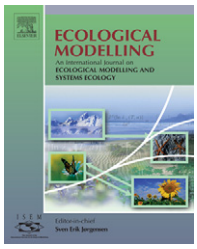
available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/ecolmodel



A selection harvesting algorithm for use in spatially explicit individual-based forest simulation models

*Ken Arii**, *John P. Caspersen*, *Trevor A. Jones*[1], *Sean C. Thomas*

Faculty of Forestry, University of Toronto, 33 Willcocks Street, Toronto, ON, Canada M5S 3B3

ARTICLE INFO

Article history:
Received 5 July 2006
Received in revised form
25 July 2007
Accepted 7 September 2007
Published on line 29 October 2007

Keywords:
Harvesting algorithm
Selection harvesting
Gaps
Forest management
Tolerant hardwoods
Spatially explicit individual-based simulation model

ABSTRACT

There is growing interest in using spatially explicit, individual-based forest simulation models to explore the ecological and silvicultural consequences of various harvesting regimes. However, simulating the dynamics of managed forests requires harvesting algorithms capable of accurately mimicking the harvest regimes of interest. Under selection silviculture, trees are harvested individually or in small groups, with the aim of retaining trees across a full range of size classes. An algorithm that reproduces selection harvesting must therefore be able to recreate both the spatial and the structural patterns of harvest. Here we introduce a selection harvest algorithm that simulates harvests as a contagious spatial process in which the cutting of one tree affects the probability that neighboring trees are also cut. Three simple and intuitive parameters are required to implement this process: (1) the probability of cutting a "target" tree (P_t) (often a function of tree size), (2) the probability of cutting its nearest neighbor (P_n), and (3) the total number of target trees to cut (N_t). Specification of these parameters allows representation of both the spatial and the structural patterns of harvest expected under selection silviculture.

Based on this simple process, we built two different versions of the harvesting algorithm. An "empirical" algorithm was designed and calibrated to reproduce the observed spatial and size distribution of stumps (harvested trees) at a study site in central Ontario, and was successful in reproducing harvesting patterns found in the field, notably variability in the cluster size of harvested trees. The "user-defined" algorithm implements alternative harvesting regimes (user-defined harvest targets), which may differ in the intensity of harvesting, the size-specificity of harvesting, and the spatial pattern of harvesting. We show that the user-defined harvesting algorithm succeeds in meeting harvest targets specified by the user (e.g., size class distribution and basal area of trees harvested), while simultaneously adjusting the gap size specified (i.e., the distribution of harvested trees per cluster). Incorporation of this harvesting algorithm into spatially explicit, individual-based models will permit analyses of long-term responses of forest stands to harvesting scenarios that more realistically capture the complex patterns of within-stand variability generated by selection silviculture as practiced in actual managed forests.

* *Corresponding author*. Current address: Health, Environment and Life Sciences Institute, College of Asia Pacific Studies, Ritsumeikan Asia Pacific University, 1-1 Jumonjibaru, Beppu, Oita 874-8577, Japan. Tel.: +81 977 78 1271; fax: +81 977 78 1271.
E-mail address: ken.arii@mail.mcgill.ca (K. Arii).
[1] Current address: Department of Geography, University of British Columbia, 1984 West Mall, Vancouver, BC, Canada V6T 1Z2.
0304-3800/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.ecolmodel.2007.09.007

1. Introduction

Individual-based models have long been used to simulate natural forest dynamics because canopy gaps formed by the death of individual trees play an important role in maintaining species diversity and productivity (Botkin et al., 1972; Bormann and Likens, 1979; Shugart, 1984; DeAngelis and Gross, 1992; Pacala and Deutschman, 1995). For example, individual canopy gaps allow tree species that are only moderately tolerant of shade to coexist with species that cast and tolerate deep shade, provided that the less tolerant species are better able to colonize and/or capture single-tree gaps (Tilman, 1994; Pacala and Rees, 1998). Thus, models that can resolve individual gaps are able to simulate gap-phase regeneration, and thereby predict the observed persistence of mid-tolerant species in old-growth stands (Pacala et al., 1996). The ability to resolve individual gaps is also advantageous for simulating processes that depend on gap size, such as the process of gap-size partitioning, whereby shade-intolerant trees can coexist with more tolerant species if they are better able to colonize and/or capture large gaps formed by natural disturbances (Busing and White, 1997; Busing and Mailly, 2004).

Over the last decade, there has been a growing effort to develop individual-based models that can simulate dynamics of managed forests that are subject to periodic partial harvests (e.g., Pausas et al., 1997; Coates et al., 2003; Pennanen et al., 2004). While much effort on this topic has focused on model parameterization and analysis of long-term impacts, an additional critical aspect is the development of harvesting algorithms that accurately mimic the structural patterns of partial harvesting regimes (Söderbergh and Ledermann, 2003). For example, several existing harvesting algorithms allow the user to specify standard harvest targets: either the amount to be harvested from each diameter class, or the amount to be retained in each diameter class (e.g., Solomon et al., 1995; McCarter et al., 1998; Gustafson et al., 2000; Huth and Ditzer, 2001). Few algorithms, however, allow the user to simultaneously modify the spatial pattern of harvesting to attain a particular gap size distribution (Coates et al., 2003), despite the fact that the gap size distribution is one of the defining traits of a forest disturbance regime (Seymour and Hunter, 1999). Furthermore, harvesting algorithms developed to date have been rarely evaluated against field data to determine how well they reproduce observed harvesting patterns. Clearly, harvesting algorithms that can accurately reproduce both the structural and spatial pattern of harvesting must be developed and tested in order to have confidence in simulations of gap dynamics in managed forests.

Selection silviculture is practiced throughout eastern North America, in parts of western North America, and also in Europe (O'Hara, 2002; Pommerening and Murphy, 2004). Under standard selection silviculture, trees are harvested individually, or in small groups, with the aim of retaining trees across a full range of size classes (Nyland, 1998; OMNR, 1998). In eastern North America, the typical selection harvesting regime removes one third of the basal area every 20 years, and the size distribution of residual trees typically is approximated by a negative exponential distribution (many stems in small diameter classes and few stems in larger diameter classes). In southern and central Ontario, the standard harvest regime leaves residual trees that conform to a modified Arbogast distribution (Arbogast, 1957, see OMNR, 2004 for more details).

Despite these standard procedures, harvesting regimes do vary from one management unit to the next, depending on the structure and the composition of the forest, and the objectives of the manager (Hansen and Nyland, 1987; OMNR, 1998). For example, a manager may modify the number of trees to be harvested, or their distribution across size classes, to optimize timber yields. Managers may also modify the spatial pattern of harvesting to encourage the regeneration of tree species that are less tolerant of shade. For example, group selection in southern Ontario calls for creation of multi-tree gaps (as large as 0.2 ha) that are interspersed among single-tree gaps (OMNR, 2004). In addition, logistical considerations, in particular physical impacts related to the felling and skidding of harvested timber (and minimization of such impacts), very commonly result in multi-tree gaps. Thus, even in cases where the objective is single-tree selection, gaps of various sizes are inevitably created during harvesting operations.

In this paper, we describe and test a harvesting algorithm designed to simulate selection silviculture as practiced in the tolerant hardwood forests of central Ontario, Canada. There were three objectives that guided the design of our harvesting algorithm. The first objective was to mimic the harvesting regime employed at our primary field site, the Haliburton Forest and Wildlife Reserve. In particular, we designed the algorithm so that it could reproduce the observed spatial distribution of harvested trees (hereafter "stumps") and their observed diameter distribution. The second objective was to design the algorithm so that it could also simulate alternative harvesting regimes, including regimes that may differ in the intensity of harvesting, the size-specificity of harvesting, and the spatial pattern of harvesting. The third objective was to design the algorithm so that it would be simple to use (i.e., has just a few intuitive parameters) and could be implemented in any spatially explicit individual-based forest simulator, including the most recent version of SORTIE-ND (http://www.sortie-nd.org/index.html). To achieve these objectives, we developed two different versions of the harvesting algorithm: the first is called the "empirical" version since it was designed to reproduce the patterns found in field data; the second is called the "user-defined" version to indicate that the user can implement alternative harvesting regimes in SORTIE, or any other spatially explicit individual-based model where each individual (or a tree) within the model has its own unique spatial coordinate within a simulated stand (e.g., Pretzsch et al., 2002; Parrott and Kok, 2001; Nuttle and Haefner, 2007). However, we shall refer to both versions as "nearest neighbor" algorithms to emphasize the underlying process used to simulate harvesting, and to distinguish them from other spatially explicit harvesting algorithms (Söderbergh and Ledermann, 2003).

2. The nearest neighbor algorithm

Both the empirical and user-defined versions simulate harvesting as a contagious spatial process in which the cutting of one tree affects the probability that neighboring trees are

also cut. The algorithm is thus similar in some respects to the approaches used to simulate the spread of fire in forests (Drossel et al., 1993; Malamud et al., 1998). Here, we provide a brief description of the nearest neighbor algorithm, focusing on the contagious process and the parameters that govern the spatial distribution of stumps in the simulation. In subsequent sections, we will describe the algorithm in more detail, including the parameters that dictate the size distribution of stumps, and the methods used to calculate parameter values from field data.

At least three parameters are required to run the nearest neighbor algorithm: (1) the probability of cutting a "target" tree (P_t), (2) the probability of cutting its nearest neighbor (P_n), and (3) the total number of target trees to cut (N_t). Once these parameters are specified, the nearest neighbor algorithm begins by searching for the largest tree in the stand and then by cutting the tree with the probability of P_t. If the target tree is cut, the algorithm then searches for the nearest neighbor of the stump, and draws another random number between 0 and 1; if the random number is greater than P_n, the nearest neighbor remains standing, in which case we refer to the original stump as a "singleton"; if the random number is less than or equal to P_n, the algorithm cuts the nearest neighbor as well, in which case we refer to the two stumps as a "pair", indicating that at least one of the stumps is the other stump's nearest neighbor.

Once the nearest neighbor is also cut, this contagious process may continue to propagate: the algorithm searches for the nearest neighbor of the second stump, and cuts the nearest neighbor with the probability P_n. In this way, harvesting spreads from one tree to the next, until a random number exceeds P_n, at which point the contagious process terminates. Hereafter, we will refer to the resulting stumps as a "group", which includes the nearest neighbor of all its member stumps, with the possible exception of the last stump to be cut. Finally, we will refer to all stump categories (singletons, pairs, and groups) as "clusters", and we will use the terms cluster and gap interchangeably.

Once the contagious process has terminated, the algorithm searches for the next largest live tree in the stand, then cuts this second target tree with the probability P_t, and thereby reinitiating the contagious process described above. The algorithm continues to cut the target trees (proceeding from the largest to the smallest trees in a stand), and their neighbors, until it has cut N_t target trees.

For the empirical version of the algorithm, all three parameter values are calculated directly from field data, but in the user-defined version they are specified by the model user.

