

Intensifying a heuristic forest harvest scheduling search procedure with 2-opt decision choices

Pete Bettinger, Kevin Boston, and John Sessions

Abstract: Forest management problems with even-flow and adjacency considerations are difficult to solve optimally. A heuristic search intensification process, which uses two types of decision procedures, changes to single-decision choices (1-opt moves) and changes to two-decision choices simultaneously (2-opt moves), was used in an attempt to locate feasible and efficient solutions to these problems. One-opt moves involve changing the timing of timber harvests for a single land unit and are commonly used in heuristic techniques. Two-opt moves involve swapping the harvest timing between two land units, which intensify the search process. We apply the procedures to two management problems, one with 40 land units and the other with 700 land units. The goal is to achieve the highest, and most even, flow of timber volume over five time periods, with adjacent units being unavailable for harvest in the same period. One-opt moves, used alone, allowed the search process to produce good feasible solutions to these management problems and to generate a relatively even spread (number) of harvests over the planning horizon. The use of 2-opt moves resulted in better solutions, although the number of harvests per time period remained static. These procedures, used alone, may not be appropriate for all problems, because of their nature and limitations.

Résumé : Les problèmes d'aménagement forestier pour lesquels des contraintes de contiguïté des superficies et d'uniformité des volumes doivent être prises en compte sont difficiles à résoudre de façon optimale. Un processus heuristique d'amélioration successive, utilisant deux types de procédures de prise de décision : le premier type comportait des changements vers des choix à option unique (1-opt) et l'autre des changements vers des choix à deux options simultanées (2-opt), a été utilisé dans une tentative visant à générer des solutions réalisables et efficaces pour ce type de problèmes. La procédure 1-opt est couramment utilisée dans les heuristiques et elle implique un changement dans le moment prévu pour la récolte du bois provenant de la même unité territoriale. La procédure 2-opt implique une permutation des moments prévus pour la récolte de deux unités territoriales ce qui intensifie le processus de recherche de solutions. Les deux procédures sont appliquées pour résoudre deux problèmes d'aménagement, un premier cas impliquant 40 unités territoriales et l'autre 700 unités territoriales. Le but est d'atteindre un volume de bois récolté le plus élevé et le plus uniforme possible durant une période de 5 ans, tout en s'assurant que des unités adjacentes ne soient pas récoltées au cours de la même période. La procédure 1-opt, utilisée seule, a permis au processus de recherche de produire des solutions acceptables et réalisables à ces problèmes d'aménagement tout en générant une répartition relativement uniforme du nombre de récoltes au cours de la période. La procédure 2-opt a donné de meilleurs résultats, bien que le nombre de récoltes par unité de temps soit demeuré statique. La nature et les limites de ces procédures font qu'elles peuvent ne pas convenir à tous les problèmes d'aménagement lorsqu'elles sont utilisées seules.

[Traduit par la Rédaction]

Introduction

The spatial arrangement of wildlife habitat and forest management activities is important for a number of reasons, such as in the adherence to regulatory restrictions, in the compliance with organizational goals and policies, and in the maintenance of aesthetic conditions. Forest regulations,

for instance, are placing increasingly restrictive limits on the size and spatial relationships of harvest units (Daust and Nelson 1993). As a result of a need to manage forest land within regulatory frameworks, forest management planning now often requires multiple resource goals and the use of spatial constraints on the selection of timber harvest units (O'Hara et al. 1989).

Forest planning algorithms that optimize the spatial arrangement of forest resources to meet a set of management goals vary from the more traditional optimizations techniques, such as linear or mixed integer programming (e.g., Hof et al. 1994) to nontraditional heuristic programming techniques (e.g., Murray and Church 1995). Forest planning problems that incorporate goals such as adjacency restrictions are combinatorial problems by nature; thus, as the problem size increases, the solution space also increases, yet at a disproportionately greater rate (Lockwood and Moore

Received September 9, 1998. Accepted July 29, 1999.

P. Bettinger.¹ Department of Forest Resources, Oregon State University, Corvallis, OR 97331, U.S.A.

K. Boston. Warnell School of Forest Resources, University of Georgia, Athens, GA 30602, U.S.A.

J. Sessions. Department of Forest Engineering, Oregon State University, Corvallis, OR 97331, U.S.A.

¹Corresponding author. e-mail: Pete.Bettinger@orst.edu

1993). Mixed integer programming and integer programming techniques have been used to produce management plans with adjacency concerns, but these types of techniques have severe limitations (directly related to problem size) when applied to large combinatorial problems (Lockwood and Moore 1993).

The use of heuristic techniques for forest management planning is becoming more prevalent, as problems that include complex, nonlinear goals now have an opportunity to be examined. Many of these types of goals (e.g., spatial and temporal distribution of elk habitat, as described in Bettinger et al. 1997) have traditionally been considered too complex to solve with traditional optimization techniques. Heuristic programming techniques have been applied to traditional forest harvest scheduling problems (Hoganson and Rose 1984) as well as to forest transportation problems (Nelson and Brodie 1990; Pulkki 1984; Weintraub et al. 1994; Murray and Church 1995; Weintraub et al. 1995), wildlife conservation and management (Arthaud and Rose 1996; Haight and Travis 1997; Bettinger et al. 1997), aquatic system management (Bettinger et al. 1998), and biological diversity (Kangas and Pukkala 1996).

