Scheduling Spatial Arrangement and Harvest Simultaneously

Paul C. Van Deusen

Van Deusen, P.C. 2001. Scheduling spatial arrangement and harvest simultaneously. Silva Fennica 35(1): 85–92.

A method based on the Metropolis algorithm is developed for creating desirable spatial configurations on the landscape while simultaneously dealing with other objectives commonly associated with harvest scheduling. Spatial configurations are loosely specified and stochastically attained, which contrasts with other adjacency constraints based on specific block size limits. This method makes it possible to improve habitat and connectivity, and to create buffer zones as part of the scheduling process. It also works with a mapped set of polygons/forest stands and does not require a gridded system.

Keywords Metropolis algorithm, simulated annealing, adjacency constraints
Author's address NCASIS, Northeast Regional Center, 600 Suffolk Street, Fifth Floor, Lowell, MA 01854, USA **E-mail** pvandeusen@ncasi.org
Received 5 September 2000 **Accepted** 9 January 2001

1 Introduction

There is an ever increasing need to meet objectives other than timber production in forest ecosystem management. These new objectives often involve spatial considerations, such as limiting clearcut block size. For example, the major forest-industry landowners in the USA have agreed to abide by a set of voluntary sustainable forestry initiatives (SFI). Companies that don't follow SFI guidelines can no longer be members of the American Forest and Paper Association (AF&PA 1994). Among other things, SFI calls for controlling clearcut size and improving wildlife habitat. Individual States or municipalities may add more regulations. For example, the State of Maine requires a management plan signed by a professional forester for clearcuts larger than 20 acres.

Clearcut-block size constraints are referred to as adjacency constraints in the harvest scheduling literature and numerous algorithms have been presented to deal with this issue. Integer or mixed integer programming can be used for small problems (Meneghin et al. 1988, Torres-Rojo and Brodie 1990, Jones et al. 1991, Yoshimoto and Brodie 1994, Snyder and Revelle 1996). Carter et al. (1997) dealt with larger problems by using multi-year periods with spatial unit based adjacency (Murray 1999), which would not satisfy the SFI definition of adjacency.

There are a number of heuristic approaches that have been used to control blocksizes. Tabu search has been used successfully (Glover and Laguna 1993, Murray and Church 1995, Bettinger et al. 1997, 1998, and Richards and Gunn 2000). Simulated annealing (SA) is another approach that can handle adjacency constraints (Lockwood and Moore 1993, Van Deusen 1996, 1999, Tarp and Helles 1997). Genetic algorithms can also be formulated to solve (Mullen 1996) this problem as can dynamic programming (Hoganson and Borges 1998, Borges et al. 1999).

Mathematical programming techniques have been used to spatially manipulate a managed landscape to improve wildlife habitat (Hof and Raphael 1993, Hof and Joyce 1993, and Hof et al. 1994). However, these methods are limited to small problems because of the huge number of additional constraints that must be generated to describe the desirable spatial configurations. A method is presented here for handling less specific spatial objectives that can work within an SA algorithm. The spatial abjectives are expressed in terms of the spatial juxtaposition of management regimes. This allows for the possibility of stochastically creating desirable habitat or increasing the connectivity within the landscape. For example, one can create a dummy regime that is applied to all ponds to serve as a pond identifier. Suppose all forest stands have a do-nothing regime as one of the suite of regimes that are allowed for each stand. Now the ability to specify that do-nothing regimes should be adjacent to pond regimes would cause the scheduler to place buffer strips around ponds. In the developments that follow, it will become clear that more complicated spatial objectives than buffer strips can also be achieved with the methods developed here.

The new spatial capability is achieved by adding an objective function component to the heuristic scheduling algorithm described in Van Deusen (1999). This algorithm is briefly reviewed (section 2) and the new spatial objective function component is developed (section 3). Simulated data are used to demonstrate (section 4) the algorithm's overall capability to schedule harvest and create desirable spatial patterns on the landscape.