2.1. Gap size distribution

There are three important features of the nearest neighbor algorithm: (1) it creates gaps (clusters) of various sizes, (2) the size distribution of gaps can be modified by changing the value of P_n, and (3) the theoretical size distribution of gaps can be calculated from the value of P_n: the probability of a gap containing x stumps is $(1 - P_n)P_n^{(x-1)}$. For most values of P_n, the distribution of gap sizes approximates a negative exponential distribution. However, the algorithm does afford some flexibility in specifying the shape of the gap size distribution: the distribution is rather flat when $P_n > 0.9$, all stumps will be singletons when $P_n = 0$, and all trees will be cut when $P_n = 1$.

2.2. When the nearest neighbor is already a stump

As a cluster grows, it is possible that it will "run into" another cluster, as happens when the nearest neighbor of a recently cut stump is already a stump. At this point, the algorithm would normally cut the nearest neighbor with a probability of P_n, and the contagious process would continue to spread. In the event of running into another cluster, we considered terminating the contagious process and preventing the current cluster from growing any further. However, this might limit the size of clusters, and thereby truncate the tail of the gap size distribution. Therefore we decided to terminate or grow the current cluster, depending on the value of the random number: if the random number is less than P_n the cluster is allowed to grow, otherwise the nearest neighbor is "resurrected" (its status is reassigned to live), and the cluster growth is terminated. In addition, to reduce the probability that clusters run into one another, we modified the algorithm so that it does not select target trees whose nearest neighbor is already a stump. As a result, target trees are more spatially dispersed than if chosen at random.

2.3. Other factors limiting gap size

In practice, the gap size distribution of a simulated harvest is expected to deviate somewhat from the theoretical distribution given above for several reasons. First, the contagious harvesting process may be interrupted if the simulation domain is too small, or the simulation domain is divided into grids beyond which the process cannot spread, in which case the tail of the size distribution will be truncated. Second, the contagious harvesting process may also be interrupted whenever both trees in an adjacent pair are each other's nearest neighbor, because contagious harvesting cannot spread beyond such a pair (see Section 3.7.1 for more detail on reciprocal tree pairs). In this case, the tail of the size distribution will also be truncated. Finally, since the algorithm is stochastic, the realized size distribution may differ from expected size distribution by chance alone, particularly when the number of target trees to be harvested (N_t) is small.

3. The empirical algorithm

3.1. Study site and data collection

The field data used for the empirical algorithm were collected in Haliburton Forest and Wildlife Reserve (45°15′N, 78°34′W) as part of a related study on tree growth (Jones, 2006). As is typical of the Great Lakes—St. Lawrence region (Rowe, 1972), the upland hardwood forests of Haliburton Forest are dominated by sugar maple (*Acer saccharum* Marsh.), American beech (*Fagus grandifolia* Ehrh.), eastern hemlock (*Tsuga canadensis* (L.) Carrière), and yellow birch (*Betula alleghaniensis* Britt.). The forest has been managed under selection silviculture for the past 40 years, and was selectively harvested for yellow birch and white pine (*Pinus strobus* L.) prior to that time. The spatial

extent and harvest date of each cut-block has been recorded for the last 20 years.

Here we used 11 cut-blocks representing 6 harvest years (1994, 1997, 1998, 2001, 2002 and 2003) for this study. In the summer of 2004, a network of plots was established within each cut-block using a systematic sampling protocol. The plot locations were selected by walking down primary skid trails and stopping every 100 m to place one plot on each side of the skid trail, at a distance of 50 m from the skid trail. The plots were 20 m in radius, and every stem (tree and stump) greater than 8 cm DBH within the plot was mapped (only trees above 10 cm DBH was used for this study). For each live tree, DBH was measured at 1.3 m. For stumps, diameter was measured at the highest point, and was used to approximate DBH at 1.3 m (when the DBH was rounded to nearest centimeter, there were no significant difference when between the two values). A total of 111 plots and 5477 stems were mapped, covering a combined area of approximately 14 ha. Of these, 75% of the plots were used for model calibration (84 plots), and the remaining 25% was used for model validation (27 plots).

While the basal area of stumps accurately reflects the amount of wood harvested in a particular year, the basal area of live trees is somewhat greater than the standing crop at the time of harvest, since growth has occurred since the time of harvest. Thus, this may introduce some bias when comparing the basal area of live trees against that of stumps. However, given that the average time since harvest was only approximately 3.9 years, we simply assumed that the basal area of live trees provides an adequate measure of the basal area at the time of harvest.

3.2. *Calibration of the empirical algorithm*

The empirical version of the algorithm was designed to reproduce both the diameter distribution of stumps and the spatial distribution of stumps. To reproduce the size distribution, we allowed the probability of cutting a target tree (P_t) to depend on the diameter of the target tree at breast height (DBH), and used five discrete diameter classes that are commonly used to specify harvesting targets in southern Ontario (OMNR, 2000): polewood (10 cm ≤ DBH < 26 cm), small sawlogs (26 cm ≤ DBH < 38 cm), medium sawlogs (38 cm ≤ DBH < 50 cm), large sawlogs (50 cm ≤ DBH < 62 cm), and extra-large sawlogs (DBH ≥ 62 cm). Thus, the probability of cutting down a target tree is specified by five parameters: $P_{t(pole)}$, $P_{t(small)}$, $P_{t(med)}$, $P_{t(large)}$, $P_{t(x\text{-}large)}$. The values of these parameters were obtained from the field data by simply calculating the proportion of stumps in each diameter class. Hereafter we will use the hat symbol (∧) to denote parameter values that are calculated from the calibration dataset (list of symbols used in this article is provided in Appendix A).

The value of this parameter was obtained from the field data by calculating the proportion of stumps whose nearest neighbor is also a stump ($\hat{P}_n$). For example, if there were 100 stumps in the entire calibration dataset, and 20 of these had a stump as their nearest neighbor, then $\hat{P}_n$ would be equal to 20/100, or 0.2.

The value of the parameter N_t was obtained by calculating the number of clusters in each plot ($\hat{N}_{t(plot)}$), and the total number of clusters ($\hat{N}_t$) in the entire calibration dataset. As described in Section 2, a cluster may consist of a singleton, a pair, or a group of stumps, each of which is defined by the number of nearest neighbors included in the cluster. Thus, it is possible to divide stumps into clusters based on the status (live or stump) of their nearest neighbor. This clustering method is most readily explained using terminology from graph theory (Bollobás, 2002) as shown in Appendix B.

3.3. *Testing whether the spatial distribution of stumps is non-random*

To test whether stumps are non-randomly distributed, we compared the proportion of stumps whose nearest neighbor is a stump ($\hat{P}_n$) to the proportion that would be expected if stumps were randomly distributed ($\hat{P}_{n(rand)}$). The proportion expected at random was calculated by first randomly reassigning the status of each stem while maintaining the same number of stumps in each plot, then recalculating the proportion of stumps whose nearest neighbor is a stump ($\hat{P}_{n(rand)}$). This randomization procedure was repeated 10,000 times, so that the mean value of $\hat{P}_{n(rand)}$ could be calculated and compared to the observed proportion $\hat{P}_n$. We plotted $\hat{P}_n$ on the frequency distribution of $\hat{P}_{n(rand)}$ to determine probability of obtaining the observed value of $\hat{P}_n$ by chance alone.

3.4. *Simulating harvest using field data as input*

The empirical version of the algorithm can run on any tree map that includes the diameter and spatial coordinates of each tree, whether the tree map is produced by an individual-based forest simulator, or obtained from mapped plots. Here, we summarize the four steps taken by the algorithm to simulate a harvest using data from mapped plots as input: (1) the algorithm reads the tree map into memory. (2) All trees in the tree map are assigned a "live tree" status, but size and the spatial coordinates of trees are retained. (3) The algorithm searches for the largest tree in the plot and initiates the nearest neighbor algorithm (see Section 2). In the empirical algorithm, the probability of cutting a target tree is $\hat{P}_{t(size)}$, while the probability of cutting the nearest neighbor is P_n. (4) If the number of target trees harvested in the plot equals $\hat{N}_{t(plot)}$, the algorithm moves on to the next plot; if not, the algorithm repeats steps 3 and 4, after choosing the next largest tree in the plot.

Once the harvesting is completed in all the plots, the algorithm terminates and produces simulated tree maps of all the harvested plots. These output tree maps are identical to the input tree maps, except that a different set of trees are designated as stumps. Thus, these two sets of maps can be compared using several harvest indices: (1) the total number of stems harvested across all the plots (ha^{-1}, original: $\hat{Y}_{nm}$, simulated: $\tilde{Y}_{nm}$), and the total number by size class (original: $\hat{Y}_{nm(size)}$, simulated: $\tilde{Y}_{nm(size)}$), (2) the total basal area of stems harvested across all plots (ha^{-1}, original: $\hat{Y}_{ba}$, simulated: $\tilde{Y}_{ba}$), and the total basal area harvested by size class (original: $\hat{Y}_{ba(size)}$, simulated: $\tilde{Y}_{ba(size)}$), and (3) the proportion of stumps whose nearest neighbor is also a stump (original: $\hat{P}_n$, simulated: $\tilde{P}_n$). The first two groups of harvest indices compare the yield of the original and the simulated harvest, while the third examines the spatial pattern.

3.5. *Simulating harvest in an individual-based simulator*

The empirical algorithm can also accept as input tree maps that are divided into grids, as is the case for SORTIE-ND (http://www.sortie-nd.org/index.html). In this case, the harvesting of each grid cell is simulated using the same five steps described above, except that the number of target trees to cut in each grid cell ($\hat{N}_{t(grid)}$) is calculated from $\hat{N}_t$ (total number of clusters in entire calibration dataset) as follows. First, the average density of clusters $\hat{N}_{t/ha}$ in the calibration dataset is calculated by dividing $\hat{N}_t$ by the total area (ha) of plots in the calibration dataset. Second, the cluster density is multiplied by the area of the entire tree map to calculate the total number of target trees to harvest in the simulation. Finally, the total number of target trees to harvest is randomly distributed randomly among the grid cells such that the frequency distribution of $\hat{N}_{t(grid)}$ is Poisson.

3.6. *Validation of the empirical algorithm*

To validate the empirical algorithm, we ran simulated harvests using both the calibration and validation datasets as input tree maps, and compared the simulated harvest patterns to the original harvest patterns using harvest indices (as described in Section 3.4). The simulated harvests were repeated 100 times on each dataset and the harvest indices were calculated for each plot after each replication. The means and the standard deviations of the simulated harvest indices were then calculated and compared to the observed harvest indices.

3.7. *Further details about the harvesting algorithm*

Prior to running the simulation and validating the model, there were some additional concerns that needed to be addressed. In this section, we describe these concerns and additional analyses and adjustments we have made to the algorithm to address them.

