NRC Research Press

ARTICLE

A Voronoi tessellation based approach to generate hypothetical forest landscapes

Gregor Passolt, Miranda J. Fix, and Sándor F. Tóth

Abstract: Optimization models used for forest planning can be computationally complex and the demand for real forest data to test them far exceeds the supply. As a result, hypothetical forest landscapes are often used, although their capacity to match the characteristics of real forests is limited and they offer little control over important landscape metrics such as average adjacency. Using four landscape metrics that are believed to be relevant to the computational efficiency of forest harvest scheduling models, we describe a new method for generating hypothetical landscapes of prespecified characterization. The new approach produces landscapes based on Voronoi tessellation, created from points chosen by a combination of random point processes. Through a series of multiple regressions, the proposed algorithm determines appropriate control parameters to ensure that the output landscape will match target characteristics within a given statistical tolerance and with a predefined probability. The new method can produce landscapes with a wide range of specifications, covering the characteristics of real forests and extending into extreme cases unlikely to be encountered in reality. At the same time, the method provides greater flexibility and control over the generated landscapes than previous methods.

Résumé : Les modèles d'optimisation utilisés pour la planification forestière peuvent nécessiter des calculs complexes et la demande pour de vraies données forestières afin de les tester surpasse largement l'offre. Par conséquent, des paysages forestiers hypothétiques sont souvent utilisés bien que leur capacité à reproduire les caractéristiques de vraies forêts soit limitée et qu'ils offrent peu de contrôle sur les métriques importantes du paysage, telles que la contiguïté moyenne. À l'aide de quatre métriques du paysage considérées comme pertinentes pour l'efficacité des calculs de modèles de planification des coupes, nous décrivons une nouvelle méthode pour générer des paysages hypothétiques dont les caractéristiques sont prédéterminées. La nouvelle approche produit des paysages basés sur la tessallation Voronoi, créée à partir de points choisis par une combinaison de processus ponctuels aléatoires. Au moyen d'une série de régressions multiples, l'algorithme proposé détermine les paramètres de commande appropriés pour s'assurer que le paysage ainsi généré possèdera les caractéristiques cibles avec une tolérance statistique donnée et une probabilité prédéterminée. La nouvelle méthode peut produire des paysages avec une vaste gamme de spécifications qui correspondent aux caractéristiques de vraies forêts mais qui peuvent aussi représenter des cas extrêmes qui ont peu de chances d'être rencontrés dans la réalité. Cette méthode offre en même temps une plus grande flexibilité et un meilleur contrôle que les méthodes précédentes sur les paysages qui sont générés. [Traduit par la Rédaction]

Introduction

Spatial optimization is often used in forest management to identify harvest plans that maximize timber revenues or other objectives. Historically, timber production was the top priority, but Tóth and McDill (2009) showed that many objectives for forest use, including recreational value, carbon sequestration, wildlife habitat management, and watershed protection, can be considered jointly to develop management schemes that balance all of these considerations and use the forest resources as efficiently as possible. Forests are typically divided into management units to give forest managers small, well-defined regions on which different management actions can be carried out (Petroski 2006); such actions could include diverse treatments such as thinning, cutting, doing nothing, or developing recreational facilities. Optimization models are often required to find management schedules for each unit extending for a given planning horizon, sometimes close to a century in length, and subject to a variety of financial, logistical, and environmental constraints.

Spatially explicit harvest scheduling models are typically classified based on the spatial constraints present in the problems they are used to solve. One of the most common spatial restrictions are the adjacency or greenup constraints. These constraints limit the size of contiguous clearcuts within a predefined time period called greenup or exclusion period. Problems with adjacency constraints are divided into two main types. Unit restriction models (URMs) are for problems where no two adjacent management units can be harvested within the greenup period because the total area of any pair of adjacent units would exceed the maximum allowable harvest opening size. Area restriction models (ARMs) are a generalization of the URM: adjacent units may be harvested within the greenup period as long as their combined area is less than the maximum clearcut size. Thus, in an ARM, the area of each management unit must be taken into account to determine whether a specific prescription (i.e., harvest regime) is feasible (Murray 1999). ARMs are more difficult to formulate and to solve to optimality.

Other classes of spatial harvest scheduling models include the minimum patch size problem, which requires a given forest habitat type, such as old-forest habitat, to occur in contiguous patches of a certain minimum size (Rebain and McDill 2003). Spatial restrictions also arise in the related reserve selection literature. One example is the connectivity problem (Önal and Briers 2006; Conrad and Smith 2012) where the selected sites must be connected by a network of other protected sites.

Received 18 June 2012. Accepted 6 November 2012.

G. Passolt, M.J. Fix, and S.F. Tóth. Quantitative Resource Ecology and Management, University of Washington, Seattle, WA 98195, USA.
Corresponding author: Sándor F. Tóth (e-mail: toths@uw.edu).

Published at www.nrcresearchpress.com/cjfr on xx xx 2013.

The above models rely on complex systems of linear inequalities and objective functions, called mathematical programs, to ensure that the spatial restrictions are met. Forest analysts need to test the models by using real or hypothetical forest data to see if the models are computationally feasible. Ideally, the models are tested on data sets covering a wide range of potential topologies and geometries to understand the conditions under which they perform well and, more importantly, under which they do not perform well. Unfortunately, real forest data are in short supply for proprietary and other reasons, often leaving hypothetical data as the only option (e.g., McDill and Braze 2000; McDill et al. 2002; Rebain and McDill 2003; Constantino et al. 2008; McNaughton and Ryan 2008). Moreover, the tools used to produce these hypothetical forests do not provide explicit control over most of the output characteristics; thus, they limit the value of the computational tests for new harvest scheduling models.

For the purposes of forest planning models, a forest landscape is defined as a set of polygons representing the management units. In a review of existing generators, Li et al. (2010) divided the existing algorithms by method. The two most promising candidates were based on random graphs and Voronoi diagrams, but in their current states, none of the existing methods meet the goals outlined above. Specifically, the ability to vary the output characteristics in a controlled manner is absent. Next, we provide an overview of the relative merits of existing methods.

Optimization models "see" a landscape as a table of adjacency relations among management units and a list of corresponding areas and other information pertinent to each unit. Thus, it is natural to think of a forest landscape as a graph where management units correspond to nodes and adjacencies are indicated by edges, with a vector of areas mapped in correspondence with the nodes. Associating other characteristics with the nodes, such as initial age or road access, is a trivial addition. Under this abstraction, every possible landscape corresponds to a planar graph with an area mapping. The generation of random graphs has been a topic of significant mathematical research. Erdős and Rényi (1960) analyzed the properties of random graphs created on the basis of a specific degree (number of nodes) and then selecting a certain number of edges uniformly from the set of all possible edges. Unfortunately for this application, their results show that it is unlikely for purely random graphs that have more than half as many edges as nodes to be planar. This low upper bound on the number of adjacencies for planarity would be a challenge for following a random graph-based approach to generate hypothetical forests.

Constantino et al. (2008) generated hypothetical forests in a deterministic way by using as building blocks two specific graphs, a four-node cycle denoted as "F-instance" and a more complicated 10-node graph called "G-instance". Grids of F- and G-instances are connected to each other, with each node having an assigned area. This method gives good control over the number of management units, but is in no way random. Topologically, there is little or no difference between hypothetical landscapes of equal sizes constructed by this method. Many implementations of the ARM and the URM problems exhibit improved computational performance if supplemental information on specific subgraphs, like cycles or cliques of certain sizes, are incorporated in them as inequalities (e.g., Goycoolea et al. 2005). It is possible that a model implementation works very efficiently or inefficiently on a certain local topology, so when hypothetical test forests are composed entirely of repeating subgraphs carrying that specific topology (as in Constantino et al. 2008), the evaluation of computational performance could be biased. To control for such possibilities in testing a model, some degree of randomness should be present in generated data.