2 The Scheduling Algorithm

The algorithm presented in Van Deusen (1999) is briefly reviewed, since the new spatial capabilities developed here represent an extension of that algorithm. The algorithm operates on polygons, which can be forest stands, ponds, stream reaches, or any other spatial entity. The algorithm assigns a regime to each polygon to create a management schedule. A regime is a list of years where a management action and/or output occurs. The output could be a volume of wood, a cost, a present net value, or acres of habitat. The user of this algorithm is interested in obtaining management schedules that meet a number of objectives and are not too far from optimal. For some users, optimality may be defined by maximizing present net value (PNV), whereas other users may want to maximize a particular kind of habitat.

The algorithm evolves solutions by iteratively seeking to minimize an objective function. The value of the objective function at iteration r is

$$E\left(X^r\right) = \sum_{j=1}^{J} w_j^{r-1} C_j\left(X^r\right) \tag{1}$$

where X^r represents the management schedule at iteration r, w_j^r is a weight determined from the iteration r schedule, and $C_j(X^r)$ is the jth objective function component evaluated at the rth schedule. The list of regimes assigned to polygons 1,...,N is contained in the vector $X^r = \{x_1^r, ..., x_N^r\}$. Each objective function component controls different attributes of the schedule. For example, there are flow components to control even-flow, and a spatial model component will be developed below to control spatial juxtaposition of management regimes.

The Metropolis et al. (1953) algorithm is used to generate potential schedules by iteratively attempting to update each polygon sequentially with a new proposal regime. If the proposal regime improves the overall schedule, it is accepted. Otherwise, the proposal regime may be accepted according to a computed probability. This has the effect of preventing the algorithm from being trapped at a local minimum. No attempt is made to force the result to converge to a single optimal solution, which differentiates this approach from SA as presented by Lockwood and Moore (1993).

A unique feature of this algorithm is the manner in which the weights on each component are determined. The weights, w_j^r, control how much influence the associated objective function component has on possible solutions. The appropriate weights depend on the mix of objective function components and the data, so there is no way to analytically determine reasonable values. The algorithm deals with this by letting the user specify lower and upper goal limits that are scaled from 0 to 1. A goal of 1 means that total attainment is desired, whereas a goal of 0 means that no attainment is required for that objective function component. A goal function is built into the algorithm for each component and the attained goal is computed after each Metropolis iteration. The goal function for component j is $g_j(X^r)$, and depends on the current schedule. Weights are adjusted after each iteration as follows:

if $g_j(X^r) > U_j$ then $w_j^r = aw_j^{r-1}$

if $g_j(X^r) < L_j$ then $w_j^r = w_j^{r-1}/a$

where U and L are the user specified upper and lower limits and a is an adjustment factor between 0 and 1. As shown, the weights are decreased when the goal is over-attained and increased when it is under-attained. In between U and L, the algorithm is said to have converged for that component, and no weight adjustment is needed.

Objective function components can control a wide range of schedule characteristics and 3 basic components are developed in Van Deusen (1999). For review purposes, consider the flow component,

$$C_j\left(X^r\right) = \sum_{t=1}^{T}\left(y_t - \hat{y}_t\right)^2 / F$$

where T is the length of the planning period, y_t represents the total output of some good at time t for the iteration r schedule, $\hat{y}_t$ is the target value for y_t, and F is a scaling factor. The flow component has 2 goals that control its objective function weight. The first goal controls the relative size of deviations from the target value, and the second goal controls year-to-year deviations

$$g1(t) = 1 - \min\left(\frac{\left(\left|y_t - \hat{y}_t\right|\right)}{\hat{y}_t + \delta}, 1\right) \qquad t = 1, \ldots, T$$

and

$$g2(t) = 1 - \min\left(\frac{\left(\left|y_t - \hat{y}_{t-1}\right|\right)}{\hat{y}_{t-1} + \delta}, 1\right) \qquad t = 2, \ldots, T$$

where δ is a small positive number to prevent dividing by 0. The weight for this component is adjusted until both of these goals are met to at least the lower limit for each time period. This allows the user to control the flow of any particular good. See Van Deusen (1999) for more details.