3.7.1. *Reciprocal pairs*

In almost any stand of trees there will be pairs of trees in which both trees are the other tree's nearest neighbor. If the harvesting algorithm is implemented in a stand where reciprocal tree pairs are abundant, the spatial pattern of stumps will be less aggregated than was intended because contagious harvesting cannot spread beyond such a pair. In particular, the proportion of stumps whose nearest neighbor is also a stump (simulated value, $\tilde{P}_n$) will be less than the value of the input parameter P_n (the probability of cutting the nearest neighbor of a target tree). To prevent this problem, we adjusted the probability of cutting the nearest neighbor in a reciprocal pair ($P_{n(adj)}$) by setting it equal to $\sqrt{P_n(P_n+1)^{-1}}$, as described in Appendix C. This adjustment ensures $\tilde{P}_n$ approximates the input parameter value $\hat{P}_n$.

3.7.2. *Effects of plot boundaries on the value of* $\hat{P}_n$

The value of $\hat{P}_n$, the proportion of stumps whose nearest neighbor is also a stump, was calculated by searching for nearest neighbors among all the stumps within each plot; those outside of the plot boundary were ignored. Thus, we were concerned the value of $\hat{P}_n$ may have been different had we included trees outside of the plot. To examine any bias that may arise from this calculation method, we compared the values of $\hat{P}_n$ before and after excluding stumps that may have its nearest neighbor outside of the plot. We determined whether or not a stump should be excluded by measuring the distance to the nearest neighbor and by comparing it to the shortest distance to the plot boundary; if the distance to the boundary was less than the distance to the nearest neighbor, the stump was excluded from the calculations (note: these excluded stumps were still used as neighbors of stumps that are included in the calculations). This "edge-corrected" value of $\hat{P}_n$ did not differ substantially from the uncorrected value (as shown in Section 5.1), so we did not employ the edge-correction in any of our other analyses.

4. The user-defined algorithm

4.1. *Deriving parameter values from harvest targets*

The user-defined version of the harvesting algorithm allows one to modify the empirical harvesting regime by altering the parameter values in the previous section. To modify the harvest regime, the user must specify the target amount to be harvested in each diameter class, or the amount to be retained in each diameter class. Furthermore, the user must choose to specify these targets in terms of numbers of stems or basal area. Here, we describe how the parameter values are derived from these targets, assuming that the user has chosen to specify the number of stems to remove from each diameter class. The diameter classes used for this purpose are the same as those defined in Section 3.2, except that the large and extra-large sawlogs were grouped together. The total number of stems to be cut will be denoted as C, and the target numbers for each size class will be referred to as $C_{(pole)}$, $C_{(small)}$, $C_{(med)}$, and $C_{(large)}$.

Once the user defines the harvest targets, the algorithm automatically calculates values for the input parameters $P_{t(size)}$ and N_t. We will refer to the parameter values calculated from the harvest targets as $\tilde{P}_{t(size)}$ and $\tilde{N}_t$. Note that the value of P_n is independent of the harvest targets and can take any value between 0 and 1; this user-defined parameter value will be denoted as $\tilde{P}_n$.

$\tilde{P}_{t(size)}$ is calculated by reading in a input tree map, counting the number of trees in each size class, then dividing each $C_{(size)}$ by the total number of trees found in each class. This provides the proportion of trees that are to be cut in each diameter class.

The parameter N_t specifies the number of target trees to cut (i.e., number of clusters to create) and should be set to an appropriate value that will meet the harvest target, C. Furthermore, the appropriate value of N_t will depend on the size structure and spatial structure of the stand to be harvested. Thus, to solve for N_t, we opted to take an iterative simulation approach in which we repeatedly simulate the harvest of the tree map using various values of N_t, then choose the value of N_t that best satisfies the harvest target in the simulation (see Appendix D for details, note: this iterative approach takes approximately 3 min on a computer with Pentium M (1.5 GHz)

and 1 GB of memory). This procedure ensures that the total number of trees harvested by the algorithm approximates the value specified by the user.

4.2. Evaluation of the user-defined algorithm

Here, we evaluate the user-defined version of the algorithm by addressing the following two questions: (1) does the user-defined algorithm meet the harvest targets ($C_{(size)}$)? (2) Does changing the value of P_n have the expected effect on the spatial aggregation of stumps? To address these questions, we ran two simulations using as input a tree map from an old-growth stand in Haliburton, then compared simulated yields with the harvest targets, and analyzed the degree of spatial aggregation of stumps in the two output tree maps using spatial point pattern analysis.

The input dataset used in these two simulations has been previously described in Caspersen and Saprunoff (2005) and is not related to the calibration and validation datasets used for the empirical algorithm. The size of the plot we used was 200 m × 200 m (4 ha; the original tree map was trimmed), and we set the size of each grid cell at 40 m × 40 m. The basal area in this plot was 29.1 $m^2\,ha^{-1}$ and the number of stems was 515.8 ha^{-1}. In both simulations, the harvest targets ($C_{(size)}$) and the probability of cutting a neighbor tree ($\tilde{P}_n$) were set at the same arbitrary values. However, the probability of harvesting a nearest neighbor $\tilde{P}_n$ was 0.284 in the first simulation and 0.568 in the second simulation (note: 0.284 is the observed value used in empirical harvesting algorithm). It was expected that doubling $\tilde{P}_n$ would significantly increase the degree to which stumps are aggregated.

To assess whether or not the user-defined algorithm can meet the harvest targets, we compared the simulated harvest yields ($\tilde{C}_{(size)}$) to the harvest targets $C_{(size)}$. To assess whether doubling the value of $\tilde{P}_n$ increased stump aggregation, we used Ripley's K function (Baddeley and Turner, 2005, Spatstat) to characterize the spatial pattern of stumps in the two output tree maps, and compared them to the pattern that would be expected if trees were harvested at random. The random harvesting patterns were created by taking the two output tree maps and randomly reassigning the status of trees while preserving the total number of stumps (this was done separately for each size class) to create randomized tree maps. This randomization was repeated 999 times, and the K function was calculated after each iteration. Using the results from this repeated randomization process we determined the critical values of K that would be expected under the random harvest pattern ($\alpha = 0.01$). If the algorithm-generated K function lied outside these critical values, the spatial pattern of stumps was considered to be statistically different from a random distribution.

4.3. Comparing light regimes under different harvest regimes

The user-defined algorithm allows the user to increase the value of P_n to simulate group selection, a practice that is designed to create larger gaps with higher light levels. To determine whether increasing $\tilde{P}_n$ has the expected effect on understory light regimes, we imported five of the tree maps described in Section 4.2 (the input map, the two output maps, and two randomized maps) into SORTIE, and used the GLI Map Creator to calculate an understory light level for every 1 m × 1 m grid. We then compared the frequency distribution of light in each tree map to assess whether aggregation increased understory light levels.

5. Results

5.1. Calibration of the empirical harvesting algorithm

The observed proportion of stumps in each size class increased substantially with tree size ($\hat{P}_{t(pole)} = 0.04$, $\hat{P}_{t(small)} = 0.18$, $\hat{P}_{t(med)} = 0.37$, $\hat{P}_{t(large)} = 0.48$, $\hat{P}_{t(x\text{-}large)} = 0.62$), as did the proportion of basal area cut (Fig. 1). This reflects the preferential harvesting of the larger trees, and is to be expected under selection silviculture. We also examined the proportion of stems cut in each plot and found that harvesting intensity varied substantially from one plot to the next (Fig. 2). The average fraction of stems harvested per plot was 0.17 ± 0.08 (S.D.), whereas the average fraction of basal area harvested per plot was 0.31 ± 0.12 (S.D.).

The number of clusters in a plot varied considerably, as well as the size of each cluster (Fig. 3). Approximately 80% of the clusters found in the Haliburton dataset were singletons (cluster size = 1), and most of the remaining clusters were paired (cluster size = 2) (Fig. 3a). We found no significant correlation between the number of clusters per plot and the average size of clusters within a plot ($p = 0.365$).

The proportion of stumps observed to have a stump as the nearest neighbor ($\hat{P}_n$) was 0.284. When the edge-correction was applied (Section 3.7.2), the value of $\hat{P}_n$ was only marginally lower, 0.276, which suggests that edge correction is not necessary here. Thus, all the values of $\hat{P}_n$ and $\tilde{P}_n$ reported in the following sections were calculated without edge correction.

The proportion expected at random ($\hat{P}_{n(rand)}$) was 0.159, although there was significant variation around this value (Fig. 4). The chances of obtaining a value equal to the observed through random harvesting are extremely small (Fig. 4). Thus we conclude that trees are harvested in a non-random and aggregated manner at Haliburton, and speculate that this is the case with selection silviculture in general.

5.2. Validation of the empirical algorithm

The total simulated yields closely matched the observed yields in both the calibration and the validation datasets (Table 1). The distribution of simulated yield across the diameter classes was also consistent with the observed yields (Fig. 5). The $\tilde{P}_n$ values were also consistent with the empirical values, which implies that the algorithm was able to reproduce the spatial distribution pattern of stumps as captured by the nearest neighbor method (Table 1). However, for the validation dataset, there were some discrepancies between the empirical value $\hat{P}_n$ and the simulated value $\tilde{P}_n$ (Table 1). This may be attributed to the smaller sample size of the validation data set (27 plots), which may have caused the proportion in the validation data ($\hat{P}_n = 0.219$) to deviate from the proportion in the original 84

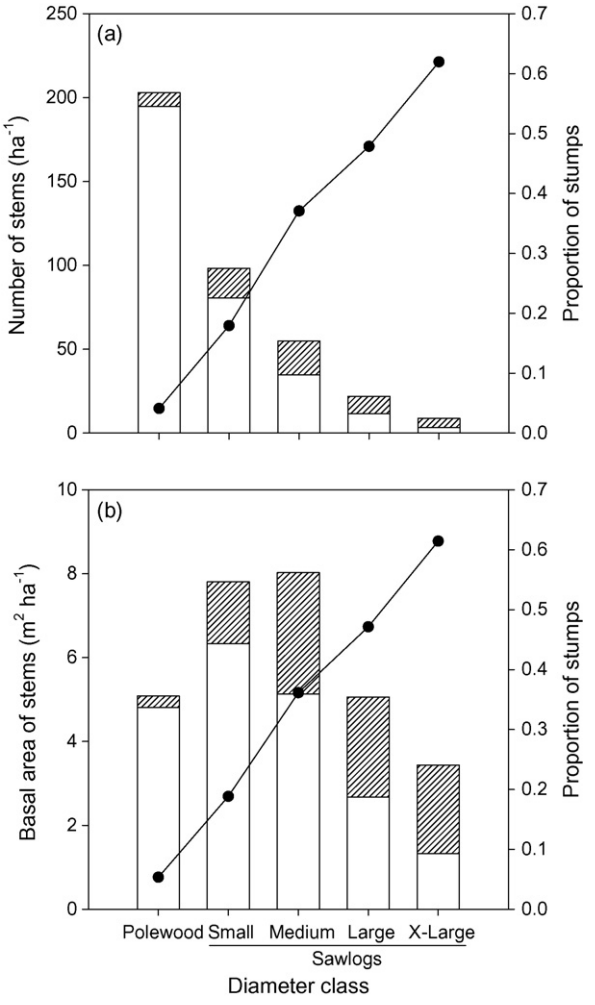


Fig. 1 – (a) The number of stems (includes both live trees and stumps) and (b) the basal area of stems by diameter size class, observed in the 84 plots used for calibration. The shaded area within the bar graph represents the stumps, while the line with dots indicates the proportional value of stumps ($\hat{P}_n$) in each size class. The total number of live trees in the 84 plots was 3426 (average per plot = 40.8 ± 11.5 (S.D.)), while that of stumps was 657 (average per plot = 7.8 ± 3.6). Total basal area of live trees was 20.3 m^2 ha^{-1}, and the total basal area of stumps was 9.1 m^2 ha^{-1}.