The MAKELAND program of McDill and Braze (2000) operates by first generating a set of random nodes and then connecting each node to a given number of its nearest neighbors. These initial nodes are placed randomly with an inhibition parameter, a minimum separation distance between the points. To create planarity, the algorithm deletes intersecting lines until there are no crossings, always deleting the line that has the most intersections per unit length. The faces of the graph are taken as the forest management units, and lines continue to be deleted until the number of faces matches a user-defined level. This approach offers control over the resulting adjacencies through two parameters: the inhibition parameter governing point placement and the connection parameter giving the number of connections per point initially drawn. The authors note that a high inhibition parameter "tends to create maps with a more even distribution of polygon sizes." In sum, the MAKELAND algorithm allows for some control over area distribution, but not over vertex degree (average number of adjacent units per unit) distribution.

Another hypothetical landscape-generating approach that yields planar maps uses Voronoi tessellations or diagrams. Voronoi diagrams (also called Thiessen polygons) are defined based on a set of points, where a polygon for each point is created enclosing all of the area that is closer to it than to any other point. Voronoi diagrams have wide-ranging applications, such as behavioral ecology, image compression, and cell biology (Du et al. 1999), and algorithms for computing Voronoi diagrams are readily available. Barrett (1997) applied Voronoi tessellations to a related forest management problem where the goal is to delineate the units within a real, undivided forest. Others who used Voronoi diagrams include Wyszomirski and Weiner (2009) who looked at crowding and competition in plants and found that the area distribution of the polygons was dependent on the level of clustering in the initial points. As Li et al. (2010) pointed out, one problem with the landscapes generated from Voronoi diagrams is that the resulting polygons are always convex. Additionally, the corners of the Voronoi polygons are formed by the intersection of three boundaries, whereas corners of degree 4 (as would be formed by two intersection lines) are common in landscapes that have been divided into management units by humans. We show that these limitations can be overcome in the initial point placement process for Voronoi diagrams and in postprocessing the resulting hypothetical landscape.

This paper describes the development of another Voronoi diagram based algorithm, called rlandscape, that improves on the existing methods. We identify four landscape characteristics thought to affect the computational performance of harvest scheduling models and present a procedure that allows tight user control over these parameters. By drawing the initial points from a mixture of four random distributions and editing the resulting diagram, all of the known weaknesses of the Voronoi diagram method are eliminated. We show that the range of landscape characteristics provided by the new method more than covers the range of characteristics observed in real landscapes and in those generated by MAKELAND. The ability to target characteristic ranges is shown to work efficiently enough to quickly produce a large number of landscapes with prespecified characteristics.

Methods

We start with a description of the metrics we will use to characterize a landscape and then detail the point processes used to create Voronoi tessellations. A description of postprocessing to allow for polygon nonconvexity is next. Finally, we describe the modeling performed to relate input parameters and output characteristics through the stochastic landscape creation process.

Choosing metrics

The number of polygons, area distribution, and degree distribution are the primary landscape characteristics that have been shown to affect the computational complexity of harvest scheduling models (see McDill and Braze 2000; Constantino et al. 2008; Tóth et al. 2012). Presented below are the metrics used to describe the area and degree distributions. Taken together, all of these metrics define what we refer to as the *characteristics* of a given landscape.

Number of management units (n)

There is a strong agreement among researchers that the number of management units (n) in a spatial forest planning problem has a large impact on the computational properties of harvest scheduling models. While many argue that a larger number of units makes the problems harder to solve (e.g., Goycoolea et al. 2005), some of the empirical evidence is mixed in this regard (Tóth et al. 2012). Our proposed landscape generator will allow the user to control for this metric.

Management unit area (CV)

The second landscape characteristic that we use in our proposed Voronoi approach is total forest area or average management unit area. It is important to emphasize that for a hypothetical landscape, total forest area (or average unit area) can be set post hoc without affecting the geometry of the landscape. This is because the distance units used are arbitrary. However, the distribution of area among the polygons is also important. The difference between the ARMs and URMs introduced above is the size of management units relative to the maximum allowable contiguous harvest area (Murray 1999), and the solution methods of the much more difficult ARM problems usually involve groupings of small polygons (Goycoolea et al. 2005; Önal and Briers 2006; McNaughton and Ryan 2008). Thus, we need to consider a measure of the spread of the distribution of areas. Standard deviation is a poor choice because in a hypothetical landscape, the units of standard deviation are just as arbitrary as the units of area. The mean polygon area, and thus the standard deviation as well, will vary significantly with both the number of polygons and the area of the total landscape, which would prevent comparisons between landscapes; thus, a relative measure is required. The coefficient of variation (CV) (defined as standard deviation divided by mean times 100%) is an appropriate measure of the spread of the area distribution because it is a standardized standard deviation and allows for comparisons between forest landscapes, both real and hypothetical, at any scale.

Vertex degree distribution (μ_d, σ_d)

Vertex degree (also known as degree) denotes the number of management units adjacent to a given unit. Many believe that the distribution of vertex degrees across the forest landscape has an impact on the solvability of harvest scheduling models. As an example, Tóth et al. (2012) argued that degree distribution directly affects the extent to which some of the critical constraints used in many ARMs can be strengthened for better computational performance. This finding agrees with work done on even broader generalizations of similar graph theoretical problems that use degrees extensively in bounding complexity and simplifying problem formulation (Berman and Fürer 1994). Unlike area, the degree distribution is scale invariant; thus the standard measures of a distribution, mean and standard deviation, μ_d and σ_d, are appropriate in this case. In spatially explicit harvest scheduling models, both *weak* and *strong* notions of adjacency are used. Two polygons are *weakly adjacent* if they share a finite number of points (e.g., touching only at one corner) and are *strongly adjacent* when they share an infinite number of points, i.e., they have a common border (Goycoolea et al. 2005). In this study, we consider only the strong adjacency type, and as a result, the hypothetical landscapes generated will be equivalents of planar graphs, where each node corresponds to a polygon and edges between nodes correspond to a shared border. Bollobás (1998) gave a proof that in a planar graph with $n > 3$ vertices, the maximum number of edges is $e = 3n - 6$. Since each edge connects two vertices, the maximum average degree is $\max(\mu_d) = 2(3n - 6)/n = 6(n - 2)/n$, which approaches 6 in the limit as n approaches infinity. The lower bound on μ_d is dependent on how connected the landscape is. Technically, μ_d would be 0 if each polygon was completely isolated, but it approaches 2 for connected graphs. Our Voronoi tessellation based method will not enforce full connectivity, as real forests often contain disjoined regions.

In summary, the four landscape statistics that our proposed landscape generator will have control over are the number of units, n, coefficient of variation for unit area, CV, and mean and standard deviation of vertex degree, μ_d and σ_d, respectively. Since these metrics are readily calculated from typical forest data, adjacency and area tables, the outputs of the proposed generator can easily be compared with real data. As a reference point, we calculated statistics for forests with adjacency tables or area tables posted as public data on the University of New Brunswick's Integrated Forest Management Lab website (http://www.unb.ca/fredericton/forestry/research/ifmlab/index.html). Posted forests were omitted only if they were hypothetical, and statistics describing the degree distribution were omitted if the published adjacencies included weak adjacencies.