Monte Carlo search is one type of heuristic technique and has been used in evaluating harvest unit and transportation system problems (Nelson and Brodie 1990), in linking strategic and tactical forest planning efforts (Nelson et al. 1991), in facilitating wildlife conservation planning (Haight and Travis 1997), and in addressing adjacency constraints (O'Hara et al. 1989; Daust and Nelson 1993; Clements et al. 1990; Jamnick and Walters 1993). Simulated annealing is a second type of heuristic programming technique and has been applied to harvest unit and transportation system problems (Murray and Church 1995) and adjacency problems (Lockwood and Moore 1993). Tabu search, a third heuristic technique, has been applied to the development of log bucking rules (Laroze and Greber 1997), to harvest unit and transportation system problems (Murray and Church 1995), and to wildlife and aquatic resource planning problems (Bettinger et al. 1997; Bettinger et al. 1998). Other random search heuristics (Yoshimoto and Brodie 1994; Yoshimoto et al. 1994), as well as nonlinear programming algorithms (Roise 1990), have been used to account for adjacency restrictions in forest planning models.

In any search process, consideration is given to the factors that can be modified during the search, the rules for selecting new solution configurations, and the length of time the search is allowed to proceed. We are concentrating here on the factors that can be modified during the search process. One-opt and 2-opt moves are common interchange approaches used in combinatorial optimization techniques and are specific instances of the λ -opt exchange approach (Martello and Toth 1990). One-opt moves involve a change in one attribute of one decision choice of a solution, creating a new solution. Two-opt moves involve simultaneous changes to an attribute of two decision choices. Most of the efforts associated with harvest unit location and scheduling have involved some form of 1-opt moves, where the status of a single attribute of a decision choice is altered. Murray and Church (1995) used a modification where harvest units affected by the change of the status of an adjacent unit were subsequently unscheduled.

In our case, we are intensifying a heuristic search process by switching the timing of the harvest of one harvest unit with that of another harvest unit. This involves the use of 2-opt moves, where an attribute of one decision choice is simultaneously changed with that of another. This reduces the magnitude of the impact on the objective function value and allows a heuristic technique to refine the solution to a management problem. An example may help clarify this point. Let us assume that sometime during a harvest scheduling search process, period 2 of a planning problem has 50 000 units of volume, and period 3 has 51 000 units. Now assume that the best available change to the current solution (in the process of finding other feasible solutions) unit 25, currently scheduled for period 2, is chosen now be harvested in period 3. Unit 25 may contribute 5000 units during period 2, and 5300 during period 3. After switching the harvest timing of unit 25, the resulting volumes for periods 2 and 3 would be 45 000 and 56 300, respectively. This change was assuming a 1-opt move was made. As an alternative to this, let us assume that sometime during a harvest scheduling search process a second unit, unit 19 (currently scheduled for harvest in period 3, with 6000 units available during period 2 and 6400 units available during period 3), will be involved in a 2-opt move with unit 25. Unit 25 will then be scheduled for harvest in period 3, and unit 19 for harvest in period 2. Given the volumes above (50 000 during period 2, 51 000 during period 3) period-2 volume would then change to 51 000 units (50 000 - 5000 + 6000), and period-3 volume would change to 49 900 (51 000 - 6400 + 5300). Thus the magnitude of the change in the solution using 2-opt moves would be much less (assuming even-flow of volume was a goal) than the change in the solution when 1-opt moves are made. Finally, it is not necessarily true that two 1-opt moves, made in sequence, would produce the same solution as a single 2-opt move. Therefore, the use of 2-opt moves may allow refinements in the exploration of the solution space and exploration of more of the solution space than the 1-opt moves allow.

Combining λ -opt moves with heuristic search processes has been addressed in the broader literature (e.g., Hanafi and Freville 1998) yet not in forest management literature. Thus, an illustration of the use of this approach to solve forest management problems, and a discussion of the implications and computational effort required for implementation in forestry, are the main contributions of this research.

Methods

We examine the use of a heuristic programming technique (tabu search) to solve a relatively simple forest management problem: how does one harvest n units of timber in m different time periods, with none of the harvests being adjacent to each other within a given time period and with the highest even-flow of volume across the m time periods achieved. Forest management problems can be formulated in a variety of ways, representing various goals of both public and private organizations. We have chosen here to maximize timber harvest subject to even-flow and adjacency constraints. Even-flow conditions are represented in the objective function. This type of goal emulates a desire for harvest (or cash flow) stability over time, which is not an uncommon goal.

Table 1. Volume (m³/ha) of the 40 units assumed for the harvest scheduling problem over five time periods.

Unit	Period 1	Period 2	Period 3	Period 4	Period 5
1	615.7	689.1	762.6	829.0	892.0
2	426.8	496.7	566.7	633.1	699.6
3	398.8	465.2	531.7	594.7	657.6
4	531.7	594.7	657.6	713.6	762.6
5	241.4	307.8	370.8	433.8	496.7
6	566.7	633.1	699.6	762.6	818.5
7	657.6	713.6	762.6	811.5	857.0
8	566.7	633.1	699.6	762.6	818.5
9	531.7	594.7	657.6	713.6	762.6
10	496.7	559.7	615.7	664.6	713.6
11	258.9	328.8	398.8	465.2	531.7
12	290.3	377.8	461.7	538.7	615.7
13	461.7	538.7	615.7	689.1	762.6
14	657.6	559.7	762.6	811.5	857.0
15	398.8	465.2	531.7	594.7	657.6
16	461.7	538.7	615.7	689.1	762.6
17	496.7	559.7	615.7	664.6	713.6
18	461.7	538.7	615.7	689.1	762.6
19	566.7	633.1	699.6	762.6	818.5
20	272.8	349.8	426.8	496.7	566.7
21	615.7	664.6	713.6	755.6	797.6
22	657.6	713.6	762.6	811.5	857.0
23	615.7	689.1	762.6	829.0	892.0
24	657.6	713.6	762.6	811.5	857.0
25	426.8	496.7	566.7	633.1	699.6
26	657.6	713.6	762.6	811.5	857.0
27	566.7	633.1	699.6	762.6	818.5
28	615.7	689.1	762.6	829.0	892.0
29	272.8	349.8	426.8	496.7	566.7
30	241.4	307.8	370.8	433.8	496.7
31	290.3	377.8	461.7	538.7	615.7
32	426.8	496.7	566.7	633.1	699.6
33	461.7	538.7	615.7	689.1	762.6
34	531.7	594.7	657.6	713.6	762.6
35	370.8	433.8	496.7	559.7	615.7
36	272.8	349.8	426.8	496.7	566.7
37	566.7	633.1	699.6	762.6	818.5
38	426.8	496.7	566.7	633.1	699.6
39	615.7	689.1	762.6	829.0	892.0
40	615.7	689.1	762.6	829.0	892.0

Note: Douglas-fir (*Pseudotsuga menziesii* (Mirb.) Franco) yields are adapted from McArdle and Meyer (1930) and assigned to each land unit, based on a random assignment of age and site index.