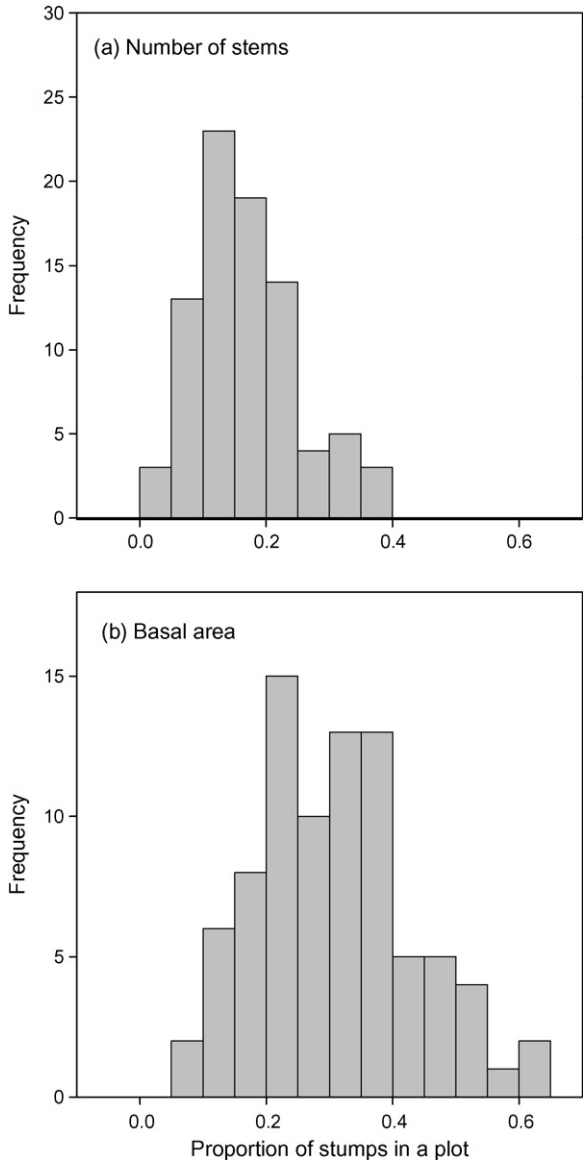


Fig. 2 – Frequency distribution of the proportion, cut calculated based on (a) the number of stems per plot and (b) the basal area per plot.

plots ($\hat{P}_n = 0.284$). Nevertheless, the value of $\hat{P}_n$ in the validation dataset is still within two standard deviations from the mean value of $\tilde{P}_n$ (0.292 ± 0.044). Moreover, the algorithm was quite successful in reproducing the value of the input parameter P_n, which was set at 0.284.

5.3. *Evaluation of the user-defined algorithm*

Overall, the harvesting algorithm was successful in meeting the harvest targets (Table 2). The spatial point pattern analysis of harvested trees using *K* function showed that the pattern of stumps created by the harvesting algorithm was significantly more clustered than what would be expected in random harvesting (Fig. 6). Spatial clustering is significant for distances from 1 to 11 m when the value of $\tilde{P}_n$ is set at 0.284, and when this value is set at 0.568, the clustering is significant for distances from 1 to 22 m. Thus larger clusters were created in the algorithm by increasing the value of $\tilde{P}_n$, as designed. When $\tilde{P}_n$ is set at 0.284, the average cluster size was 1.46 stumps/cluster, and when $\tilde{P}_n$ is set at 0.568, the value increased to 1.70 stumps/cluster (note: average cluster size for a random harvest is 1.20 stumps/cluster).

5.4. *Comparing light regimes under different harvest regimes*

Comparing the distribution of predicted light levels revealed that aggregated harvesting resulted in a more skewed dis-

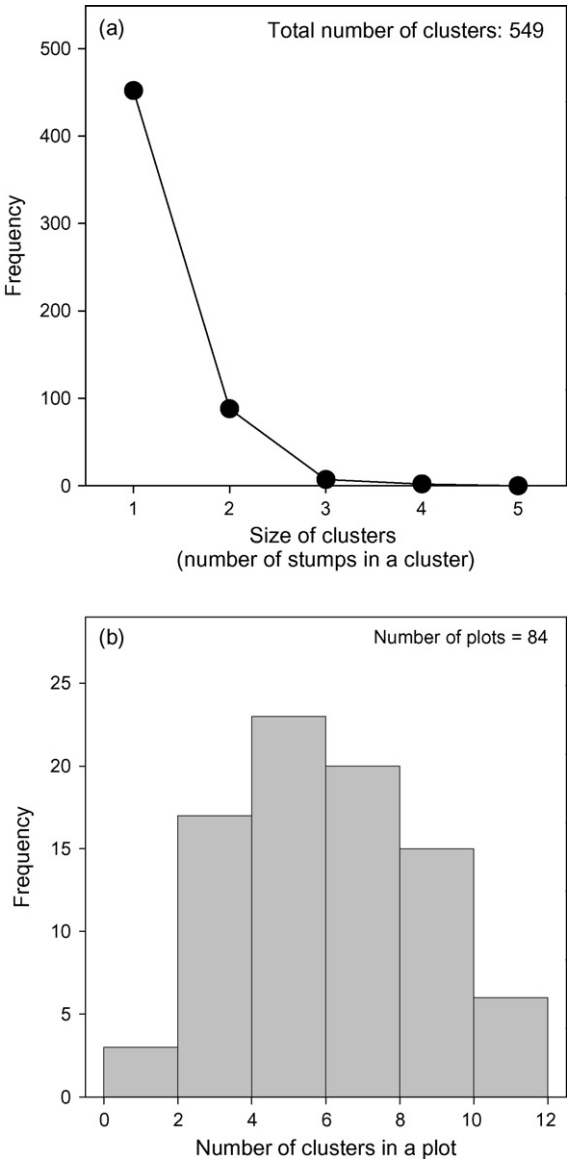


Fig. 3 – (a) Frequency distribution of cluster size found in the Haliburton data set (number of plots = 84, total number of clusters = 549). (b) Frequency distribution of number of clusters per plot.

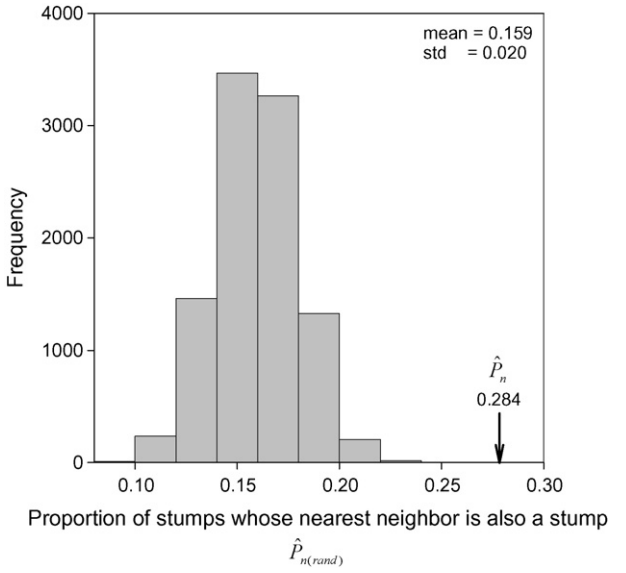


Fig. 4 – The frequency distribution of $\hat{P}_{n(rand)}$, the probability of stump's nearest neighbor being a stump, based on random harvesting (stumps are randomly placed within the plots). The empirical value $\hat{P}_n$ at Haliburton was 0.284, significantly higher than the mean value of $\hat{P}_{n(rand)}$ (0.159).

tribution than random harvesting (Fig. 7). In particular, the distribution under aggregated harvesting had a longer "tail" (Fig. 7), and this tail was more conspicuous when the value of $\tilde{P}_n$ was set at a higher value (Fig. 7b). When $\tilde{P}_n = 0.284$, the total area of the grids cells that had GLI values greater than 30 (a value above 30 was not found prior to harvest) was 2132 m^2 (5.3% of the total area); the value based on random harvesting was 863 m^2 (2.2% of the total area). When $\tilde{P}_n = 0.568$, the total area with GLI values greater than 30 was 3013 m^2 (7.5% of the total area), while the values based on random harvest was 1217 m^2 (3% of the total area). These results show that the area under high light (GLI > 30) was more than two times greater under aggregated harvesting than random harvesting.

6. Discussion

Gaps created by harvesting can vary in size, shape, and orientation, and as a consequence, gaps give rise to significant spatial heterogeneity within managed forests. This heterogeneity is thought to play an important role in driving the successional dynamics (Seymour and Hunter, 1999) and in maintaining both species diversity (Thomas et al., 1999; Loehle, 2000) and forest productivity (Botkin et al., 1972; DeAngelis and Gross, 1992; Pacala and Deutschman, 1995; Webster and Lorimer., 2002). Although high-resolution models may not be required to simulate gap-phase recruitment in single-tree gaps (see Deutschman et al., 1999), it is clear that the ability to resolve individual gaps is advantageous for simulating processes that depend on gap size, such as the process of gap partitioning (Busing and Mailly, 2004; Busing and White, 1997). The harvesting algorithm we have developed here is capable of creating gaps that range from single- to multiple-tree gaps in a manner very similar to that observed in stands managed under selection system silviculture. It is thus well suited for examining regeneration processes in selection-managed forests that depend on gap size within the context of spatially explicit, individual-based simulation models.

