are difficult (from problem formulation and computational time perspectives) to solve with exact techniques (e.g., integer programming).

REFERENCES

[1] Bettinger P, Sessions J. Spatial forest planning - to adopt, or not to adopt? J Forestry 2003; 101(2): 24-9.

[2] Lockwood C, Moore T. Harvest scheduling with spatial constraints: a simulated annealing approach. Can J Forest Res 1993; 23: 468-78.

[3] Baskent EZ, Keles S. Spatial forest planning: a review. Ecol Mod 2005; 188: 145-73.

[4] Boston K, Bettinger P. An analysis of Monte Carlo integer programming, simulated annealing, and tabu search heuristics for solving spatial harvest scheduling problems. Forest Sci 1999; 45: 292-301.

[5] Bettinger P, Boston K, Sessions J. Intensifying a heuristic forest harvest scheduling search procedure with 2-opt decision choices. Can J Forest Res 1999; 29: 1784-92.

[6] Nelson J, Liu G. Scheduling cut blocks with simulated annealing. Can J Forest Res 1994; 24: 365-72.

[7] Dahlin B, Sallnas O. Harvest scheduling under adjacency constraints: a case study from the Swedish sub-alpine region. Scand J Forest Res 1993; 8: 281-90.

[8] Chen BW, Gadow K. Timber harvest planning with spatial objectives, using the method of simulated annealing. Wiss Verl 2002; 121: 25-34.

[9] Baskent EZ, Jordan GA. Forest landscape management modeling using simulated annealing. Forest Ecol Manag 2002; 165: 29-45.

[10] Bettinger P, Sessions J, Boston K. Using tabu search to schedule timber harvests subject to spatial wildlife goals for big game. Ecol Mod 1997; 94: 111-23.

[11] Caro F, Constantino M, Martins I, Weintraub A. A 2-Opt tabu search procedure for the multiperiod forest harvesting problem with adjacency, greenup, old growth and even flow constraints. Forest Sci 2003; 49: 738-51.

[12] Boston K, Bettinger P. Combining tabu search and genetic algorithm heuristic techniques to solve spatial harvest scheduling problems. Forest Sci 2002; 48: 35-46.

[13] Falcão AO, Borges JG. Designing an evolution program for solving integer forest management scheduling models: an application in Portugal. Forest Sci 2001; 47: 158-68.

[14] Heinonen T, Pukkala T. A comparison of one- and two-compartment neighbourhoods in heuristic search with spatial forest management goals. Silva Fennica 2004; 38: 319-32.

[15] Bettinger P, Graetz D, Boston K, Sessions J, Chung W. Eight heuristic planning techniques applied to three increasingly difficult wildlife planning problems. Silva Fennica 2002; 36: 561-84.

[16] Bettinger P, Johnson DL, Johnson KN. Spatial forest plan development with ecological and economic goals. Ecol Mod 2003; 169: 215-36.

[17] Clements S, Dallain P, Jamnick M. An operational, spatially constrained harvest scheduling model. Can J Forest Res 1990; 20: 1438-47.

[18] Richards EW, Gunn EA. Tabu search design for difficult forest management optimization problems. Can J Forest Res 2003; 33: 1126-33.

[19] Cheng CT, Lin JY, Sun YG, Chau K. Long-term prediction of discharges in Manwan Hydropower using adaptive-network-based fuzzy inference systems models. Lecture Notes Comp Sci 2005; 3612: 1152-61.

[20] Cheng, CT, Chau KW. Fuzzy iteration methodology for reservoir flood control operation. J Am Water Resour Assoc 2001; 37: 1381-88.

[21] Zeng H, Pukkala T, Peltola H, Kellomäki S. Application of ant colony optimization for the risk management of wind damage in forest planning. Silva Fennica 2007; 41: 315-32.

[22] Bianchi L, Dorigo M, Gambardella LM, Gutjahr WJ. A survey on metaheuristics for stochastic combinatorial optimization. Nat Comput 2009; 8: 239-87.

[23] Bettinger P, Zhu J. A new heuristic method for solving spatially constrained forest planning problems based on mitigation of infeasibilities radiating outward from a forced choice. Silva Fennica 2006; 40: 315-33.

[24] Murray AT. Spatial restrictions in harvest scheduling. Forest Sci 1999; 45: 45-52.

[25] Li R. Integration of GIS techniques and heuristic algorithms to address spatial forest planning issues in the southern U.S. Ph.D. diss., University of Georgia 2007.

[26] Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E. Equation of state calculations by fast computing machines. J Chem Phys 1953; 21: 1087-92.

[27] Dowsland KA. Simulated Annealing. In: Reeves CR, Ed. Modern heuristic techniques for combinatorial problems. New York: John Wiley and Sons 1993.

[28] Dueck G, Scheuer T. Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. J Comp Phys 1990; 90: 161-75.

[29] Glover F. Tabu search – Part I. ORSA J Comput 1989; 1: 190-206.

[30] Glover F. Tabu search – Part II. ORSA J Comput 1990; 2: 4-32.

[31] Bettinger P, Sessions J, Johnson KN. Ensuring the compatibility of aquatic habitat and commodity production goals in eastern Oregon with a Tabu search procedure. Forest Sci 1998; 44: 96-112.

[32] Batten WH, Bettinger P, Zhu J. The effects of spatial harvest scheduling constraints on the value of a medium-sized forest holding in the southeastern United States. South J Appl Forestry 2005; 29: 185-93.

[33] Bai J. Least squares estimation of a shift in linear processes. J Time Ser Anal 1994; 15: 453-72.

[34] Bai J. Estimating multiple breaks one at a time. Econ Theory 1997; 13: 315-52.

[35] Bai J. Estimation of a change point in multiple regression models. Rev Econ Stat 1997b; 79: 551-63.

[36] Bai J, Perron P. Estimating and testing linear models with multiple structural changes. Econometrica 1998; 66: 47-78.

[37] Bai J, Perron P. Computation and analysis of multiple structural change models. J Appl Econom 2003; 18: 1-22.

[38] Chambers JM, Hastie TJ. Statistical models in S. Boca Raton, FL: CRC Press 1992.

[39] Schoenberg IJ. Spline functions and the problem of graduation. Mathematics 1964; 52: 974-50.

[40] Zhu J, Bettinger P, Li R. Additional insight into the performance of a new heuristic for solving spatially constrained forest planning problems. Silva Fennica 2007; 41: 687-98.

Received: October 11, 2009 Revised: January 25, 2010 Accepted: March 1, 2010