3 Spatial Objective Function Component

The scheduling algorithm being used here is based on the Metropolis algorithm, which is closely related to Markov Chain Monte Carlo (MCMC) methods from the statistical literature (Geman and Geman 1984, Besag et al. 1995, Van Deusen 1996). The new spatial component is derived from the pairwise interaction distribution (Besag 1986), which describes Markov random field distributions with particular spatial characteristics,

$$p(X) \propto \exp\left(-\left[\sum_{i=1}^{N} G_i\left(x_i\right) + \sum_{i=1}^{N}\sum_{i \neq j} G_{ij}\left(x_i, x_j\right)\right]\right) \qquad (2)$$

where X represents the list of assigned management regimes, $G_{ij} = 0$ if i and j are not neighbors, and otherwise the G-functions are arbitrary.

It is helpful to consider a conditional form of $p(X)$, since the solution algorithm involves making comparisons between the current schedule, X, and an alternative, Z, that differs only by the assignment to one polygon. A simple, but effective conditional form of distribution (2) is,

$$p\left(x_i = k \middle| x_{\delta i}\right) \propto \exp\left(const + \sum_h \beta_{hk} n_i(h)\right) \qquad (3)$$

where $x_{\delta i}$ represents the regimes assigned to neighbors of polygon i, $const$ is an arbitrary constant, $\beta_{kh} = \beta_{hk}$, and $n_i(h)$ is the number of neighbors assigned regime h. The β parameters control local spatial arrangement. When $\beta_{hk} < 0$, regimes h and k are encouraged to be close together and if $\beta_{hk} > 0$ they will tend to be separated. The magnitude of these parameters determines the influence of the local spatial component. Thus, this opens up the possibility of model-based scheduling to stochastically create a desired habitat.

The new objective function component is derived from the conditional distribution given by equation (3). The Metropolis algorithm uses the difference between the current schedule, X, and the proposal, Z, where polygon i has a proposed regime change. The difference form of the objective function component is

$$\Delta C_j(i) = \sum_h (\beta_{hx} - \beta_{hz}) n_i(h) \tag{4}$$

where x denotes the current regime of polygon i and z denotes the proposal regime. The summation is over all regimes to allow the user to specify that a regime should be close or apart from itself as well as to other regimes.

The suggested goal function for this component is based on counting the number of times that $\Delta C_j(i) > 0$ for $i = 1,...,N$. A positive value for $\Delta C_j(i)$ means that the proposal regime, z, is spatially more desirable than the current regime, x. The goal function is

$$g_j = 1 - \frac{\sum_{i=1}^{N} I(\Delta C_j(i) > 0)}{N}$$

where $I(.)$ is an indicator function. In this case, g_j gives the proportion of polygons for which the current regime was better than the proposal. Now the user sets upper and lower limits, U_j and L_j, which causes the weight to be adjusted until the proportion of spatially desirable polygons falls within the limit. The actual proportion indicates the number of polygons for which the current schedule is spatially better than a randomly chosen schedule.

4 Example Application

A simple but realistic example application would be to attempt to locate certain harvesting activities away from water bodies. Therefore, a simulated data set is produced to test the capabilities of the new spatial model component. A 40 × 40 grid of cells is generated and each cell is randomly assigned to a class ranging from 1–10 with each class having equal probability of occurring, and then approximately 2.5 percent of the cells are randomly assigned to a 'pond' class. Neighboring cells with the same class assignment are combined into the same polygon, which results in a total of 1090 polygons.

Each of the non-pond polygons is considered to be forested and of an age proportional to its class assignment. Forest polygons are incremented to the next class after each time period. Regimes are then created to allow each forest polygon to be cut anytime it is at class 4 or greater. The planning horizon is of length 10; so a polygon beginning at class 4 could be cut at times 1, 5, and 9, for example. Each forest polygon is also assigned a do-nothing regime as an option, and each pond polygon has a pond regime as its only management option. This leads to a total of 51 regimes.