Alternatively, we may have formulated the problem to emphasize economic efficiency, by maximizing the net present value of the resulting management plan.

Two sizes of the problem are considered, one with 40 units (a 5 row by 8 column grid) and the other with 700 units (28 rows by 25 columns). Each of the units is 10 ha in size, and each are assumed to contain mature standing timber (the 40-unit volumes are listed in Table 1). We solve these problems first as relaxed linear programming (LP) problems, then attempt to solve the problem as integer programming (IP) problems, and finally solve the problem with a heuristic programming technique (tabu search). With the

use of the heuristic programming technique, two main approaches are evaluated for assessing the worth of decision choices: the first approach assesses a change in the status of individual land units (i.e., a 1-opt move, using a change in the timing of harvest of one unit from one period to another, including the opportunity to unschedule harvests, and subsequently reschedule from an unscheduled state); the second approach assesses a change in the status to two land units simultaneously (2-opt moves), by considering a swap of the harvest timing of two land units. We employ 1-opt and 2-opt moves as choices that can be made within a tabu search framework.

Harvest scheduling problems like the one described above can be categorized as an extension of a 0–1 multiple knapsack problem, where there are α items in a set and β knapsacks to which to assign these items (Martello and Toth 1990). The task is to select β subsets of the items such that some value is maximized, subject to some constraints, such as the capacities of the knapsack. Here, time periods are the correlates for knapsacks, and harvest units are the correlates for items. We next describe the problem formulation, then our implementation of tabu search, the heuristic technique we use to solve the management problem with both 1-opt and 2-opt moves.

Problem formulation

With the use of adjacency constraints, an assumption of nondivisibility of harvest units, and the structuring of even-flow deviations, the forest management problem described above falls into a class of integer-programming problems. We formulate the problem as one that attempts to achieve the highest even-flow volume among the five time periods, with minimal fluctuation of harvest volume between periods. The objective function for this harvest scheduling problem (HSP) is

$$[1] \quad \text{Maximize } H = \sum_{j=1}^m H_j$$

where j is the index of time periods, m is the total number of time periods, and H_j is the total harvest volume in period j . The first set of constraints are as follows:

$$[2] \quad H_j - H_{j+1} = 0, \quad j = 1, 2, \dots, m-1$$

where

$$[3] \quad H_j = \sum_{i=1}^n V_{ij} X_{ij}$$

and where H_j is the harvest volume in time period j , i is the index of land units, n is the total number of land units, V_{ij} is the timber volume contribution from unit i in time period j , and X_{ij} is the decision variable representing unit i in time period j . The second set of constraints force the solution to produce a strict even flow of timber volume per time period. This constraint may be relaxed (as it is in the IP model), since achieving strict even flow is often found to be quite difficult (if not impossible) with nondivisible harvest units. The relaxation of this constraint is achieved as follows:

$$[4] \quad H_j - H_{j+1} \leq \phi, \quad j = 1, 2, \dots, m-1$$

$$[5] \quad H_j - H_{j+1} \geq -\phi, \quad j = 1, 2, \dots, m - 1$$

where ϕ is the user-defined allowable deviation from strict even flow.

Three other types of constraints are also structured in the HSP. The first indicates that each unit can be harvested at most once during the planning horizon:

$$[6] \quad \sum_{j=1}^m X_{ij} \leq 1 \quad \forall i$$

The second constraint of HSP indicates that each variable is a binary or integer variable:

$$[7] \quad X_{ij} \in \{0, 1\} \quad \forall i, j$$

While the planning goal is to assume X_{ij} are binary, this constraint was relaxed in the LP implementation of the HSP model, allowing X_{ij} to take on a continuous form from 0 to 1.

The third constraint indicates that adjacent units will not be allowed to be harvested within the same time period. The adjacency constraint is employed by disallowing the selection of orthogonally adjacent units (i.e., on the north, south, east, west sides) for harvest during the same time period. The units and their adjacent neighbors in this problem are known a priori. To achieve this, we define for each unit i a set of neighbors U_i . The constraint for binary X_{ij} is then

$$[8] \quad (X_{ij} + X_{kj}) \leq 1 \quad \forall i, j, k \in U_i$$

This constraint indicates that, in time period j , only two scenarios are possible: (i) neither unit i nor its adjacent neighbors (k) will be scheduled for harvest (the sum, when considering unit i , during time period j , is 0); or (ii) only unit i or only one of its four adjacent neighbors (k) will be scheduled for harvest (the sum, when considering unit i , during time period j , is 1, since each X_{ij} is binary). In contrast to the constraint in eq. 6, which places a limit on the number of times a unit can be harvested, the constraint in eq. 8 limits the spatial placement of harvests within each time period j .