Generating a landscape

This section describes how a landscape is generated with the proposed approach. First, points are chosen using a combination of point processes, then a Voronoi diagram is created from the points, and finally the diagram is edited to allow nonconvexity in the polygons. Voronoi diagrams are well-established mathematical objects. Our contribution is in the use of point processes to establish the initial points and in the editing algorithm performed on the diagrams. These two critical steps in our methodology are described below in detail.

Point processes

Voronoi diagrams depend entirely on the points used to create them; thus the selection of initial points is fundamental to the creation of a hypothetical landscape. We used *random point processes*, algorithms for creating random points. While many such algorithms exist with various properties (see Baddeley and Turner 2005), we settled on the use of four point processes: random uniform, cluster, simple sequential inhibition, and the lattice grid processes. Other processes were tried, but they were found to be redundant relative to the four methods chosen and they often produced landscapes with unpredictable characteristics. Explicit control over some aspects of the landscape, such as the number of units, is good to keep.

The baseline point process we used was the random uniform process, where x and y coordinates are determined by independent random draws from uniform distributions covering the feasible x and y ranges. This approach offers no control over point placement other than the bounds. The only control parameter for this method is the number of points to place.

A clustering process, where groups of points are placed in small groups, produces a landscape with high area CV because the polygons near the centers' clusters tend to be small. Initial trials used a Thomas clustering point process, where "parent" locations are chosen uniformly randomly, and then a random number of "child" points are created around each parent, with their displacement from the parent points determined by a bivariate Gaussian distribution. This introduced an unwanted level of variability in n, as the numbers of both parent and child points were determined by random draws from a Poisson distribution. Instead, we adapted the method by making the number of points per cluster an explicit parameter rather than a random variable and calculating an appropriate number of clusters based on the total number of points to place while preserving the Gaussian displacement from parent points.

To achieve the opposite extreme, i.e., landscapes with low area CV, we introduced an inhibition process, where points are placed uniformly randomly but never within a certain *inhibition distance* from another point. This produces a landscape with regularly sized polygons. Higher inhibition distance results in more evenly spaced points, which creates a more regular landscape, lowering the CV of unit area. Setting the inhibition distance to 0 makes this method equivalent to the random uniform process, which allows for clustering without actively creating clusters. This algorithm is commonly known as the *simple sequential inhibition* (SSI) point process (Baddeley and Turner 2005). McDill and Braze (2000) used a similar point process in MAKELAND.

When comparing the polygons created from these three point processes (uniform, cluster, and SSI) with the management unit boundaries in real forests, the lack of right angles was striking. In a Voronoi diagram based on points with coordinates drawn from continuous distributions, all border corners will be adjacent to three polygons, and there will be no weakly adjacent polygons. The only way for the corner of a border to be adjacent to four (or more) polygons is if it is equidistant from the four closest chosen points, an event that has probability 0 in any point process with continuous probability density. However, in actual landscapes with borders delineated by humans, right-angle intersections are not uncommon. By restricting point coordinates to a regular discrete scale, orthogonal border intersections become possible. We opted for the use of the *lattice grid* method that was specifically designed to allow for right-angled intersections of polygon borders. The lattice grid method overlays a grid over the feasible area and samples points from the lattice points of that grid. The numbers of horizontal and vertical grid lines are control parameters. In addition to right angles, this method tends to produce horizontal and vertical borders, reminiscent of how a forester might divide a forest section into management units in the absence of natural breaks. The lattice grid method works in conjunction with other processes to mimic a landscape that has natural breaks in management units as well as some imposed artificial boundaries.

The proposed landscape generator allows the use of the four-point processes outlined above both individually or in any combination. Figure 1 shows examples of Voronoi tessellations resulting from points used by each of the point processes individually. Using a mixture of these point processes allows for greater flexibility in the characteristics of the output landscape. F1

Editing the Voronoi tessellation

Landscapes produced by Voronoi tessellations have certain spatial properties that do not apply to real landscapes. While right angles of boundary intersections are imposed by using the lattice point process, additional differences remain. First, Voronoi landscapes are completely connected with no gaps or holes. Second, the resulting polygons are strictly convex. Since such restrictions are not present in real forest configurations, we edit the polygons of the Voronoi tessellation.

Polygon deletion creates holes in the landscape and opens the possibility of a disconnected forest, which occurs in real forests when a river or road cuts through the region. The final patchiness of the landscape is controlled by parameter p_H denoting the proportion of the polygons in the final landscape that are classified as holes. Deleted polygons that become holes in the final landscape are omitted from the calculation of other landscape statistics. Polygons are selected for deletion randomly with no preference given for size or adjacencies. When a polygon is deleted, the vertex degrees of all of its neighbors decrease by 1, effectively lowering the degree mean. The effects of polygon deletions on σ_d and area CV are negligible.

We address the convexity issue in a similar way: by deleting borders between two polygons and merging them into one. As in polygon deletion, a control parameter p_M is defined to denote the proportion of polygons in the final landscape that are products of merges. Borders are selected for deletion with uniform probability, thereby merging the polygons they had been separating. Merging is more complicated than simply deleting polygons because it is possible that more edges than those selected must be deleted for a merge to occur. For example, if polygon *A* is merged with polygon *B*, and *B* with *C*, the adjacencies must be checked so that if there is a border between *A* and *C*, it is also deleted. Both deleting and merging reduces the number of polygons; thus the number of points chosen initially, n_{tot}, is inflated from the target *n* so that the expected value of *n* for the final landscape matches the target after merges and deletions. For both edge and polygon deletion, the number of deletions is not random; it is determined explicitly from the control parameter. Only the selection of which item to delete is random. However, the nature of the merging prevents precise knowledge of the output *n*, as we cannot tell a priori (continuing the above example) whether the border between *A* and *C* will also be selected for deletion or if it must be deleted afterwards in the cleanup phase. Thus, merging introduces some randomness with respect to the final *n*.

Control

The previous section described how the new method creates landscapes; this section describes our method for producing landscapes with prespecified characteristics. While the proposed Voronoi tessellation algorithm is highly stochastic due to the many random processes used, the control parameters that are inputs for landscape creation are related to the characteristics of the output landscape. The control parameters relating to point processes are the proportion of points to be placed by each method, p_{unif}, p_{clust}, p_{SSI}, and p_{lat} (subject to the constraint that $p_{\text{unif}} + p_{\text{clust}} + p_{\text{SSI}} + p_{\text{lat}} = 1$), the number of horizontal and vertical gridlines for the lattice method, the inhibition distance for the SSI method, and the number of points per cluster and cluster spread for the cluster method. Control parameters not corresponding to point placement are the two editing parameters p_H and p_M, which set the proportion of polygons in the final landscape that are deleted or are the result of merging, respectively. As an example, a value of 0.5 for p_H instructs the algorithm to delete half of the polygons, while the same value for p_M instructs the algorithm to delete half of the common boundaries between polygons. The numbers of deletions and merges are determined exactly (rather than randomly with uniform probabilities p_H and p_M) to keep the variance of *n* small. The last control parameter is the horizontal to vertical aspect ratio of the entire landscape, *a*.

Extensive simulations were run to generate data from which we could learn how to choose control parameter values with a high probability of generating landscape of specified characteristics. (Data generation is described in the next section.) We considered three options for determining appropriate control parameter values from landscape characteristics. Two involve modeling explicit relationships: parametrically using linear models or nonparametrically using generalized additive models (GAMs). The third is to estimate the distributions of control parameters for specific landscape characteristics through a Monte Carlo simulation without making any assumptions or inferences about the relationships between the two. The third distribution-based approach has one major drawback: to provide good estimates, it relies on repeatedly estimating multivariate distributions, which, depending on the amount of data used, could be computationally expensive and rely on constant access to a large amount of data. In the modeling approach, relationships are determined from data that can then be discarded; subsequent evaluations is as easy as evaluating an expression. Given these considerations, we decided to start with the linear model approach.