Now the spatial component can be demonstrated by looking at the results of biasing do-nothing regimes to be near pond regimes. A schedule is first developed by controlling only even-flow of volume. The spatial aspects of the schedule are not controlled. The results are displayed (Fig. 1) with stands under some management in white, stands assigned do-nothing regimes are gray, and ponds are black. The spatial component specified by equation (4) is then added to encourage do-nothing regimes to be close to pond regimes. Specifically, the relevant spatial parameter, $\beta_{\text{pond,do-nothing}}$, is set to –1 with all other spatial parameters set to 0. The upper and lower goal limits are set to 0.99 and 0.94. This has some impact on the spatial juxtaposition (Fig. 2) of ponds and do-nothing regimes. Notice that only 1 pond has no neighboring do-nothing regime (Fig. 2), whereas 6 ponds previously (Fig. 1) had no do-nothing neighbors. Neighbors are defined as any polygon that is adjacent to a side or corner of the target polygon. Even-flow is

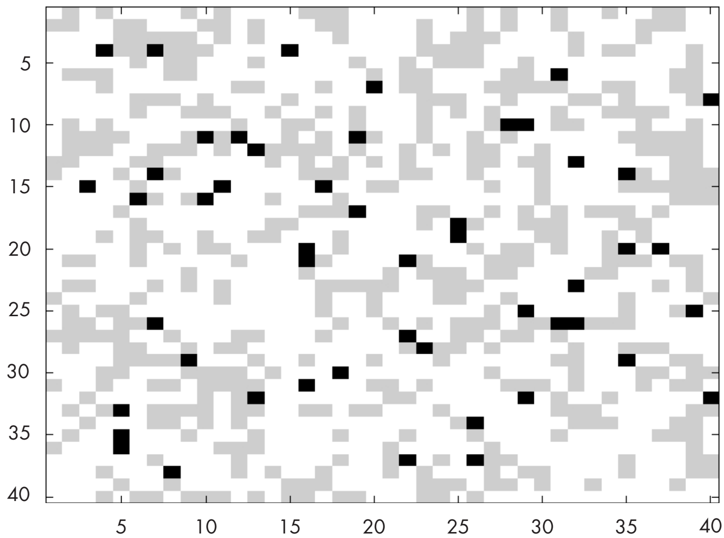

Fig. 1. There are 1090 polygons on this simulated landscape. The black polygons represent ponds and the remaining polygons are forested stands. Forested polygons assigned a do-nothing regime are displayed in gray and the remaining forested polygons are white. The harvest scheduling objective function contains only an even-flow component.

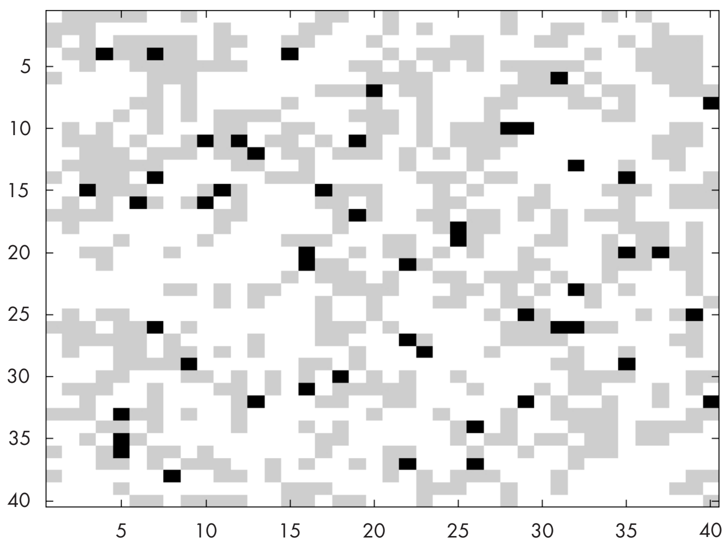

Fig. 2. This is the same as Fig. 1., except there is an even-flow and a spatial model component. The spatial model component requires that some effort be put into placing do-nothing (gray) regimes near ponds (black).