In the LP implementation of the HSP, X_{ij} are not binary, yet the idea that adjacent units will not be allowed to be harvested within the same time period was implemented using

$$[9] \quad \sum_{j=1}^m X_{ij} - 99M_{ij} \leq 1 \quad \forall i$$

$$[10] \quad M_{ij} \in \{0, 1\} \quad \forall i, j$$

$$[11] \quad (M_{ij} + M_{kj}) \leq 1 \quad \forall i, j, k \in U_i$$

Thus, when any X_{ij} (now continuous) enters the solution, M_{ij} receives the value of 1 (rather than 0). Adjacent M_{ij} can add up to either 0 or 1, allowing only one M_{ij} to enter the solution from the set of U_i during any time period. Therefore, while X_{ij} are now continuous, allowing some flexibility in achieving an even flow of volume over time, adjacent units will not be harvested in the same time period.

Modified formulation

The heuristic technique imposes each of these constraints, yet the objective function is slightly more complex, allowing the search process to find the highest even flow possible.

This slightly modified formulation is given as follows (denoted HSP2):

$$[12] \quad \text{Minimize } C = w_1 C_1 + w_2 C_2$$

where

$$[13] \quad H \geq H_j \quad \forall j$$

$$[14] \quad C_1 = |T - H|^\kappa$$

$$[15] \quad C_2 = \sum_{j=i}^{m-1} \sum_{l=j+1}^m (H_j - H_l)^2$$

and H_j , X_{ij} , and X_{kj} are as defined in eqs. 3, 6, 7, and 8, where w_1 is the weight applied to part 1 of the objective function, w_2 is the weight applied to part 2 of the objective function, T is the target even-flow harvest volume, H is the highest volume cut of the five time periods, and κ is an exponent ranging from 1.5 to 2.0.

The first part of the HSP2 objective function (C_1) draws the volume harvested per period towards a target volume goal. The exponent was set to 1.5 for the 40-unit problem, and 2.0 in the 700-unit problem, based on numerous trial runs of the heuristic, where the range of the exponent varied from 1.0 to 3.0. Our empirical evidence suggested that an exponent lower than 3.0 provided solutions which forced the level of harvest toward the target while not over-shadowing the even-flow portion of the objective. T was specified as the LP maximum harvest volume per period. The second part of the objective function (C_2) penalizes the solution for deviations among harvest volumes per period, by assessing the differences among the 10 combinations of 2 harvest periods. This drives the solutions toward an even flow of timber harvest volume each period. The weights (w_1 and w_2) can take on any value to further emphasize one portion of the objective over the other, yet we apply a value of 1.0 to each. The goal of formulating the problem as such was to provide the highest, and most even, harvest volumes. Since a strict even flow is probably not achievable, some other type of formulation was required. As a result, the HSP2 formulation of the problem is not exactly equivalent to the HSP formulation, and a comparison among these approaches must be made with this in mind.

Tabu search

Tabu search originated as a method for solving real-world combinatorial problems in scheduling and has been successfully applied to a number of important problems outside of forestry and wildlife management, such as telecommunications, transportation, shop sequencing, machine scheduling, and layout and circuit design problems (Glover 1990; Glover and Laguna 1993). Within forestry it has been applied to problems formulated for scheduling timber harvests subject to adjacency (green-up) requirements (Murray and Church 1995) and for meeting spatial goals for elk (Bettinger et al. 1997) and aquatic habitat (Bettinger et al. 1998).

Tabu search, in general, is a hill-climbing procedure consisting of two key elements: the search is constrained by considering certain choices as forbidden (i.e., tabu), and the search can be freed by a memory function that allows

“strategic forgetting” that certain choices are forbidden (Glover 1989). One of the chief limitations of hill-climbing procedures is that they generally stop processing once no improving moves are available; thus, the local optimum that is obtained may not be the global optimum to the problem at hand. Tabu search allows a search process to continue searching beyond local optima, without being confounded by the absence of improving moves and without following a sequence of moves that may ultimately lead back (i.e., cycle back) to a previously defined solution (Glover 1989). The key is to constrain the search in a manner that allows it to select improving moves, while assuring that the search will not revisit previously visited solutions except by following a path not previously traveled (Glover 1990).

Two essential components of tabu search include the development of a “neighborhood” of choices, or moves, and the tracking of a list of choices that may not be available for the development of a new configuration of the solution (i.e., they are tabu). We describe below the development and use of the neighborhoods in our implementation of tabu search and the methods we use to determine which moves are tabu.