Generating data

Preliminary analysis revealed that CV is strongly dependent on the point processes used, with the inhibition and lattice methods lowering CV and the cluster method elevating CV. Thus, to serve as a baseline, we simulated 500 landscapes using only the uniform

Fig. 1. Sample Voronoi tessellations of 100 polygons created using each of the four point processes. The locations of the initial points are marked with circles. Visually, the difference in area distributions is readily apparent: the lattice and inhibition methods produce regularly sized cells, the uniform method produces more variation, and the clustering method produces very high area variation. Also evident are the right angles and intersections of more than three borders produced exclusively by the lattice grid method.

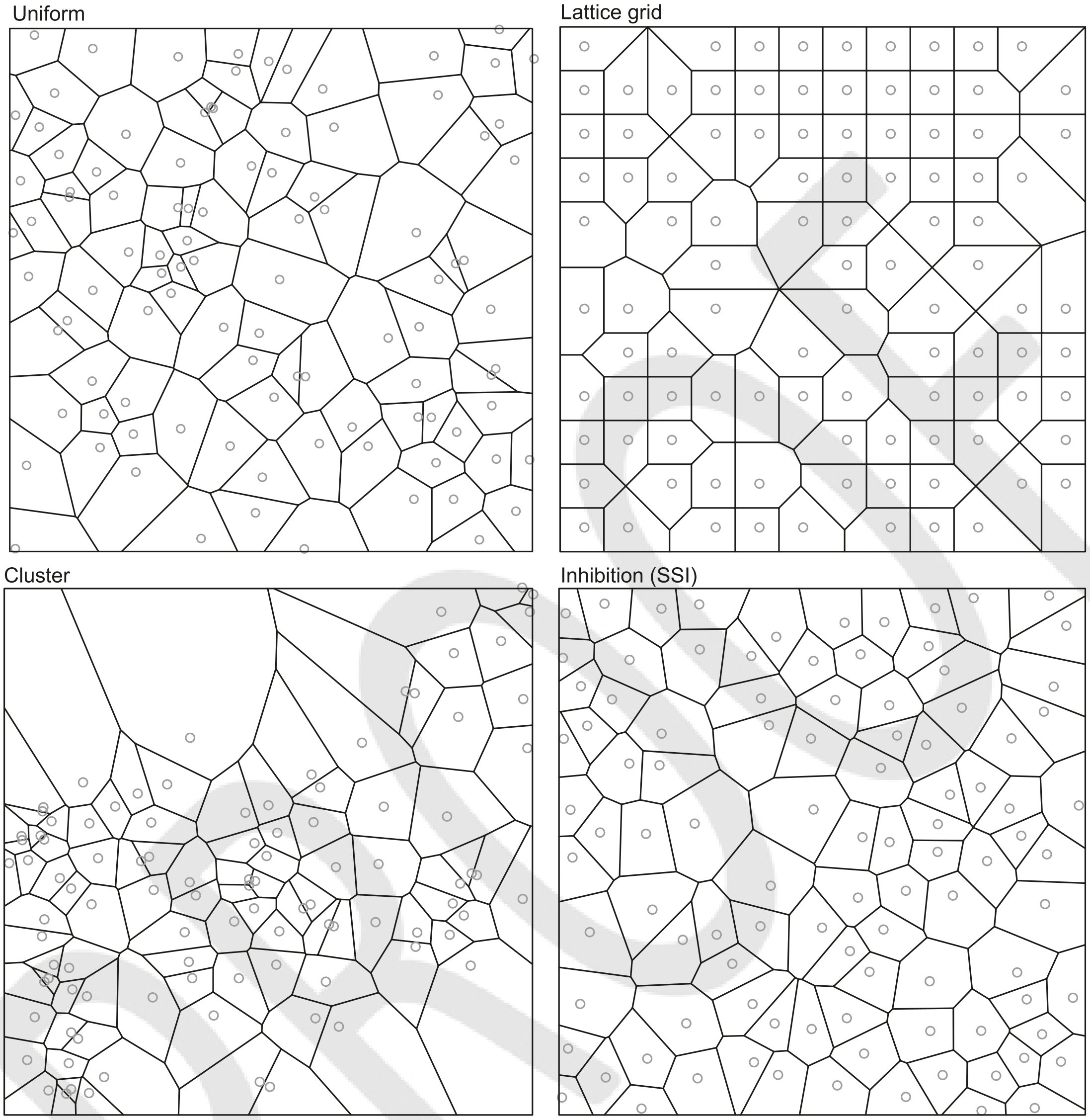


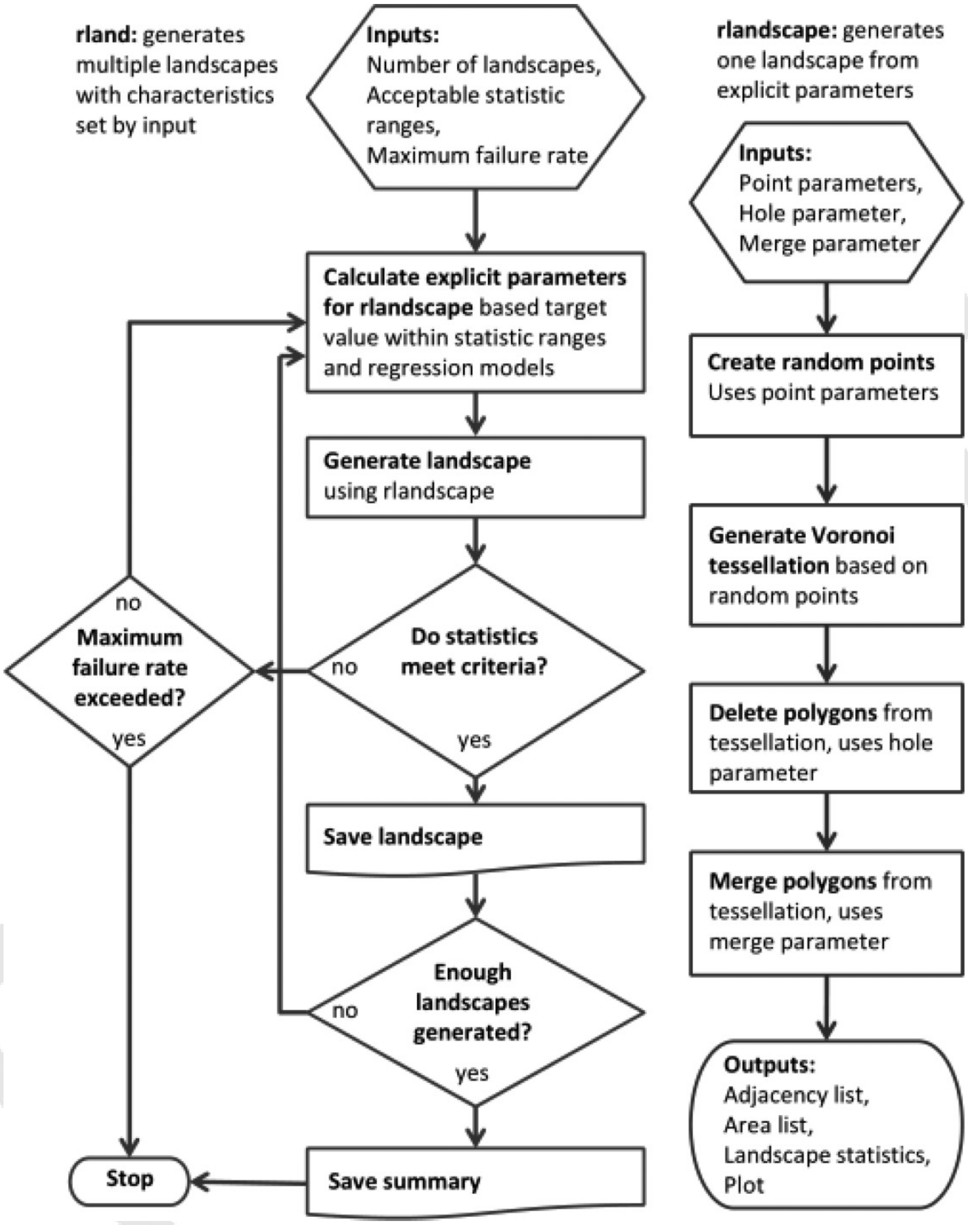
Table 1. Summary of parameters used in modeling.