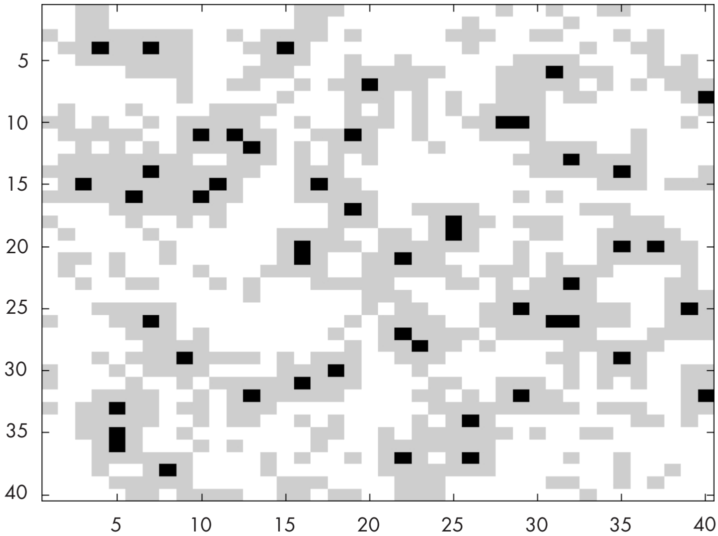

Fig. 3. The same as Fig. 2, but the spatial model component requires more effort on placing do-nothing regimes near ponds.

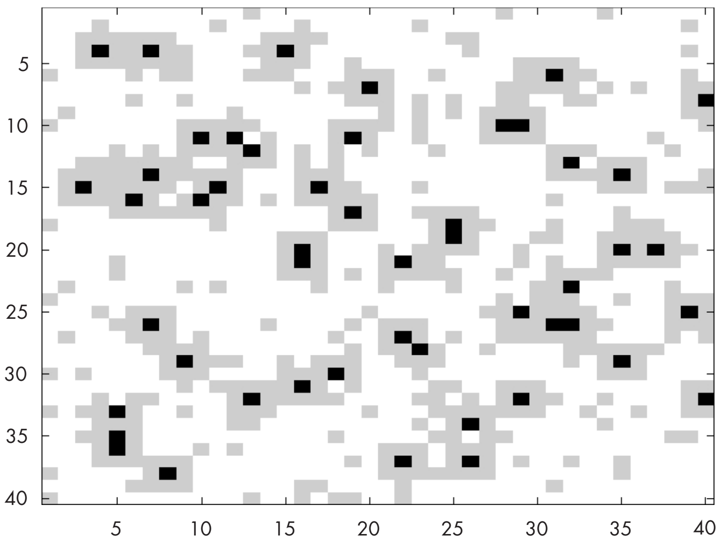

Fig. 4. The same as Fig. 3, but there is a second spatial model component added. The second spatial component puts some effort into keeping do-nothing regimes apart from other do-nothings.

still controlled and the weight on the spatial component is adaptively increased until the lower goal limit is attained.

In order to put more weight on the spatial component, the upper and lower goals are set to 1.0 so that maximum effort will be put into keeping ponds adjacent to do-nothing regimes. This results in a very distinct spatial pattern where ponds are almost completely surrounded by do-nothing regimes (Fig. 3). This has effectively created buffer strips around the ponds along with a number of randomly located corridors.

An additional modification is added in the form of a second spatial component to keep do-nothing regimes apart from other do-nothing regimes, so $\beta_{\text{do-nothing,do-nothing}}$, is set to 1. This conflicts with the first spatial component, but the upper and lower limits on the second component's goal function are set to 0.75 and 0.80 to result in less weight being put on this component relative to the first spatial component. The influence of the second spatial component causes the do-nothing buffers around the ponds to be somewhat smaller (Fig. 4) and clusters of do-nothing polygons that are not adjacent to ponds are smaller. There is also less overall connectivity in the simulated landscape.

5 Discussion and Conclusions

A method has been presented that allows a forest manager to stochastically generate desired spatial patterns on the landscape as part of the harvest scheduling process. An example application focused on putting do-nothing regimes near ponds, but much more demanding spatial objectives could be tackled. The example demonstrates that the method makes it possible to overlay sophisticated spatial goals on the harvest scheduling process.