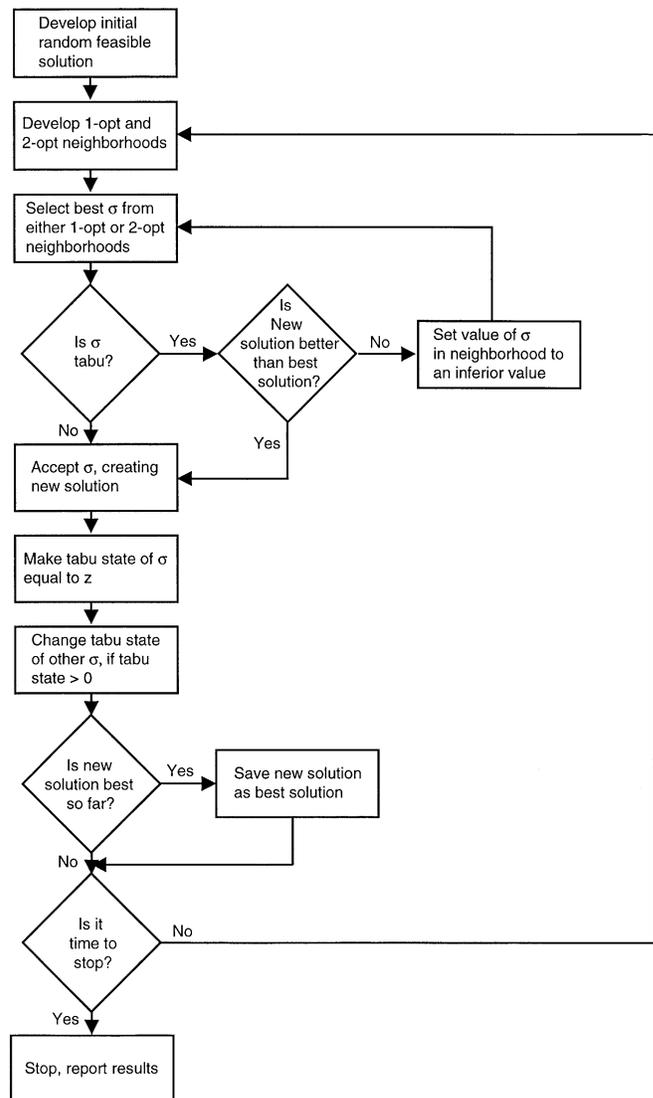
Development and use of neighborhoods

A neighborhood consists of all possible changes to a solution, x , involving a potential move (σ), or modification of the current solution. Moves here consist of the potential change to the current solution and are described by their impact on the objective function value when the harvest timing of land units are changed. For example, if a certain land unit were currently scheduled to be harvested in period 5 of a 5-period analysis, four σ would be available to the current solution and would each involve changing the harvest timing of that unit to periods 1–4. A fifth σ could also be used to unschedule the land unit for harvest. After a random start (Fig. 1), all possible σ are assessed and held in arrays, or what are termed the “neighborhoods.”

The search process then selects for consideration one σ from an assessment of the neighborhood(s). The σ selected represents either the best possible improvement in the objective function value (e.g., net present value) of a solution, or the least deterioration of the objective function value (Voß 1993), by searching the entire neighborhood(s) for the σ that best meets these conditions. We examine the use of two types of neighborhoods: one where all σ are simply a change in harvest timing of a single land unit (1-opt moves) and the other where σ represent the swapping of harvest timing among two units (2-opt moves).

The neighborhood for 1-opt moves ($N(x, \sigma_{i,j})$) of the current solution x is a set of solutions that can be reached from x by a move $\sigma_{i,j}$ (the choice of harvesting unit i in period j). All $\sigma_{i,j}$ are described by the potential objective function value, which represents the addition of a single $\sigma_{i,j}$ to x . The neighborhood for 2-opt moves ($N(x, \sigma_{i,j}, \sigma_{i',j'})$) consists of a set of solutions that can be reached from x by a move (the swapping of harvest timing (j) of land unit i , with the timing (j') of unit i'). These types of moves belong to a class of moves that require more complex rules and memory structures (Glover 1990; Glover et al. 1995). Here, 2-opt moves are considered as a composite move, where the value (harvest timing) of one decision variable is swapped with that of another.

Fig. 1. A flow chart of the heuristic tabu search process.



As mentioned earlier, two versions of the tabu search heuristic are to be compared: one where only the 1-opt neighborhood is developed and used and the other where both the 1-opt and 2-opt neighborhoods are developed and used. When the heuristic is operating with the assumption that only 1-opt moves are available, only one neighborhood is developed and examined. When 2-opt moves are assumed to be available, both neighborhoods are used, and a choice is made after examining both neighborhoods.

Each of the heuristic runs are started with a randomly defined initial feasible solution. We have found through experimentation that, when both neighborhoods are used, the search first (with the 1-opt moves) tries to even out the number of harvests per period and, second (with the 2-opt moves), refines the solution. By not using 1-opt moves to even out the harvests, the solutions may never reach reasonable potential, since the number of harvests per period are randomly assigned and would shift, but essentially never change. Further results of the experimentation have shown that the 1-opt moves are only utilized in the first few iterations of the heuristic.

Table 2. LP solution, IP solution, and average harvest volume per period (m³), maximum, minimum, and SD of solutions from a heuristic technique that used 1-opt moves, and 1-opt and 2-opt moves (40-unit problem).

	Period 1	Period 2	Period 3	Period 4	Period 5
LP solution					
Volume	55 034	55 034	55 034	55 034	55 034
IP solution^a					
Volume	54 324	54 289	54 289	54 849	54 254
Heuristic technique, 1-opt moves					
Mean volume	53 578	53 529	53 572	53 556	53 597
Maximum	54 360	54 501	54 428	54 500	54 464
Minimum	52 890	52 473	52 855	52 609	52 820
SD	327.3	336.6	318.4	369.3	321.8
Best volume ^b	53 519	53 522	53 449	53 554	53 344
Heuristic technique, 1-opt and 2-opt moves					
Mean volume	53 616	53 615	53 618	53 615	53 618
Maximum	54 569	54 570	54 570	54 535	54 534
Minimum	52 469	52 471	52 470	52 505	52 470
SD	347.0	346.1	348.7	346.9	347.5
Best volume ^b	54 569	54 570	54 570	54 535	54 534

^aThe IP results were the best observed, yet not optimal, results and were 1.4% lower than LP results.

^bResults of the best solution found during the search process.

Determining which moves are tabu

It is possible, during the operation of a heuristic such as this, that $\sigma_{i,j}$ or $\sigma_{i,j}\sigma_{i',j'}$ can be selected in a regular, recurring, periodic manner. This may happen as the period of harvest of a unit changes, then changes back to its original state. A short-term memory tabu restriction disallows the selection of a $\sigma_{i,j}$ or $\sigma_{i,j}\sigma_{i',j'}$ that was recently selected (and thus considered tabu), unless formal acceptance will result in an objective function value that is better than any previously observed objective function value (i.e., meeting the “aspiration criteria”). The use of aspiration criteria frees the search to allow moves by strategically forgetting one of the rules of the search (i.e., that some moves are tabu). Thus, it provides added flexibility in the search by overriding the tabu status of a move. While tabu restrictions in the form of short-term memory are essential to the search, aspiration criteria are not. Here, we use both tabu restrictions and aspiration criteria to guide the search process. Regardless of whether aspiration criteria are obtained, $\sigma_{i,j}$ or $\sigma_{i,j}\sigma_{i',j'}$ recently brought into a solution may not be eligible for selection again until a user-defined number of iterations from the current iteration of the model. This type of short-term memory essentially keeps the model from cycling back to a local optimum that has already been identified (Voß 1993; Murray and Church 1995).