Landscape characteristics	
n	Number of management units in the final landscape
CV	Coefficient of variation of the area distribution of the management units
μ_d	Mean of the management unit adjacency degree distribution
σ_d	Standard deviation of the management unit adjacency degree distribution
Control parameters	
n_{tot}	Total number of points to be generated in the random point patterns
a	Horizontal to vertical aspect ratio of the rectangle that the landscape is generated in. Specified or drawn randomly; used only as a predictor in the modeling process
p_{unif}, p_{clust}, p_{lat}, p_{SSI}	Proportions of points in the pattern generated by each method (must sum to 1)
p_H	“Hole fraction”, the proportion of polygons deleted from the initial tessellation
p_M	“Merge fraction”, the proportion of edges (postpolygon deletion) deleted from the tessellation
σ	Spread of points in the cluster method, i.e., the standard deviation of the isotropic Gaussian distribution of offspring points about their parents

Note: The landscape characteristics, along with a, are always independent predictors, while the control parameters are treated as dependent.

method (p_{unif} = 1) allowing p_H, p_M, and a to vary uniformly randomly (with p_H and p_M bounded above by 0.8 and a restricted between 1 and 5). Then, for each other point process, 50 000 landscapes were generated with points placed by a random mixture of that process and the uniform point process, with the mixing proportion varying from 0 to 1. Other parameters were varied uniformly randomly as above, except for some of the point process related control parameters that were given scale-dependent de-

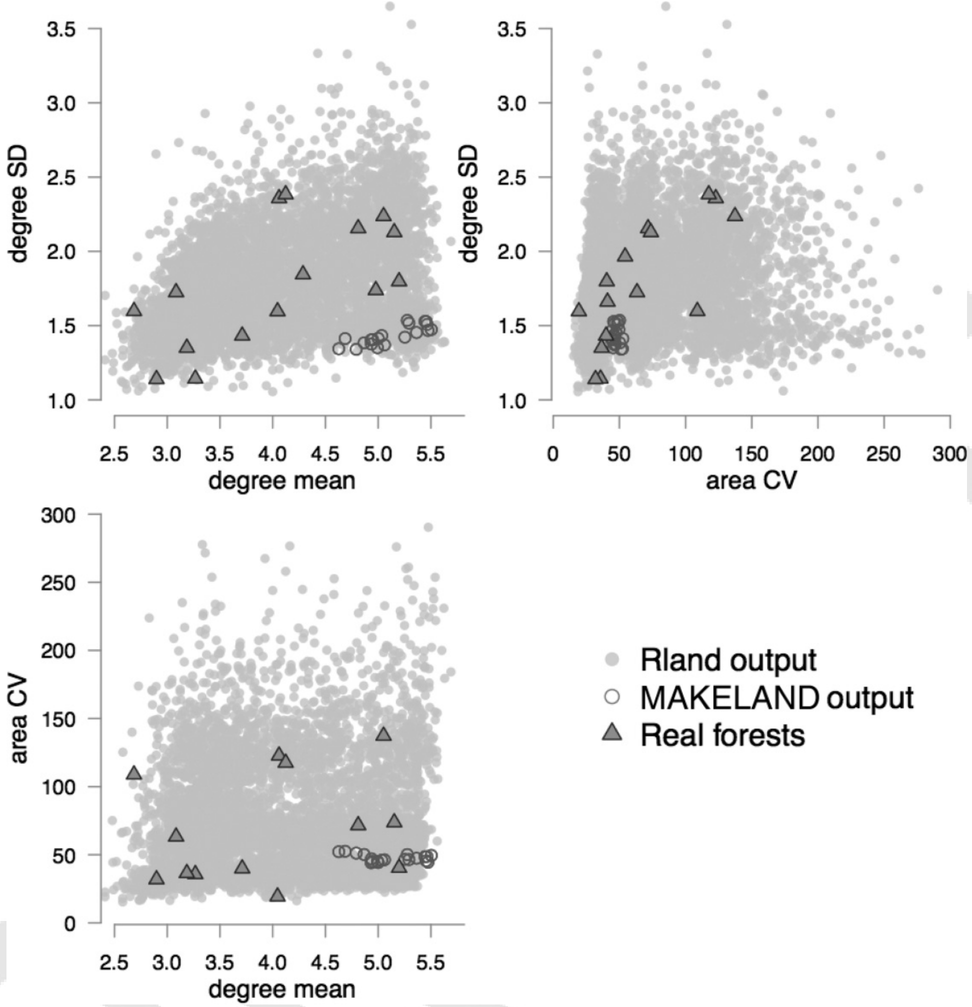
Fig. 2. This flowchart shows the approach taken to produce hypothetical forest landscapes with statistics that meet input criteria. The right column shows the flow of the rlandscape program, which generates a single landscape by editing a Voronoi tessellation created from points chosen using a mixture of random point processes. The left flow chart shows how the wrapper program rland estimates parameters for rlandscape based on input landscape statistics, tests the output for compliance, and saves a summary of the results when the run is complete (or aborts if a maximum allowable failure rate is exceeded).


rland: generates multiple landscapes with characteristics set by input
Inputs: Number of landscapes, Acceptable statistic ranges, Maximum failure rate
Calculate explicit parameters for rlandscape based target value within statistic ranges and regression models
Generate landscape using rlandscape
Do statistics meet criteria?
no
yes
Maximum failure rate exceeded?
no
yes
Save landscape
Enough landscapes generated?
no
yes
Save summary
Stop
rlandscape: generates one landscape from explicit parameters
Inputs: Point parameters, Hole parameter, Merge parameter
Create random points Uses point parameters
Generate Voronoi tessellation based on random points
Delete polygons from tessellation, uses hole parameter
Merge polygons from tessellation, uses merge parameter
Outputs: Adjacency list, Area list, Landscape statistics, Plot

faults (described below). These data formed the bases for sequential regressions to find each necessary model parameter based on the specified landscape characteristics.