A knowledgeable user can employ the spatial model component developed here to create desirable spatial configurations of habitat by considering the vegetation that will result from specific management regimes, and then biasing certain regimes to be adjacent or apart. A regime describes outputs and actions that take place over the entire planning horizon, so the spatial characteristics of the landscape being created are not static. Obtaining the desired results may require some iteration, but there is no doubt that the spatial model component allows one to manipulate the spatial characteristics of a managed forested landscape and therefore the desirability of the habitat.

References

Besag, J. 1986. On the statistical analysis of dirty pictures. Journal of the Royal Statistical Society, Series B 48(3): 259–302.

— , Green, P., Higdon, D., & Mengersen, K. 1995. Bayesian computation and stochastic systems. Statistical Science 10(1): 3–66.

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

— , Sessions, J. & Johnson, K.N. 1998. Ensuring the compatibility of aquatic and commodity production goals in eastern Oregon with a tabu search procedure. Forest Science 44(1): 96–112.

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

Carter, D., Vogiatzis, M., Moss, C. & Arvanitas, L. 1997. Ecosystem management or infeasible guidelines? Implications of adjacency restrictions for wildlife habitat and timber production. Canadian Journal of Forest Research 27: 1302–1310.

Geman, S. & Geman, D. 1984. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6: 721–741.

Glover, F. & Laguna, M. 1993. Tabu search. In: Reeves, C. (ed.). Modern heuristic techniques for combinatorial problems. Halsted Press, New York.

Hof, G.H. & Raphael, M.G. 1993. Some mathematical programming approaches for optimizing timber age-class distributions to meet multispecies wildlife population objectives. Canadian Journal of Forest Research 23: 828–834.

— & Joyce, L.A. 1993. A mixed integer linear programming approach for spatially optimizing wildlife and timber in managed forest ecosystems. Forest Science 39(4): 816–834.

Hof, J., Bevers, M., Joyce, L. & Kent, B. 1994. An integer programming approach for spatially and temporally optimizing wildlife populations. Forest Science 40(1): 177–191.

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

Jones, J.G., Meneghin, B.J. & Kirby, M.W. 1991. Formulating adjacency constraints in linear optimization models for scheduling projects in tactical planning. Forest Science 37: 1283–1297.

Lockwood, C. & Moore, T. 1993. Harvest scheduling with spatial constraints: a simulated annealing approach. Canadian Journal of Forest Research 23: 468–478.

Meneghin, B.J., Kirby, M.W. & Jones, J.G. 1988. An algorithm for writing adjacency constraints efficiently in linear programming models. In: Kent, B.M. & Davis, L. (tech coords.). Proc., the 1988 Symp on Systems Analysis in Forest Resources. USDA Forest Service, General Technical Report RM-161. p. 46–53

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. & Teller, E. 1953. Equation of state calculations by fast computing machines. Journal of Chemical Physics 21: 1087–1091.

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

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

Mullen, D.S. 1997. A comparison of genetic algorithms and Monte Carlo integer programming for optimization of adjacency constrainted harvest scheduling problems. Master of Science thesis. Jacksonville, Florida. University of North Florida. Department of Computer and Information Sciences.

Richards, E.W. & Gunn, E.A. 2000. A model and tabu search method to optimize stand harvest and road construction schedules. Forest Science 46:188–203.

Snyder, S. & ReVelle, C. 1996. Temporal and spatial harvesting of irregular systems of parcels. Canadian Journal of Forest Research 26: 1079–1088.

Tarp, P. & Helles, F. 1997. Spatial optimization by simulated annealing and linear programming. Scandinavian Journal of Forest Research 12: 390–402.

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

Van Deusen, P. 1996. Habitat and harvest scheduling using bayesian statistical concepts. Canadian Journal of Forest Research 26: 1375–1383.

— 1999. Multiple solution harvest scheduling. Silva Fennica 33(3): 207–216.

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

Total of 25 references