Glover (1990) notes that there most likely exists a robust range of the number of iterations, or lengths of short-term memory (z), that allow heuristics to work effectively for obtaining progressively improved solutions, by driving the search beyond local optima. To determine the tabu state (z) for the 1-opt move neighborhood of the 40-unit problem, we started the heuristic from a fixed starting position (i.e., a fixed feasible solution), used z values that ranged from 20 to 180, and allowed the heuristic to run for 25 000 iterations. Since the total number of decision choices was 240 in the 1-opt move neighborhood, this range was very appropriate. We found that the solution values were very similar when z

ranged from 60 to 80. When we allowed choices to be made from both the 1-opt and 2-opt move neighborhoods, a single z was used for both neighborhoods. Here, we ran the heuristic with z values ranging from 20 to 100 and found solution values to be very similar where z ranged from 60 to 85. When we experimented with z values for the 700-unit problem, we used values ranging from 50 to 1000. We found that the solution values were very similar when z ranged between 50 and 150.

Therefore, for comparative purposes, we selected a z of 75 to use when comparing the two techniques (1-opt moves alone, 1-opt and 2-opt moves together) in both the 40-unit and 700-unit problems. We then generated solutions with each technique, each starting with a randomly defined, feasible solution to the problem. For the 40-unit problem the heuristic was permitted to run for 25 000 iterations in generating each of 200 solutions, where one iteration consisted of the acceptance of a $\sigma_{i,j}$ or $\sigma_{i,j}\sigma_{i',j'}$ into x . The best combination of harvest timing for the 40 units was stored in computer memory for each run of the heuristic.

For the 700-unit problem, we generated 100 solutions using 1-opt moves, and 18 solutions using both 1-opt and 2-opt moves. Further, the number of iterations for the 1-opt and 2-opt moves was decreased to 2000, since through experimentation we always found the best solution within 1000 iterations. This was not the case with 1-opt moves alone, nor with the 40-unit problem, and those we continued to run with 25 000 iterations.

Results

The relaxed LP solution to the 40-unit problem produced even-flow volumes of 55 034 m³ per period (Table 2). This solution was generated in about 8 s using LINDO (LINDO Systems, Inc. 1993) software on a personal computer equipped with 32 Mb of RAM and a 200 MHz processor chip. The IP solution was allowed to deviate in timber

Table 3. LP solution, IP solution, and average harvest volume per period (m^3), maximum, minimum, and SD of solutions from a heuristic technique that used 1-opt moves, and 1-opt and 2-opt moves (700-unit problem).

	Period 1	Period 2	Period 3	Period 4	Period 5
LP solution					
Volume	973 568	973 568	973 568	973 568	973 568
Heuristic technique, 1-opt moves					
Mean volume	941 294	941 289	941 572	941 638	941 716
Maximum	944 460	944 843	944 590	945 420	946 212
Minimum	935 884	936 236	938 124	935 098	936 633
SD	1 471.8	1 556.6	1 405.3	1 685.0	1 819.5
Best volume ^a	943 998	944 843	944 107	944 162	943 753
Heuristic technique, 1-opt and 2-opt moves					
Mean volume	946 241	946 266	946 243	946 236	946 206
Maximum	946 692	946 661	946 796	946 778	946 701
Minimum	945 679	945 608	945 686	945 805	945 582
SD	278.6	242.3	291.9	276.7	277.3
Best volume ^a	946 692	946 661	946 796	946 778	946 701

^aResults of the best solution found during the search process.

volume from one time period to the next, and we tried several iterations until a deviation of 500 m^3 per period never returned a feasible solution in a suitable amount of time (>5 h). The best IP solution, while not the optimal IP solution, was about 1.15% of volume lower than the LP solution. The heuristic techniques produced feasible solutions in, on average, 4 and 9 min, respectively, for the 1-opt, and 1-opt and 2-opt techniques. The best solution from each was within 2.83 and 0.87% of volume per period of the LP solution for the 1-opt, and 1-opt and 2-opt techniques, respectively. Every solution generated with both 1-opt and 2-opt moves was better than those generated solely with 1-opt moves.

The LP solution was for a relaxed problem, where the decision variables were not considered binary, and allowed a single unit to be harvested in more than one time period. The IP solution was the best solution located in a timely manner for the fully constrained problem, when experimenting with narrowing ranges of allowable deviations of harvest volumes among the five time periods. We attempted several IP models with varying ranges of ϕ , and most of these ran for three or four consecutive days without terminating at a proven optimal solution. These findings are similar to those reported by Glover and Kochenberger (1995), and our results indicate that forest management problems with even-flow and adjacency constraints may be very difficult to solve optimally. We were also creative, placing the entire "best" tabu search solution from the 40-unit problem as a hard constraint in the IP model formulation of the 40-unit problem. The IP technique recognized this solution as the optimal solution, although it still may not necessarily be the global optimal solution to the problem. We then placed one half of the "best" tabu search solution as a hard constraint in the IP model. This IP technique terminated without locating a proven optimal solution.