To deal with bounding issues and to reduce dimensionality, we set default values for some of the point process related control parameters. This is necessary because the possible ranges for these variables depend on area (determined by a) and the number of points to place. For example, if the inhibition distance is set too high, it will be impossible to place enough points, and number of possible points placed by the lattice method is clearly limited by the total number of lattice points. To avoid confounding, in these two cases, we selected default values for the point process related control parameters, excepting the number of points placed by each method. The inhibition distance is set at $0.639\sqrt{a/n_{\mathrm{tot}}p_{\mathrm{SSI}}}$, where a is the horizontal to vertical aspect ratio of the entire landscape and $n_{\mathrm{tot}}p_{\mathrm{SSI}}$ is the number of points to be placed by the SSI method. This value was selected because in simulations attempting to place between 50, 100, 300, 500, and 1000 points in unit area with increasing inhibition distances, with each point–distance pair replicated 100 times, this was the (approximate) largest distance that resulted in all of the points being placed. For the lattice method, the number of horizontal and vertical grid lines, h and v, respectively, are set to defaults given by $h = \lceil\sqrt{n_{\mathrm{tot}}p_{\mathrm{lat}}/a}\rceil$ and $v = \lceil\sqrt{n_{\mathrm{tot}}p_{\mathrm{lat}} \times a}\rceil$, the minimum values such that $h_{\upsilon} \geq n_{\mathrm{tot}}p_{\mathrm{lat}}$ with the number of horizontal and vertical segments corresponding to the aspect ratio of the landscape. In the clustering process, preliminary results initially tended to increase the number of points per cluster to absurdly high levels when the CV was higher than average (often on the order of 50 points per

Fig. 3. Landscape statistics from 17 real forests (triangles), 20 landscapes generated by MAKELAND (open circles), and 500 landscapes generated by rlandscape (shaded circles) with targets varying over the range of the real forest statistics. Landscapes generated by rlandscape more than cover the characteristic space of the real forests. MAKELAND's output has somewhat varied μ_d, correlated with the small variation in σ_d; however, in this output the MAKELAND CV is static. The real forests are those with publicly posted adjacencies and areas on the University of New Brunswick's Integrated Forest Management Lab (http://ifmlab.for.unb.ca/fmos/datasets/).



cluster, which results in polygons of near 0 area at the cluster center). To prevent this, the default number of points per cluster is set to 5, with the spread left to be determined from the required landscape characteristics.

Modeling relationships

From a cause–effect viewpoint, it is the landscape characteristics that depend on the control parameters. However, to find appropriate control parameters, we turn that relationship on its head and use the characteristics as predictors in the linear models and the control parameters as responses. See Table 1 for a summary of the parameters involved. T1 Due to the difficulties of parameter and transformation selection in multivariate multiple regression, our strategy was to perform a series of sequential regressions beginning with a single control parameter as a response and all of the characteristics as predictors and then take each successive control parameter in turn, adding previously estimated parameters to the set of predictors. The exception is the aspect ratio a. Values of a that are not extreme (say $a < 10$) do not restrict the possible landscape characteristics, so rather than select an a, we ask the user to specify a value or range of values from which an aspect ratio will be drawn randomly. If an extreme value of a is desired, then this can also be specified. Also, as the mixing proportions for the point processes must sum to 1, selecting values for three of them determines the last.

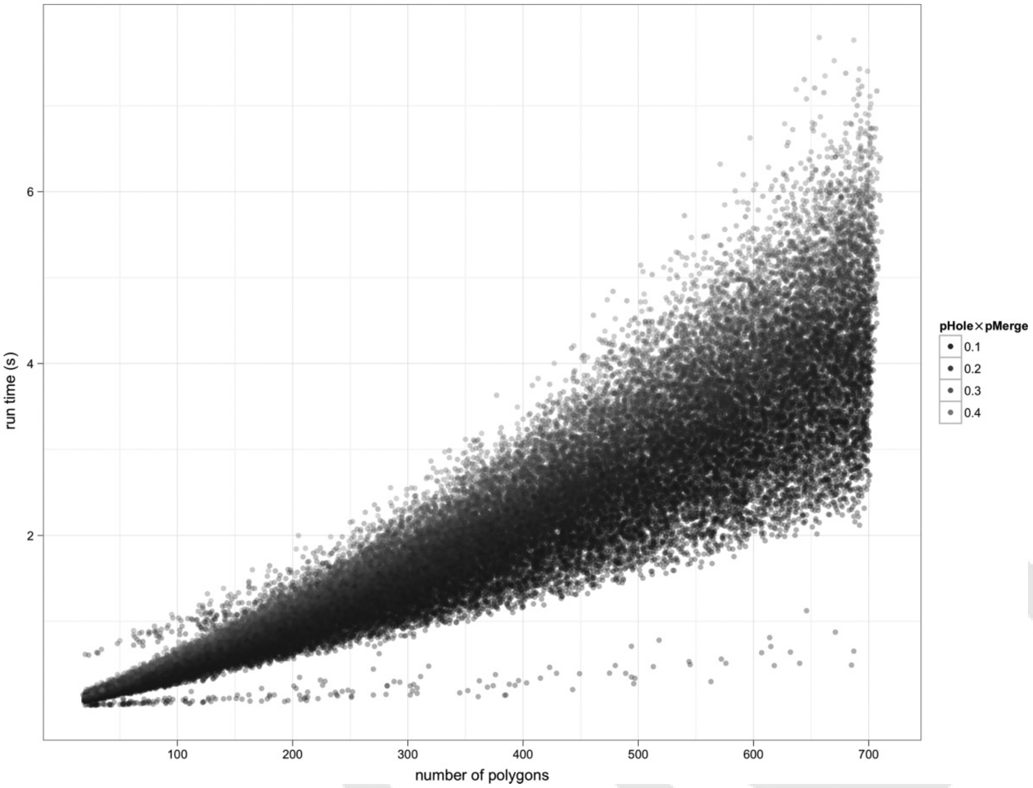
From N simulations, we have k vectors of predictors, one for each landscape characteristic (and one for the aspect ratio), $x_1, \ldots, x_k$, each vector of length N, and a similar set of m vectors of control parameters, $y_1, \ldots, y_m$. We then fit a linear model, with interaction terms and a Box–Cox transformation of the response, as necessary, to give $\hat{y}_1 = f_1(x_1,\ldots,x_k)$.

We selected candidate models using stepwise model selection based on Bayesian information criterion (BIC). The Bayesian information criterion was used rather than the Akaike information criterion due to the large sample size. Parameter and interaction term inclusion significance was evaluated using t tests. This process was then repeated for y_2 with the fitted values from f_1, $\hat{y}_1$ included in the set of predictors, and so on iteratively until all control parameters were estimated.

Producing specified landscapes

Mass landscape creation with specific targets is handled by two algorithms illustrated in Fig. 2. F2 Both are implemented in R (R Development Core Team 2011), making use of the Voronoi tessellation function provided in the deldir package (Turner 2012). We follow the naming convention of an "r" prefix to denote a

Fig. 4. Run time to produce a single landscape is plotted against the total number of polygons in that landscape, with shading showing the product of the hole proportion and merge proportion terms for about 45 000 simulations.



random generating function. The function rlandscape generates a single landscape from control parameters following the steps in the "Generating a landscape" section above. The second algorithm, rland, takes as inputs the number of landscapes to be produced and the acceptable bounds for their characteristics. It then picks specific targets for each landscape to be generated and uses the pregenerated linear models to select control parameters for these targets. Then, rland calls rlandscape to generate landscapes, tests to see if their characteristics are within bounds, and repeats this process until enough landscapes have been generated.