The ability to specify the gap size distribution is also essential for simulating selection silviculture because it emulates a primary parameter that managers manipulate to promote the regeneration of different species. At the same time, the ability to set harvest targets is important because managers must balance the need to regenerate desired species with the need to maintain timber supply. In order to simulate harvesting practices under selection silviculture, we must be able to represent variability in gap sizes, while conforming to the harvest targets. The harvesting algorithm we have developed

Table 1 – Comparison of the empirical and the simulated patterns of harvest for the calibration and the validation datasets

	Basal area of stumps(m^2 ha^{-1})			Number of stumps (ha^{-1})			Proportion of stumps whose nearest neighbor is also a stump		
	Haliburton (Y_{ba})	Simulated ($\tilde{Y}_{ba}$)		Haliburton (Y_{nm})	Simulated ($\tilde{Y}_{nm}$)		Haliburton (P_n)	Simulated ($\tilde{P}_n$)	
		Mean	S.D.		Mean	S.D.		Mean	S.D.
Calibration	9.13	9.31	0.11	62.22	62.72	0.89	0.284	0.305	0.024
Validation	7.12	7.19	0.22	51.86	52.86	1.71	0.219	0.290	0.044

The values for simulated harvesting are averages of 100 replicate simulations. S.D. represents one standard deviation. In all simulated harvests, the same parameter values derived from the 84 calibration plots were used. The P_n value was set at 0.284, and the probability of cutting down the target tree (P_t) for each size class was defined as follows: $P_{t(pole)} = 0.04$, $P_{t(small)} = 0.18$, $P_{t(med)} = 0.37$, $P_{t(large)} = 0.48$, $P_{t(x\text{-}large)} = 0.62$. The number of target stumps to cut to in each plot ($N_{t(plot)}$) was obtained empirically by analyzing the number of clusters in each plot (see Section 3.2 and Appendix D).

can achieve both of these tasks simultaneously, and thus is capable of creating various harvesting scenarios under the selection system. Once our algorithm is incorporated into an individual-based forest stand simulator, it will be possible to study long-term forest stand development and dynamics under various harvesting scenarios. Such studies may yield new insights into the dynamics of managed stands and provide guidance in designing new management regimes.

6.1. *Empirical algorithm*

Our empirical algorithm was successful in reproducing the harvesting patterns found in Haliburton Forest, and is one of the very few harvesting algorithms that have been tested against field data. It is important to note, however, that we have only compared observed and simulated patterns at a very coarse scale by averaging across all the plots. If empirical and simulated harvests were compared at the plot level, there would likely to be substantial differences between the two. Such plot-level differences are expected to arise due to the stochasticity of the harvesting algorithm and because the observed patterns of harvest will vary from one plot to the next.

Harvest targets in forest management are always set at the stand scale (i.e., cut-blocks), but at the plot level there is no single standard management regime; the harvesting patterns at this scale are mostly determined by the knowledge and the experience of foresters and loggers. Consequently, at this scale, there is bound to be considerable local variation in the patterns of harvest, even if the overall harvest targets were set identically, and even if the same stand were harvested. This local variation in harvesting pattern itself may lead to substantially different stand development and dynamics following a harvest (Pacala and Deutschman, 1995). This potential variation in local harvesting patterns can be captured with our harvesting algorithm by running it repeatedly onto the same stand. If this repeated process is done within a forest simulator, we may also be able to capture the resulting variability in subsequent forest development. Our knowledge of potential variability in the state of the forest years after implementing a particular harvesting regime may have significant implications for forest management and conservation planning.

6.2. *The proportion of stumps in the larger neighborhood*

Our empirical harvesting algorithm was successful in reproducing the spatial distribution patterns of stumps as captured by nearest neighbor statistics. Yet, it must be noted that by considering only the nearest neighbor, we may fail to

Table 2 – Comparison of the target ($C_{(size)}$) and the simulated ($\tilde{C}_{(size)}$) yield for the user-defined algorithm

		Target		Simulated	
				$\tilde{P}_n = 0.28$	$\tilde{P}_n = 0.56$
(a) Total number of harvested trees (ha^{-1})	C	108	$\tilde{C}$	115	120
By size class					
Polewood	$C_{(pole)}$	34	$\tilde{C}_{(pole)}$	43	49
Small sawlog	$C_{(small)}$	33	$\tilde{C}_{(small)}$	33	30
Medium sawlog	$C_{(med)}$	29	$\tilde{C}_{(med)}$	26	28
Large sawlog	$C_{(large)}$	13	$\tilde{C}_{(large)}$	14	13
(b) Total basal area of harvested trees (m^2 ha^{-1})		10.8		10.8	10.8

The target harvest values were arbitrarily determined. The total target basal area of stumps was obtained by summing the target basal area of stumps in each diameter class, which is calculated by multiplying the number of trees to harvest in each class ($C_{(size)}$) by the average basal area per tree for the respective size class.

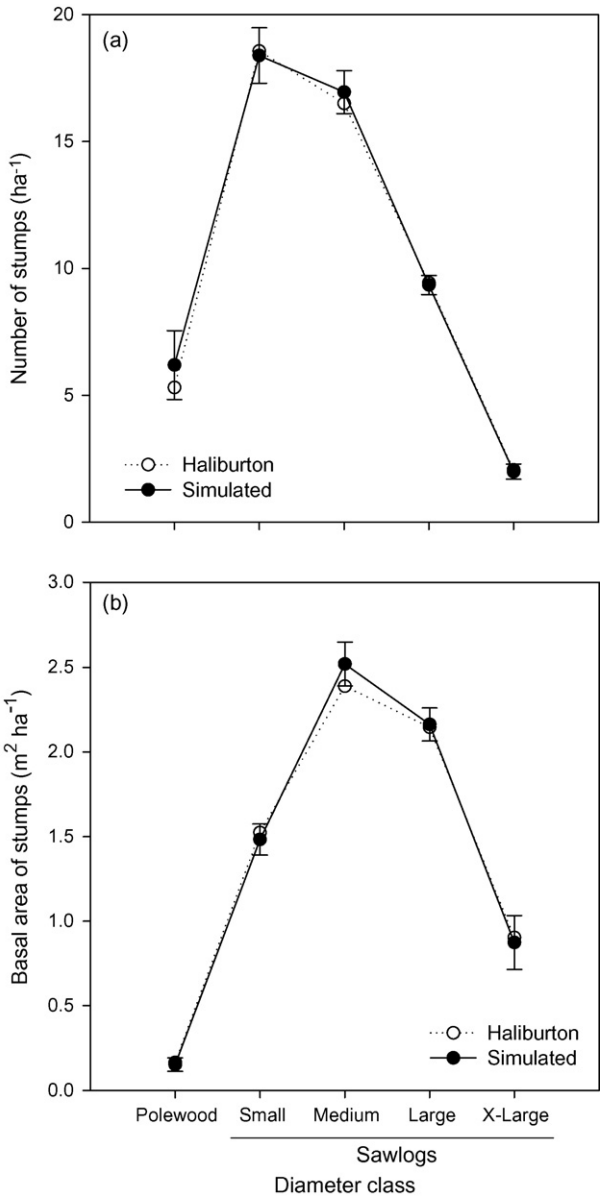


Fig. 5 – Comparison of empirical ($\hat{Y}_{nm(size)}$, $\hat{Y}_{ba(size)}$) and simulated ($\tilde{Y}_{nm(size)}$, $\tilde{Y}_{ba(size)}$) yield by size class (results for the validation dataset): (a) number of trees and (b) basal area harvested by diameter class. Open circles show the empirical values, while the closed circles are average values from the simulated harvest (100 replicated runs). Error bars indicate one standard deviation.

reproduce the full spatial distribution pattern of stumps. For example, the algorithm may reproduce the proportion of stumps whose nearest neighbor is also a stump ($\hat{P}_n$), but may fail to reproduce the proportion of stumps within the four nearest neighbors. Such a discrepancy is expected if the spatial distribution of clusters are different from those found in the empirical data. For example, if the clusters are aggregated, the proportion of stumps in the larger neighborhood might be higher than what would be expected if the clusters were distributed randomly in space. However, we have found that our nearest neighbor algorithm was in fact capable of capturing the spatial pattern of harvesting in the larger neighborhood (Supplementary Material). The success of our harvesting algorithm in reproducing the spatial pattern of stumps in the larger neighborhood may be due to the fact that clusters are being formed centered on the larger sized trees (i.e., preferential harvest of larger stems as target trees). The spatial distribution of large stems in a stand will largely dictate the spatial distribution of clusters, and because these large stems are less abundant in the stand, there may be limited variability in the spatial distribution pattern of clusters.

In the actual practice of selection silviculture, larger trees are also preferentially harvested, and multi-stump gaps are most likely to be created around these individuals. Thus the similarity in the way the trees are selected for harvest in the field and in the algorithm, may have allowed the algorithm to better capture the spatial pattern of harvesting in the larger neighborhood.

6.3. *User-defined algorithm*

In the user-defined algorithm, there are only two sets of parameters that need to be defined prior to running the harvesting algorithm: harvest target by size class ($C_{(size)}$) and the probability of cutting the nearest neighbor (P_n). The former allows the user to modify the intensity and the size-specificity of harvesting, while the latter enables modifications to the spatial distribution of harvesting within the framework of selection silviculture (i.e., single-tree selection and group selection). Both of these input variables are intuitively easy to understand, and therefore should aid in allowing the user to create various harvesting scenarios for use in an individual-based forest simulation model.

Overall, the user-defined algorithm was successful in meeting the target harvest yields, but there were slight differences between the target and the simulated yield in the polewood diameter class (Table 2). This discrepancy between the target and the simulated values is likely to increase as the value of $\tilde{P}_n$ (user-defined input value) or the average cluster size is increased. This pattern is expected for the following reasons: in the harvesting algorithm, the user-defined harvest targets are matched through the selective cut of target trees, but the algorithm has no control over which diameter size trees are harvested during the propagation process of stumps. Thus as the average cluster size is increased, it inevitably results in more trees being harvested without any consideration to its size. Given that the smallest size class (polewood trees) is also the most abundant (Fig. 1), stems belonging to this size class will be most affected by the propagation process, resulting in the largest discrepancy between the target and the simulated yields. However, this difference between the target and the simulated yield in the polewoods is expected to have relatively minor effects on the overall effectiveness of the algorithm for two reasons: (1) due to their size, polewoods have very little influence on the total basal area harvested, (2) polewoods are the most abundant size class within a stand, therefore when the harvested yields are viewed in terms of the proportional values, the target and the simulated values do not differ substantially. The harvest target for polewoods ($C_{(pole)}$) can be expressed proportionally as 0.08 (Table 2), given that the total number of polewoods in the tree map was 1364. Sim-