The relaxed LP solution to the 700-unit problem produced even-flow volumes of $973 568 \text{ m}^3$ per period (Table 3). This solution was generated in about 34 s using CPLEX (CPLEX Optimization Inc. 1998) software on a personal computer equipped with 128 Mb of RAM and a 233 MHz processor

chip. The IP formulation was also run in CPLEX, with varying levels of ϕ , yet we did not obtain a feasible solution in a reasonable amount of time. The HSP2 techniques produced feasible solutions in, on average, 57 and 215 min, respectively, for the 1-opt and 1-opt and 2-opt techniques. The best solution from each was within 3.02 and 2.76% of volume per period of the LP solution for the 1-opt and 1-opt and 2-opt techniques, respectively. Here again, every solution generated with both 1-opt and 2-opt moves was better than those generated solely with 1-opt moves. The best solution generated from using both the 1-opt and 2-opt moves was also about 2600 m^3 higher per period than that produced by the 1-opt moves alone.

Discussion

The use of 2-opt moves allowed the search procedure to fine tune the solution because the changes in the objective function value are not as severe as with using a 1-opt neighborhood alone, where changes are made simply to the status of individual harvest units. As a result, much better solutions are generated from the use of 1-opt and 2-opt moves than from 1-opt moves alone. One might ask whether the 1-opt neighborhood is even needed. We believe that it is for this point alone: to allow the distribution of harvest units over time to even out. Given a random starting solution, one might have a very uneven distribution of harvests over time. One-opt moves will allow the distribution of harvests to become more even; 2-opt moves, as developed here, will not. Therefore, if we simply utilize 2-opt moves with a random starting solution, the distribution of the number of harvest units per period will never change. One might further consider just setting the number of harvest units to an equal number across time and only use a 2-opt move neighborhood. However, once this number is set, it never has the chance to change, and one will not be certain whether the full range of the solution space was given the opportunity to be sampled.

The advantages of using 1-opt moves are that the search process to produce feasible solutions required less computer

logic (as compared with 2-opt moves) and less time to produce a good feasible solution. However, these advantages must be balanced with the fact that the best of the solutions produced was not as good as the worst solution produced when both 1-opt and 2-opt moves were used. Each $\sigma_{i,j}$ in the 1-opt neighborhood represented a large fluctuation from the previously measured objective function value, never allowing the heuristic to fine tune the results.

The main disadvantages of using the 2-opt moves are the expense (time and cost) of the additional logic required for the code of the heuristic, additional verification processes, and additional computer processing time. The additional logic included not only the development and assessment of a separate neighborhood but also the tracking of a second short-term tabu state list and the tracking of the neighborhood from which each move was obtained. For larger problems than that described here, the additional computer processing time may increase exponentially with the number of additional decision choices. We have shown here that, to move from a 40-unit problem to a 700-unit problem, the time required to generate a solution was not proportional to the increase in solution time, even though the 700-unit problem ran for only 2000 iterations. Therefore, one may wonder whether both 1-opt and 2-opt neighborhoods are needed in large problems, since fluctuations in the objective function value generated by 1-opt moves may not be as influential as those in smaller problems. We have shown that the fluctuations are just as influential. The answer, however, may lie in a trade-off analysis comparing the amount of time required to incorporate 2-opt moves in a heuristic in conjunction with the extra time required to arrive at a solution, with the risks associated with making and implementing an inferior management plan. Each organization considering heuristic techniques to assist in developing management plans should balance their perception of the additional computational effort with the risk of making an inferior decision.

The heuristic techniques described here could be further enhanced with other intensification or diversification techniques. For example, the frequency at which individual land units are considered as decision choices may be tracked. With this information, one may develop logic that requires land units to be considered tabu after they have been considered an assumed number of times during the operation of the heuristic. Also, genetic algorithms can be developed to diversify the search process (Glover and Laguna 1993) by combining parts of two local optima, creating a new solution from which the search process then proceeds. These types of techniques are left for future investigation, however, and were not evaluated here.

While the heuristic technique produced a solution that was better than the IP solution, we do not intend to imply that IP techniques are inferior. It was quite difficult to produce an IP solution to this problem. Many variations of IP approaches to solving forest management problems with adjacency restrictions have been explored (e.g., Snyder and ReVelle 1997). However, our intent was not to compare the heuristic techniques against several formulations of IP problems; our intent was to show that, perhaps, an intensification of a heuristic technique could produce better solutions than a heuristic technique that did not include this intensification procedure. Therefore, the IP solution presented here was

simply to show an effort on our part to solve the same problem with the traditional IP formulation.

Conclusions

When management problems consist of utilizing goals that do not lend themselves to be easily formulated, or solved, with traditional optimization techniques, other approaches are available, such as heuristic programming techniques. What we have shown here is that a tabu search heuristic, involving a single neighborhood consisting of changing the status of a single decision choice, may not produce a set of solutions that are as good as an approach that utilizes two neighborhoods, the second evaluating swaps in harvest timing among land units. The first neighborhood forces a general evenness of harvests over time, yet the second fine tunes the results.

The results provided here are significant for forest managers and planners. We have demonstrated that the use of 2-opt moves to simultaneously change the timing of harvest of two land units can provide better solutions to management problems that include both even-flow and adjacency constraints. While the use of tabu search alone, and of 2-opt moves alone, have been thoroughly studied in the broader operations research literature, the use of both of them together has not, and until now these two techniques have not been illustrated in a forest management problem. This is important, since many forestry organizations are considering heuristic techniques to help guide management decisions. Therefore, an examination of intensification and diversification techniques for heuristics search processes may be of interest, if these techniques can help provide better solutions to management problems. Each organization, of course, would need to assess the costs of incorporating intensification and diversification techniques into their quantitative forest planning models and balance these costs against the risk that an inferior decision may be made if these techniques are not used.