Testing

We have two goals to test. (1) Can rlandscape produce hypothetical landscapes with similar characteristics to actual forests? (2) Can rland select control parameters to meet specified characteristics? To test the first goal, we calculated characteristic statistics for landscapes with publicly posted adjacencies and areas on the University of New Brunswick's Integrated Forest Management Lab's website (http://ifmlab.for.unb.ca/fmos/datasets/). We then simulated 500 landscapes using rlandscape with control parameters varying randomly (the mixing proportions chosen uniformly between 0 and 1, then normalized, n between 20 and 1000, a between 1 and 5, and the individual point process parameters chosen from their feasible ranges). We also compared this output with landscapes created by MAKELAND. To test the efficiency, we recorded the number of tries needed and required time to produce 500 landscapes under a variety of specifications. All simulations were run on a 2.4 GHz Intel Core 2 Duo Macintosh with 6 GB of RAM.

Results and discussion

Comparison with real forests

F3 The results of the range test are shown in Fig. 3, along with the characteristics of 20 landscapes produced by MAKELAND. The figure makes clear that rlandscape exceeds the ranges seen in these real forests, whereas MAKELAND is constrained in both area CV and σ_d.

Timing

Figure 4 shows the relationship between run time and the number of polygons in a landscape. F4 Run time increases approximately linearly with n (at least for $n < 700$) with a large part of the variation explained by p_H and p_M. This is because they effectively increase the number of initial points placed and polygons that must be generated, e.g., a 100-polygon landscape with no holes or merges is 100 polygons pre- and postedit, but if $p_H = 0.5$, then 200 polygons must be generated so that 100 remain after 50% are deleted in the edit step, which takes longer to run. As implemented, we have successfully created landscapes of up to 10 000 polygons (see Fig. 5). F5 Capable of producing 1000-polygon landscapes in under 10 s, rlandscape works efficiently enough to be a practical landscape generator.

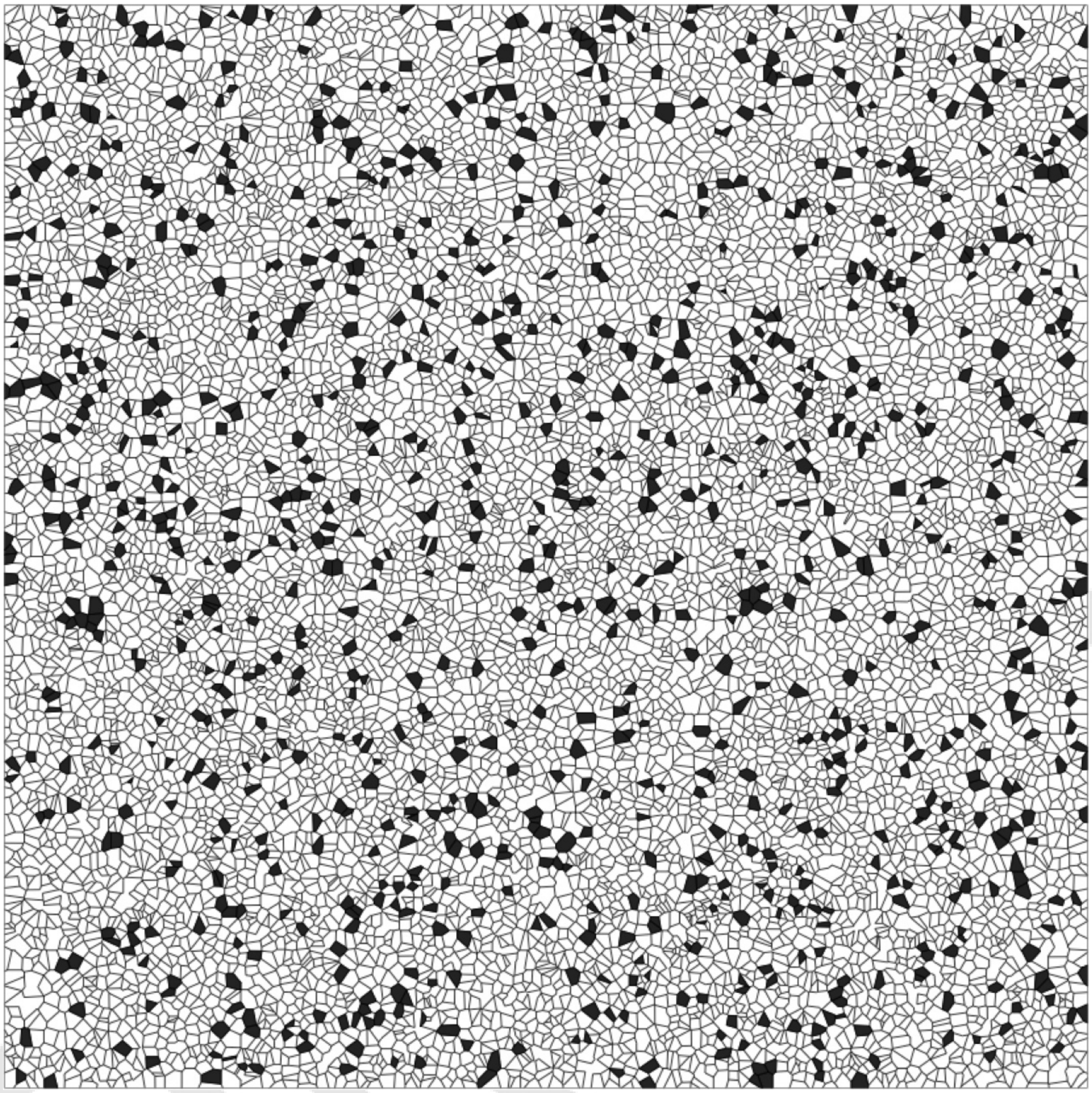
Control

Some of the relationships between control parameters and landscape characteristics can be readily intuited. For example, when a polygon is deleted, the vertex degrees of its neighbors all decrease by 1. As a result, μ_d will also decrease (Fig. 6). F6 As another example, the choice of point processes has a strong effect on the area CV as demonstrated clearly in Fig. 7. F7

Based on the baseline simulations, the random uniform point process produces landscapes with a distribution of CV with mean 56.1 and standard deviation 5.8 when no other point processes are used. We use this mean of the CV distribution as a "spline point" in regressions, fitting different models on either side to increase the proportion of cluster points if CV >56.1 or to increase the proportion of lattice and SSI points if CV <56.1.

As an example of the modeling results, the model for p_{lat} in the low-CV case is presented below. The proportion point process

Fig. 5. A landscape with 10 000 polygons, 25% placed by each point process. For this landscape, μ_d = 5.4, σ_d = 1.6, and CV = 42.9. This landscape took 199 s to generate. Black polygons indicate management units that have been deleted. Deleted edges are not printed at all.



parameters are the first estimated parameters, so they depend only on the characteristics; however, once they are estimated, they are added in as predictors for estimating the remaining parameters:

$$\begin{aligned}p_{\mathrm{lat}}^{2.6} = {} & \beta_0 + \beta_1\mathrm{CV} + \beta_2\mu_d + \beta_3\sigma_d + \beta_4 a + \beta_5\mathrm{CV}\mu_d + \beta_6\mathrm{CV}\sigma_d \\ & + \beta_7\mu_d\sigma_d + \beta_8 a\mathrm{CV} + \beta_9 a\mu_d + \beta_{10} a\sigma_d + \beta_{11}\mathrm{CV}\mu_d\sigma_d \\ & + \beta_{12} a\mathrm{CV}\mu_d + \beta_{13} a\mathrm{CV}\sigma_d + \beta_{14} a\mu_d\sigma_d + \beta_{15} a\mathrm{CV}\mu_d\sigma_d\end{aligned} \tag{1}$$

The β's from the above model are all significant at the 0.001 level, and the 2.6 power transformation of p_{lat} was determined by a Box–Cox test. The fitted coefficients are presented in Table 2. T2 This particular regression is unusual in that the aspect ratio is significant; it is insignificant for all of the other parameters. This is understandable because the geometry of the lattice grid depends on the aspect ratio, while all of the other point processes ignore it. The next sequential regression, for p_H, adds p_{lat} to the pool of predictors. We apply a similar series of multiple regressions to determine appropriate parameters for the high-CV case.