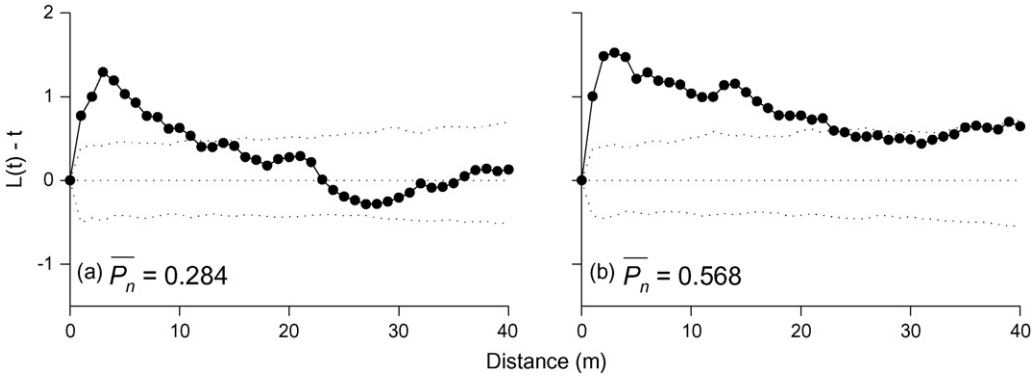


Fig. 6 – The spatial point pattern analysis (Ripley's *K* function) of the harvested trees based on the user-defined harvesting algorithm (closed circles). (a) $\bar{P}_n = 0.284$ and (b) $\bar{P}_n = 0.568$. The *K* function was transformed to $L(t) - t$, where $L(t) = [K(t)\pi]^{1/2}$, and t is distance (Ripley, 1979); $[L(t) - t] = 0$ indicates complete spatial randomness (Poisson process), $[L(t) - t] < 0$ spatial regularity and $[L(t) - t] > 0$ spatial clustering. The dotted lines represent 0.005 and 0.995 quantiles of $L(t) - t$ estimated from 999 simulations of random harvests ($\alpha = 0.01$). The calculated $[L(t) - t]$ values are significantly different from 0, if they lie outside of these bounds.

ulated yields of polewoods ($\tilde{C}_{(pole)}$, Table 2) can be expressed proportionally as 0.085 and 0.089, for $\tilde{P}_n = 0.284$ and 0.568, respectively. Thus, in terms of percentages, the target and simulated yields differ by less than 1%.

6.4. *Comparison with existing harvesting algorithms within SORTIE and other simulators*

Previous studies have examined the effect of various gap sizes on forest stand dynamics using SORTIE (Coates et al., 2003; Deutschman et al., 1997; Ménard et al., 2002). Within the recent release of SORTIE-ND (version 6.05), there are two existing algorithms that can potentially be used to harvest trees in a simulated stand. The first is an algorithm that mimics a single-tree selection. This function does not allow for creation of multi-tree gaps other than by chance (i.e., Bernoulli trial selection of harvested stems); this limits its usefulness in implementing the diverse scenarios of selection silviculture. The second SORTIE-ND algorithm is more flexible in terms of creating gaps of various sizes (or harvesting trees in an aggregated manner), but requires extensive and detailed user input. In this function, a simulated stand is split into grids, and the user defines which grids should be harvested through a visual interface. Because the size of each grid can be set at any value, gaps of any size, form and orientation can be created. This would allow representation of a single-tree selection and group-selection in a simulated stand. However, a major drawback of this algorithm is the fact that the user has to manually define each gap with the visual interface, which greatly limits its usefulness for large-scale simulation studies. Moreover, attempting to meet the harvest targets is not an easy task with this function, as the user will not know the amount of trees (BA and number of trees) that will be cut by selecting a specific grid. Our algorithm, as stated earlier, is able to meet the harvest targets as well as change the level of aggregation of stumps using a small number of user-designated parameters. Once incorporated into SORTIE, it will be possible to automate the entire harvesting process, which is an essential feature if one wishes to run many replicated simulations.

Selection harvesting has been implemented within various forest simulators other than SORTIE. For example, in many forest growth models such as FIBER (Solomon et al., 1995), there are built-in functions that allow simulation of harvesting regimes that may differ in the intensity of harvesting and the size-specificity of harvesting. However, because these models are not spatially explicit, they are not capable of representing variability in gaps sizes (i.e., spatial patterns of harvest). Models such as LSM (McCarter et al., 1998) and LANDIS (Mladenoff and He, 1999; He et al., 2005) are spatially explicit forest simulators capable of representing various types of disturbances, including harvesting based on selection systems. Both of these models can create various spatial and structural patterns of harvest. However, LMS and LANDIS are stand- and landscape-level models, both of which do not address the detailed spatial dynamics of forest structure within stands. Thus, while the harvesting algorithms within these models are capable of creating spatial patterns of harvest at stand-level resolution, they do not create detailed harvesting pattern of stumps within a stand. Söderbergh and Ledermann (2003) reviewed harvesting algorithms that were implemented in five other individual-based forest simulation models (SILVA, MOSES, PROGNAUS, STAND, BWINPro); these algorithms are capable of harvesting trees from a simulated stand in various spatial patterns. However, they do not consider the targets for residual stand structure simultaneously, and this is exactly what we have attempted to do in our algorithm so that we can recreate the harvesting patterns of selection silviculture.

6.5. *Improvements and extensions to the algorithm*

We have described here the basic structure of the harvesting algorithm, which could be profitably modified and extended in a number of directions. For example, the algorithm described here does not take species into consideration, but in many instances a user may wish to harvest only one or a few species

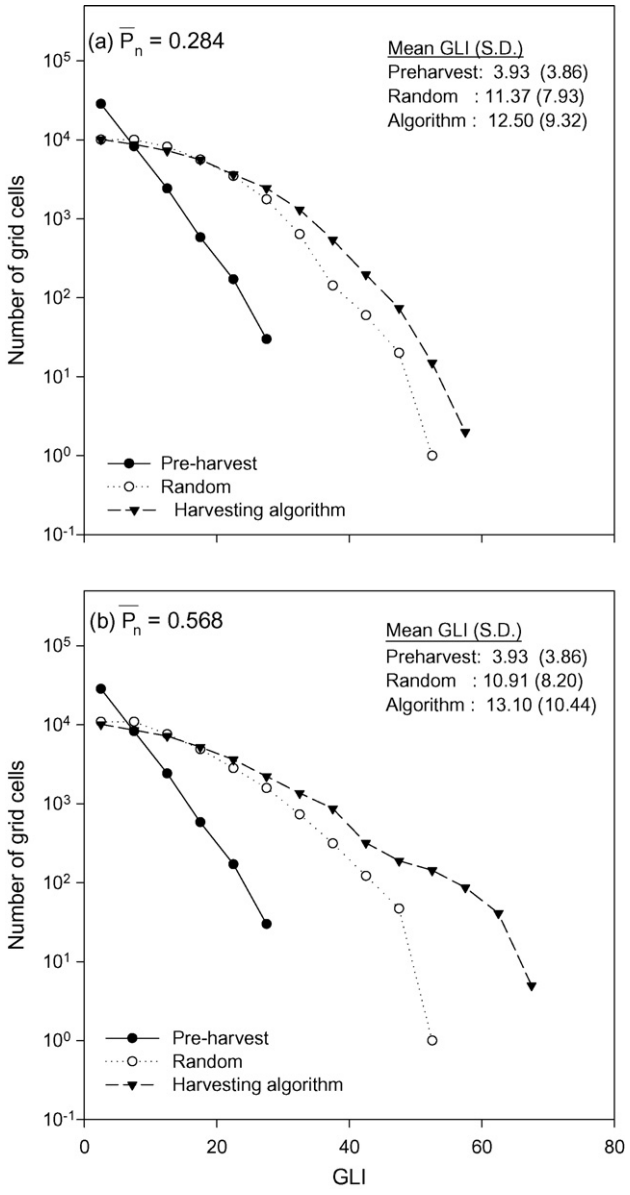


Fig. 7 – Frequency distribution of light levels produced by harvesting regimes that differ in the degree of aggregation: (a) $\bar{P}_n = 0.284$ and (b) $\bar{P}_n = 0.568$. The index of light levels used is the gap light index (GLI), which specifies the percentage of incident, growing season photosynthetic photon flux density (PPFD) that penetrates through the canopy (Canham, 1988). Open circles represent values for random harvesting and the closed inverted triangle represents the harvesting done with the empirical version of the algorithm. As a reference, the distribution of the pre-harvest conditions is given as closed circles.

from a stand. Allowing for such species-specific harvesting is not difficult to implement, since simply setting a new criterion for selecting a target tree can achieve this task. However, the neighboring individuals that are cut due to spatial propagation will not be species-specific (as they were not size-specific), and species other than sugar maple will inevitably be harvested. However, this is also to be expected in the field when a forester attempts to create multi-tree gaps, despite targeting a single species. Thus by modifying the algorithm so that it draws target trees only from a pool of single (or a few) species allows a cut that is primarily composed of the targeted species.

With minor modification, our harvesting algorithm is also capable of representing other types of harvesting regimes. For example, we may wish to simulate structural retention harvesting where trees or patches of forest are left behind to provide structural diversity and habitat for many organisms (e.g., Franklin et al., 1997; Beese and Bryant, 1999; Bebber et al., 2005). The simplest approach with our algorithm would be to designate as residual trees those stems treated as stumps in the current study, thus leaving behind clusters of trees of various sizes after the harvest. Diameter-limit cutting (Nyland, 1992, 2000) could also be implemented by drawing target trees only from a pool of trees within a certain diameter class.

The algorithm could also be made more flexible in terms of the variability in the harvesting intensity among the grid cells. In the user-defined algorithm, the harvesting intensity within a grid cell is primarily dictated by the number of target trees harvested per grid ($\tilde{N}_{t(grid)}$); in the version of the algorithm we have presented here, the theoretical frequency distribution of $\tilde{N}_{t(grid)}$ was a Poisson distribution. However, one may change the harvesting intensity by changing the frequency distribution function, or preventing harvesting in certain grid cells (i.e., $\tilde{N}_{t(grid)} = 0$). Thus, by making such modifications, the algorithm we have presented here can be much more flexible in terms of its implementation, reflecting the wide variability in harvesting practices now common in forestry.

Acknowledgments

This work was supported by the Sustainable Forest Management Network, a Canadian Network Center of Excellence, and by the National Science and Engineering Research Council of Canada. We thank Grant Domke, Sheelah Griffith, Rachel Mayberry, Justin Morgenroth, and Charles Nock for assistance with tree mapping, and Peter Schleifenbaum for access and logistical support at Haliburton Forest.