References

- Arthaud, G.J., and Rose, D. 1996. A methodology for estimating production possibility frontiers for wildlife habitat and timber value at the landscape level. *Can. J. For. Res.* **26**: 2191–2200.
- Bettinger, P., Sessions, J., and Boston, K. 1997. Using tabu search to schedule timber harvests subject to spatial wildlife goals for big game. *Ecol. Modell.* **94**: 111–123.
- Bettinger, P., Sessions, J., and Johnson, K.N. 1998. Ensuring the compatibility of aquatic habitat and commodity production goals in eastern Oregon with a tabu search procedure. *For. Sci.* **44**: 96–112.
- Clements, S.E., Dallain, P.L., and Jamnick, M.S. 1990. An operational, spatially constrained harvest scheduling model. *Can. J. For. Res.* **20**: 1438–1447.
- CPLEX Optimization Inc. 1998. CPLEX. CPLEX Optimization Inc., Incline Village, Nev.
- Daust, D.K., and Nelson, J.D. 1993. Spatial reduction factors for strata-based harvest schedules. *For. Sci.* **39**: 152–165.
- Glover, F. 1989. Tabu search—part I. *ORSA J. Comput.* **1**: 190–206.
- Glover, F. 1990. Tabu search—part II. *ORSA J. Comput.* **2**: 4–32.

- Glover, F., and Kochenberger, G.A. 1995. Critical event tabu search for multidimensional knapsack problems. *In Metaheuristics: theory and application. Edited by I.H. Osman and J.P. Kelly.* Kluwer Academic Publishers, Dordrecht, the Netherlands. pp. 407–427.
- Glover, F., and Laguna, M. 1993. Tabu search. *In Modern heuristic techniques for combinatorial problems. Edited by C.R. Reeves.* John Wiley & Sons, Inc., New York. pp. 70–150.
- Glover, F., Kelly, J.P., and Laguna, M. 1995. Genetic algorithms and tabu search: hybrids for optimization. *Comput. Oper. Res.* **22**: 111–134.
- Haight, R.G., and Travis, L.E. 1997. Wildlife conservation planning using stochastic optimization and importance sampling. *For. Sci.* **43**: 129–139.
- Hanafi, S., and Freville, A. 1998. An efficient tabu search approach for the 0–1 multidimensional knapsack problem. *Eur. J. Oper. Res.* **106**: 659–675.
- Hof, J., Bevers, M., Joyce, L., and Kent, B. 1994. An integer programming approach for spatially and temporally optimizing wildlife populations. *For. Sci.* **40**: 177–191.
- Hoganson, H.M., and Rose, D. 1984. A simulation approach for optimal timber management scheduling. *For. Sci.* **30**: 220–238.
- Jamnack, M.S., and Walters, K.R. 1993. Spatial and temporal allocation of stratum-based harvest schedules. *Can. J. For. Res.* **23**: 402–413.
- Kangas, J., and Pukkala, T. 1996. Operationalization of biological diversity as a decision objective in tactical forest planning. *Can. J. For. Res.* **26**: 103–111.
- Laroze, A.J., and Greber, B.J. 1997. Using tabu search to generate stand-level, rule-based bucking patterns. *For. Sci.* **43**: 157–169.
- LINDO Systems, Inc. 1993. LINDO, version 5.3 edition. LINDO Systems, Inc., Chicago, Ill.
- Lockwood, C., and Moore, T. 1993. Harvest scheduling with spatial constraints: a simulated annealing approach. *Can. J. For. Res.* **23**: 468–478.
- Martello, S., and Toth, P. 1990. Knapsack problems: algorithms and computer implementations. John Wiley & Sons, New York.
- McArdle, R.E., and Meyer, W.H. 1930. The yield of Douglas fir in the Pacific Northwest. U.S. Dep. Agric. Tech. Bull. No. 201.
- Murray, A.T., and Church, R.L. 1995. Heuristic solution approaches to operational forest planning problems. *OR Spektrum (Oper. Res.)* **17**: 193–203.
- Nelson, J., and Brodie, J.D. 1990. Comparison of a random search algorithm and mixed integer programming for solving area-based forest plans. *Can. J. For. Res.* **20**: 934–942.
- Nelson, J., Brodie, J.D., and Sessions, J. 1991. Integrating short-term, area-based logging plans with long-term harvest schedules. *For. Sci.* **37**: 101–122.
- O'Hara, A.J., Faaland, B.A., and Bare, B.B. 1989. Spatially constrained timber harvest scheduling. *Can. J. For. Res.* **19**: 715–724.
- Pulkki, R. 1984. A spatial database – heuristic programming system for aiding decision-making in long-distance transport of wood. *Acta For. Fenn.* No. 188.
- Roise, J.P. 1990. Multicriteria nonlinear programming for optimal spatial allocation of stands. *For. Sci.* **36**: 487–501.
- Snyder, S., and ReVelle, C. 1997. Dynamic selection of harvests with adjacency restrictions: the SHARe model. *For. Sci.* **43**: 213–222.
- Voß, S. 1993. Tabu search: applications and prospects. *In Network optimization problems. Edited by D.Z. Du and P.M. Pardalos.* World Scientific Publishing Co., Singapore. pp. 333–353.
- Weintraub, A., Jones, G., Magendzo, A., Meacham, M., and Kirby, M. 1994. A heuristic system to solve mixed integer forest planning models. *Oper. Res.* **42**: 1010–1024.
- Weintraub, A., Jones, G., Meacham, M., Magendzo, A., Magendzo, A., and Malchuk, D. 1995. Heuristic procedures for solving mixed-integer harvest scheduling–transportation planning models. *Can. J. For. Res.* **25**: 1618–1626.
- Yoshimoto, A., and Brodie, J.D. 1994. Short- and long-term impacts of spatial restrictions on harvest scheduling with reference to riparian zone planning. *Can. J. For. Res.* **24**: 1617–1628.
- Yoshimoto, A., Brodie, J.D., and Sessions, J. 1994. A new heuristic to solve spatially constrained long-term harvest scheduling problems. *For. Sci.* **40**: 365–396.