Appendix A. List of symbols

Input parameters for the nearest neighbor algorithm

N_t	number of target trees to cut down
P_n	probability of cutting down a neighbor tree
P_t	probability of cutting down a target tree

Empricial algorithm: symbols used to denote parameter values calculated from field data

$\hat{N}_t$	value of N_t calculated from field data
$\hat{N}_{t(grid)}$	value of N_t calculated from field data (used with tree maps that are divided into grids—Section 3.5)
$\hat{N}_{t(plot)}$	value of N_t calculated from field data (used with plot-based tree maps—Section 3.4)
$\hat{P}_n$	value of P_n calculated from field data
$\hat{P}_t$	value of P_t calculated from field data
$\hat{Y}_{ba}$	observed total basal area harvested

$\hat{Y}_{ba(size)}$ observed basal area harvested by size class
$\hat{Y}_{nm}$ observed total number of stems harvested
$\hat{Y}_{nm(size)}$ observed number of stems harvested by size class

Symbols used to denote variables calculated from simulation output
$\tilde{P}_n$ value of P_n calculated from simulated data
$\tilde{Y}_{ba}$ simulated total basal area harvested
$\tilde{Y}_{nm}$ simulated total number of stems harvested
$\tilde{Y}_{ba(size)}$ simulated basal area harvested by size class
$\tilde{Y}_{nm(size)}$ simulated number of stems harvested by size class

User-defined algorithm
$\bar{N}_t$ value of N_t calculated from the user-defined harvest targets
$\bar{P}_n$ value of P_n defined by the user
$\bar{P}_t$ value of P_t calculated from the user-defined harvest targets

C target yield (total number of stems)
$C_{(size)}$ target yield by size class (total number of stems)
$\tilde{C}$ simulated yield (total number of stems)
$\tilde{C}_{(size)}$ simulated yield by size class (total number of stems)

Appendix B. Calculating the average number of clusters in a plot ($\hat{N}_t$)

Here, we borrow some terminology from graph theory to explain how the number of clusters in the field data (i.e., the value of $\hat{N}_t$) were identified and counted. In this explanation, a plot will be represented by a directed graph, where all trees within the plots are considered as vertices. Arcs (directed edges) that connect the vertices are created by repeating the following procedure for each vertex that is a stump: (1) determine the nearest neighbor, and (2) if the nearest neighbor is a stump, connect the two vertices (stumps) by an arc. The nearest neighbor is the head of an arc, and the target stump is the tail (example: in Fig. B1, stump 1 is the tail, and stump 2 is the head of an arc). Note that there are special cases where a stump can be both the tail and the head of an arc between its nearest neighbor (Fig. B1, stumps 3 and 4). Once all the arcs are identified, the plot is represented by a directed graph, and the identification of clusters is equivalent to identifying path-connected components (sub-graph) within this graph. In the example in Fig. B1, six clusters are identified. A cluster can be of various sizes (Fig. B1), where a single vertex is also considered a cluster (i.e., "singleton", cluster 2 in Fig. B1). In this method of cluster identification, a vertex can belong to multiple clusters as found in the case for stump 6 in Fig. B1; this stump can belong to a cluster originating from stump 5, or a different cluster originating from stump 7. Note that clusters 4 and 5 cannot be considered as a single cluster, as there is no path from stump 5 to stump 7, or vice versa.

Appendix C. Reciprocal pairs

The presence of reciprocal tree pairs (pairs in which both trees are each other's nearest neighbor) presents a problem when implementing a nearest neighbor algorithm. To better understand this problem, consider the behaviour of the algorithm when it selects one of the trees in a reciprocal pair as a target tree, then cuts both the target tree and second tree in the pair (with probability P_tP_n). At this point, the contagious process cannot spread any further because both trees are each other's nearest neighbor. Thus, fewer neighboring trees will be cut than would be expected if there were no reciprocal tree pairs in a stand, and the spatial pattern of stumps will be less aggregated than was intended. In particular, the proportion of stumps whose nearest neighbor is also a stump ($\tilde{P}_n$) will be less than the value of the input parameter P_n (the probability of cutting the nearest neighbor of a target tree). To ensure that $\tilde{P}_n$ approximates the input parameter value P_n, we adjusted the probability of cutting the nearest neighbor in a reciprocal pair as follows.

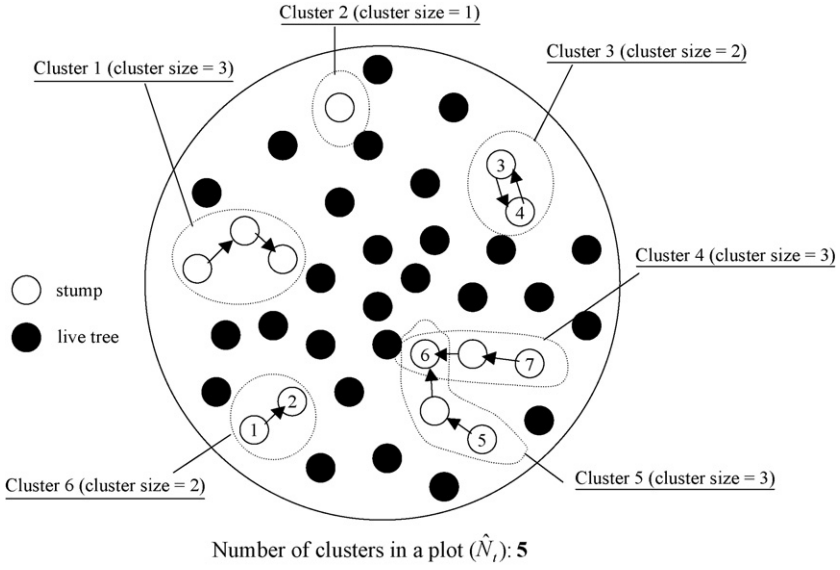
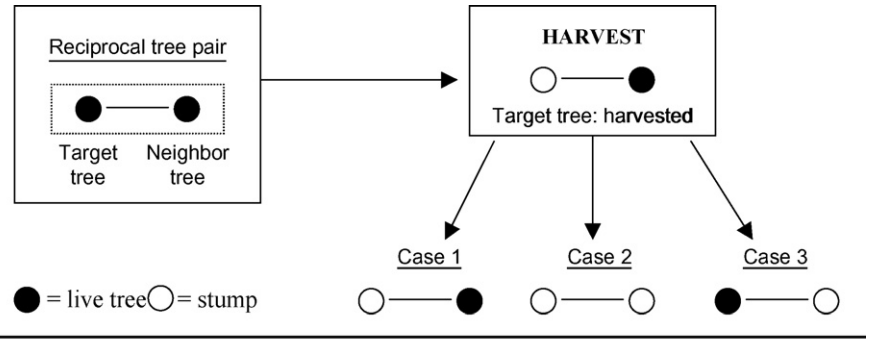


Fig. B1 – An illustration to explain how the clusters were identified in a plot. Open circles represents stumps and closed circles live trees. See text for details.



	Case 1	Case 2	Case 3
Probability of occurrence	$(1-P_n)$	$(P_n)^2$	$(P_n)(1-P_n)$
Number of stumps (in the case of m reciprocal tree pairs)	$m(1-P_n)$	$2m(P_n)^2$	$m(P_n)(1-P_n)$
Number of stumps whose nearest neighbor is also a stump (in the case of m reciprocal tree pairs)	0	$2m(P_n)^2$	0

Fig. C1 – An illustration of three possible outcomes of harvesting, when the algorithm draws one of the reciprocal tree pairs as a target tree and designates it as a stump. See text for details.

There are three possible outcomes (Fig. C1) when the algorithm first selects and cuts a target tree from a reciprocal pair: (1) the algorithm does not cut the neighboring tree if a random number is greater than, or equal to, P_n, (2) the algorithm cuts the neighboring tree if the random number is less than P_n, and recuts the target tree if a second random number is less than P_n, (3) the algorithm cuts the neighboring tree if the random number is less than P_n, and resurrects the target tree if the second random number is greater than, or equal to, P_n The probabilities of these three outcomes are $(1-P_n)$, P_n^2, and $P_n(1-P_n)$, respectively. Thus, the probability of a stump's nearest neighbor being a stump, given that it belongs to a reciprocal pair is

$$\tilde{P}_n = \frac{2m(P_n)^2}{m(1-P_n)+2m(P_n)^2+m(P_n)(1-P_n)} = \frac{2P_n{}^2}{P_n{}^2+1}, \quad (0 \le P_n \le 1) \tag{C.1}$$

where, m is the number of target trees selected from reciprocal pairs, $2m(P_n)^2$ is the number of stumps that have a stump as its nearest neighbor, and $[m(1-P_n)+mP_n^2+mP_n(1-P_n)]$ is the total number of stumps.

To ensure that $\tilde{P}_n$ approximates the input parameter value P_n, the algorithm calculates an adjusted probability of cutting the nearest neighbors in reciprocal pairs as follows. First, the adjusted probability ($P_{n(adj)}$) is substituted for P_n in Eq. (C.1):

$$\tilde{P}_n = \frac{2P_{n(adj)}^2}{P_{n(adj)}^2+1}, \quad (0 \le P_{n(adj)} \le 1) \tag{C.2}$$

then P_n is substituted for $\tilde{P}_n$ in Eqs. (C.2) and (C.2) is rearranged to obtain:

$$P_{n(adj)} = \sqrt{\frac{P_n}{P_n+1}} \tag{C.3}$$

In addition to adjusting P_n, we also allowed the algorithm to switch the status of the reciprocal tree pairs when the target tree is resurrected (i.e., target is assigned a stump status, while the neighbor becomes a live tree) to ensure that simulated basal area yield ($\hat{Y}_{ba}$) approximates the observed yield ($\tilde{Y}_{ba}$). This was deemed necessary because the neighbor trees are cut down without any consideration of their size, while target trees are selected based on their size to meet the target basal area of harvest. This modification does not affect $\tilde{P}_n$, but it allows the algorithm to better reproduce the observed yield.

Appendix D. Determining the number of target trees to cut ($\tilde{N}_t$) in the user-defined version of the algorithm

In the empirical version of the model, the number of target trees to cut was determined by examining the number of clusters found in each plot (Appendix B). In the user-defined version of the model, however, we have no direct means to determine how many target should be cut in order to attain the user-defined target number (prescription) of trees to harvest. The number of target trees that should be cut would depend on the user-defined values of the prescription and $\tilde{P}_n$. To determine the number of target trees to cut, we have used an iterative approach to solve for $\tilde{N}_t$, where we run a number of preliminary harvesting simulations by systematically varying the number of target trees that are cut. By doing so, we can choose the number of target trees to cut ($\tilde{N}_t$) that would

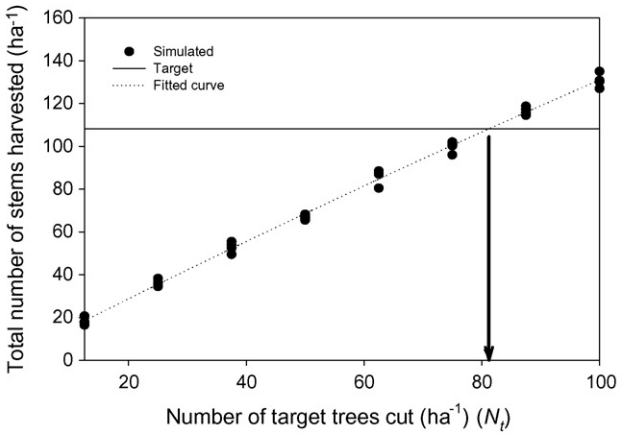


Fig. D1 – An example showing the relationship between the number of target stumps cut and the total number of stems harvested in the algorithm. See text for details.

be necessary to match the user-defined target number of trees to harvest.

In Fig. D1, we illustrate an example of this procedure. In this example, the target number of trees to be harvested over the entire plot was set at 108 trees per hectare by the user (the horizontal line in the illustration below). The *x*-axis represents the numbers of target trees that were cut in the simulation, and *y*-axis represents the total number of trees harvested by the algorithm. In this example, four replicate simulation runs were done for each level of the systematically varied *x*-value. When all data points are obtained (closed circle), a polynomial curve fitting is used to get a regression line through the data points (dashed line). The value of *x* for the point at which the horizontal line (target value) and the regression line intersect is determined as the number of target stumps ($\bar{N}_t$) necessary to attain the target harvest value (arrow) (note: while a simple linear regression may be used instead of a polynomial curve fitting in this example, the relationship is not necessarily linear).

Once the total number of target trees to be cut is determined ($\bar{N}_t$), they are distributed among the grid cells. For the current version of the model, we distributed ($\bar{N}_t$) randomly into the grid cells; the frequency distribution of the number of target stumps per grid cell will theoretically be a Poisson distribution.

Appendix E. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.ecolmodel.2007.09.007.

REFERENCES

Arbogast, C., Jr., 1957. Marking guides for northern hardwoods under the selection system. USDA Forest Service, Lake States Forest Experimental Station Paper No. 56.

Baddeley, A., Turner, R., 2005. Spatstat: an R package for analyzing spatial point patterns. J. Stat. Softw. 12, 1–42.

Bebber, D.P., Cole, W.G., Thomas, S.C., Balsillie, D., Duinker, P., 2005. Effects of retention harvests on structure of old-growth *Pinus strobus* L. stands in Ontario. For. Ecol. Manage. 205, 91–103.

Beese, W.J., Bryant, A.A., 1999. Effect of alternative silvicultural systems on vegetation and bird communities in coastal montane forests of British Columbia, Canada. For. Ecol. Manage. 115, 231–242.

Bollobás, B., 2002. Modern Graph Theory. Springer-Verlag, New York, NY, 408 pp.

Bormann, F.H., Likens, G.E., 1979. Pattern and Process in a Forested Ecosystem. Springer-Verlag, New York, NY, 253 pp.

Botkin, D.B., Janak, J.F., Wallis, J.R., 1972. Some ecological consequences of a computer model of forest growth. J. Ecol. 60, 849–872.

Busing, R.T., Mailly, D., 2004. Advances in spatial, individual-based modeling of forest dynamics. J. Veg. Sci. 15, 831–842.

Busing, R.T., White, P.S., 1997. Species diversity and small-scale disturbance in an old growth temperate forest: a consideration of gap partitioning concepts. Oikos 78, 562–568.

Canham, C.D., 1988. An index for understory light levels in and around canopy gap. Ecology 65, 1634–1638.

Caspersen, J.P., Saprunoff, M., 2005. Seedling recruitment in a northern temperate forest: the relative importance of supply and establishment limitation. Can. J. For. Res. 35, 978–989.

Coates, K.D., Canham, C.D., Beaudet, M., Sachs, D.L., Messier, C., 2003. Use of a spatially explicit individual-tree model (SORTIE/BC) to explore the implications of patchiness in structurally complex forests. For. Ecol. Manage. 186, 297–310.

DeAngelis, D.L., Gross, L.J., 1992. Individual-based Models and Approaches in Ecology Populations: Communities and Ecosystems. Chapman & Hall, New York, NY, US, 525 pp.

Deutschman, D.H., Levin, S.A., Devine, C., Buttel, L.A., 1997. Scaling from trees to forests: analysis of a complex simulation model. Science Online http://www.sciencemag.org/feature/data/deutschman/index.htm.

Deutschman, D.H., Levin, S.A., Pacala, S.W., 1999. Error propagation in forest succession models: the role of fine-scale heterogeneity in light. Ecology 80, 1927–1943.

Drossel, B., Clar, S., Schwabl, F., 1993. Exact results for the one-dimensional self-organized critical forest-fire model. Phys. Rev. Lett. 71, 3739–3742.

Franklin, J.F., Berg, D.R., Thornburgh, D.A., Tappeiner, J.C., 1997. Alternative silvicultural approaches to timber harvesting: variable retention harvest systems. In: Kohm, K.A., Franklin, J.F. (Eds.), Creating a Forestry for the 21st Century. Island Press, Washington, DC, pp. 111–139.

Gustafson, E., Shifley, S., Mladenoff, D., Nimefro, K., He, H., 2000. Spatial simulation of forest succession and timber harvesting using LANDIS. Can. J. For. Res. 30, 32–43.

Hansen, G.D., Nyland, R.D., 1987. Effects of diameter distribution on the growth of simulated unevenaged sugar maple stands. Can. J. For. Res. 17, 1–8.

He, H.S., Li, W., Sturtevant, B.R., Yang, J., Shang, B.Z., Gustafson, E.J., Mladenoff, D.J., 2005. LANDIS 4.0 users guide. LANDIS: a spatially explicit model of forest landscape disturbance, management, and succession. USDA Forest Service, North Central Research Station. General Technical Report NC-263. St. Paul, MN, 93 pp.

Huth, A., Ditzer, T., 2001. Long-term impacts of logging in a tropical rain forest a simulation study. For. Ecol. Manage. 142, 33–51.

Jones, T.A., 2006. Growth and physiological responses of canopy tree species to selection harvests in a northern hardwood forest. Ph.D. Dissertation, University of Toronto.

Loehle, C., 2000. Strategy space and the disturbance spectrum: a life-history model for tree species coexistence. Am. Nat. 156, 14–33.

Malamud, B.D., Morein, G., Turcotte, D.L., 1998. Forest fires: an example of self-organized critical behavior. Science 281, 1840–1842.
McCarter, J.B., Wilson, J.S., Baker, P.J., Moffett, J.L., Oliver, C.D., 1998. Landscape management through integration of existing tools and emerging technologies. J. Forest. 96, 17–23.
Ménard, A., Dubé, P., Bouchard, A., Marceau, D.J., 2002. Release episodes at the periphery of gaps: a modeling assessment of gap impact extent. Can. J. For. Res. 32, 1651–1661.
Mladenoff, D.J., He, H.S., 1999. Design and behavior of LANDIS, an object-oriented model of forest landscape disturbance and succession. In: Mladenoff, D.J., Baker, W.L. (Eds.), Advances in Spatial Modeling of Forest Landscape Change: Approaches and Applications. Cambridge University Press, Cambridge, UK, pp. 125–162.
Nuttle, T., Haefner, J.W., 2007. Design and validation of a spatially explicit simulation model for bottomland hardwood forests. Ecol. Model. 200, 20–32.
Nyland, R.D., 1992. Exploitation and greed in eastern hardwood forests. J. Forest. 90 (1), 33–37.
Nyland, R.D., 1998. Selection system in northern hardwoods. J. Forest. 96, 18–21.
Nyland, R.D., 2000. Forestry and silviculture in the Northeast—past, present, and the probable future. In: Proceeding of the Society of American Foresters National Convention. Society of American Foresters, Washington, DC, pp. 319–325.
O'Hara, K.L., 2002. The historical development of uneven-aged silviculture in North America. Forestry 75, 339–346.
OMNR, 1998. A Silvicultural Guide for the Tolerant Hardwood Forest in Ontario. Ontario Ministry of Natural Resources. Queen's Printer for Ontario, Toronto, Ontario, 500 pp.
OMNR, 2000. A Silvicultural Guide to Managing Southern Ontario Forests, Version 1.1. Ontario Ministry of Natural Resources. Queen's Printer for Ontario, Toronto, 648 pp.
OMNR, 2004. Ontario Tree Marking Guide, Version 1.1. Ontario Ministry of Natural Resources. Queen's Printer for Ontario, Toronto, 252 pp.
Pacala, S., Canham, C., Saponara, J., Silander, J., Kobe, R., Ribbens, E., 1996. Forest models defined by field measurements. II. Estimation, error analysis and dynamics. Ecol. Monogr. 66, 1–44.
Pacala, S.W., Deutschman, D.H., 1995. Details that matter: the spatial distribution of individual trees maintains forest ecosystem function. Oikos 74, 357–365.
Pacala, S.W., Rees, M., 1998. Models suggesting field experiments to test two hypotheses explaining successional diversity. Am. Nat. 152, 729–737.
Parrott, L., Kok, R., 2001. A generic primary producer model for use in ecosystem simulation. Ecol. Model. 139, 75–99.
Pausas, J.G., Austin, M.P., Noble, I.R., 1997. A forest simulation model for predicting eucalypt dynamics and habitat quality for arboreal marsupials. Ecol. Appl. 7, 921–933.
Pennanen, J., Greene, D.F., Fortin, M.-J., Messier, C., 2004. Spatially explicit simulation of long-term boreal forest landscape dynamics: incorporating quantitative attributes. Ecol. Model. 180, 195–209.
Pommerening, A., Murphy, S.T., 2004. A review of the history, definitions and methods of continuous cover forestry with special attention to afforestation and restocking. Forestry 77, 27–44.
Pretzsch, H., Biber, P., Dursky, J., 2002. The single tree-based stand simulator SILVA: construction, application and evaluation. For. Ecol. Manage. 162, 3–21.
Ripley, B.D., 1979. Tests of 'randomness' for spatial point patterns. J. R. Stat. Soc. B 41, 368–374.
Rowe, J.S., 1972. Forest regions of Canada. Environment Canada, Can. For. Serv. Publ. 1300.
Seymour, R.S., Hunter Jr., M.L., 1999. Principles of ecological forestry. In: Hunter Jr., M.L. (Ed.), Managing Biodiversity in Forest Ecosystems. Cambridge University Press, pp. 22–61.
Shugart, H.H., 1984. A Theory of Forest Dynamics. Springer-Verlag, New York, NY, US, 278 pp.
Söderbergh, I., Ledermann, T., 2003. Algorithms for simulating thinning and harvesting in five European individual-tree growth simulators: a review. Comp. Electron. Agric. 39, 115–140.
Solomon, D.S., Herman, D.A., Leak, W.B., 1995. FIBER 3.0: An ecological growth model for northeastern forest types. USDA Forest Service General Technical Report NE-204.
Thomas, S.C., Halpern, C.B., Liguori, D.A., Falk, D.A., Austin, K.A., 1999. Plant diversity in managed forests: understory responses to silvicultural thinning and fertilization in Douglas-fir plantations. Ecol. Appl. 9, 864–879.
Tilman, D., 1994. Competition and biodiversity in spatially structured habitats. Ecology 75, 2–16.
Webster, C.R., Lorimer, C.G., 2002. Single-tree versus group selection in hemlock-hardwood forests: are smaller openings less productive? Can. J. For. Res. 32, 591–604.