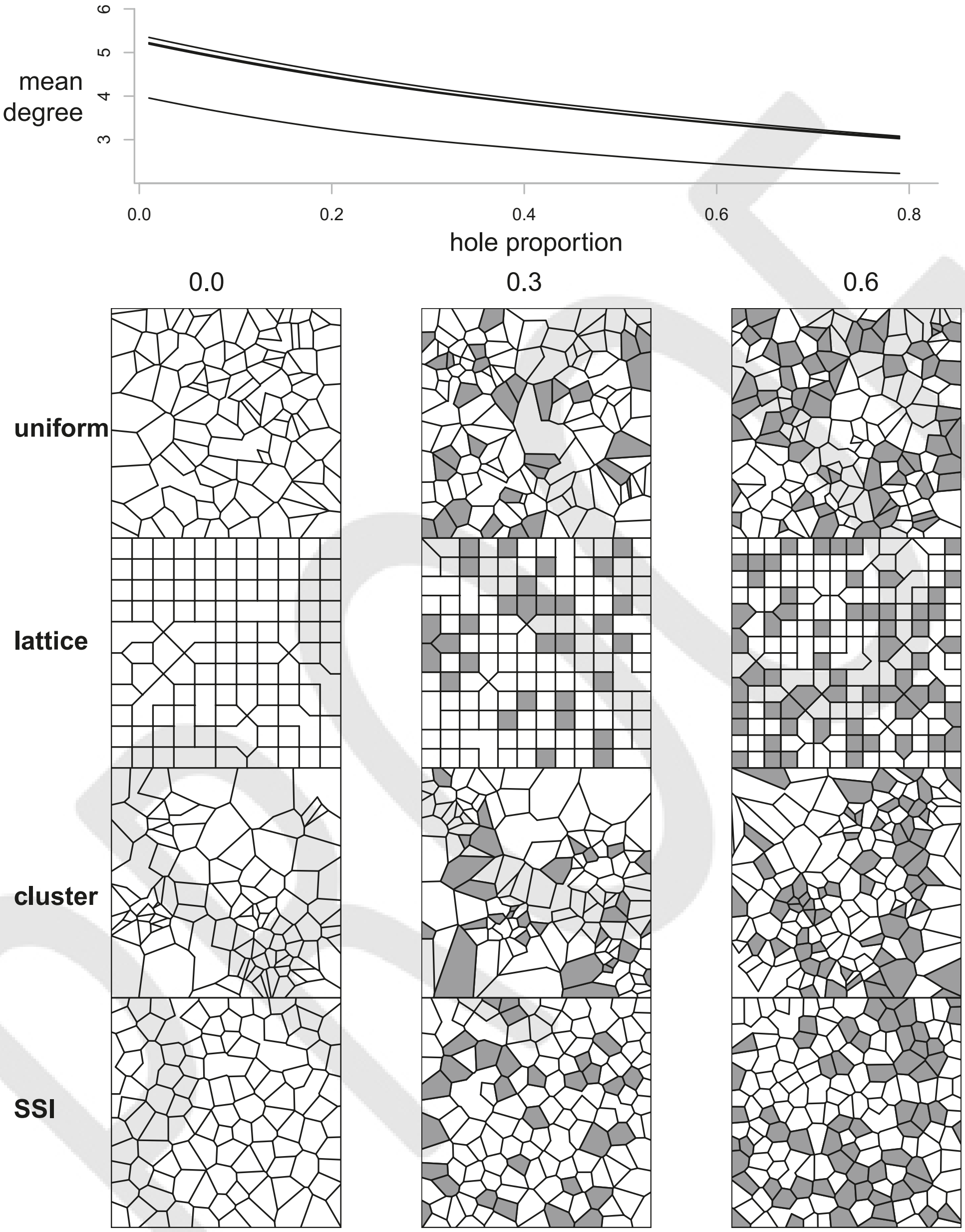
Meeting specifications

There is appreciable stochasticity built into rlandscape. To show this, 2000 landscapes of 100 polygons were generated using identical control parameters. The characteristics of the resulting landscapes were roughly normally distributed (summary statistics in Table 3). However, the regression methods are effective in T3 selecting control parameters to produce acceptable landscapes. Some results of the efficiency tests are shown in Fig. 8. For com- F8 mon characteristic values overlapping with observations of real forests, the efficiencies tend to be above 50%. At the extreme values of the ranges, there is higher variability, which results in lower success rates. This is particularly apparent in the case of high CV; however, the steep increases in time needed occur beyond the range of CV exhibited in real landscapes. High values of μ_d require many tries, but as p_H and p_M remain very low in these cases, the run time remains relatively short. Placing specifications on multiple characteristics compounds the efficiency reductions, typically resulting in efficiencies of between 10% and 20%. The parameter picking algorithm is calibrated for producing intermediate values efficiently; however, it could be adjusted to evaluate the specifications and change methods if they are outside certain thresholds.

Conclusions

We presented an article that accomplished two goals. First, we described a Voronoi tessellation based method that can produce varied, realistic forest landscapes. Second, we showed that the

Fig. 6. This figure shows how hole proportion can lower the degree mean of forest landscapes. Loess-smoothed degree means (μ_d) from 20 000 simulations are plotted in the top graph with sample landscapes below. The order of the lines in the top graph, from top to bottom, is uniform, cluster, SSI, and lattice. This ordering is consistent throughout the domain, with landscapes generated using the lattice method having noticeably lower degree means than landscapes generated from other point processes. This example uses landscapes generated using each point process exclusively to highlight the differences between them. In practice, landscapes will generally use a mixture of point processes. The shaded filled polygons indicate management units that have been deleted during the editing process.



new method can produce landscapes en masse with targeted spatial characteristics. The success of the first goal is demonstrated in Fig. 3, which shows that rlandscape output well covers the ranges of characteristics observed in real forests and beyond. The benchmarking shown in Fig. 8 suggests that rlandscape is viable for generating data sets of any feasible set of characteristics. The algorithms developed here have been uploaded as a package, rlandscape, to the Comprehensive R Archive Network (CRAN) and are available at http://cran.r-project.org/web/packages/rlandscape/index.html.

While we hope that rlandscape will be useful for a variety of forest planning or reserve selection applications, several modifications can be made to the proposed algorithm to enhance its utility. Incorporating support for weak adjacency constraints, as an example, could increase the set of optimization models that the output landscapes could test. Other modifications could alter the polygon deletion and merging processes to better mimic real forests or to improve fine control of the landscape characteristics. While we view the current, purely random implementation of these processes as an advantage, spatially correlating deletion probabilities could simulate discontinuities present in actual forests, like rivers and roads. Alternatively, the deletions and merges could target polygons that are outliers in degree or area to reduce or increase variability, which in turn would improve efficiency and precision in producing landscapes of targeted characteristics at the expense of randomness.

One caveat is that, due to the stochasticity, rland can have trouble mimicking regular landscapes. For example, some real forests

Fig. 7. An example of a pairs plot showing the relationship between pairs of variables. This particular plot demonstrates that the area CV statistic (examining the left column of plots) is strongly related to the cluster parameters. The correlation coefficients are displayed in the upper right half of the figure.

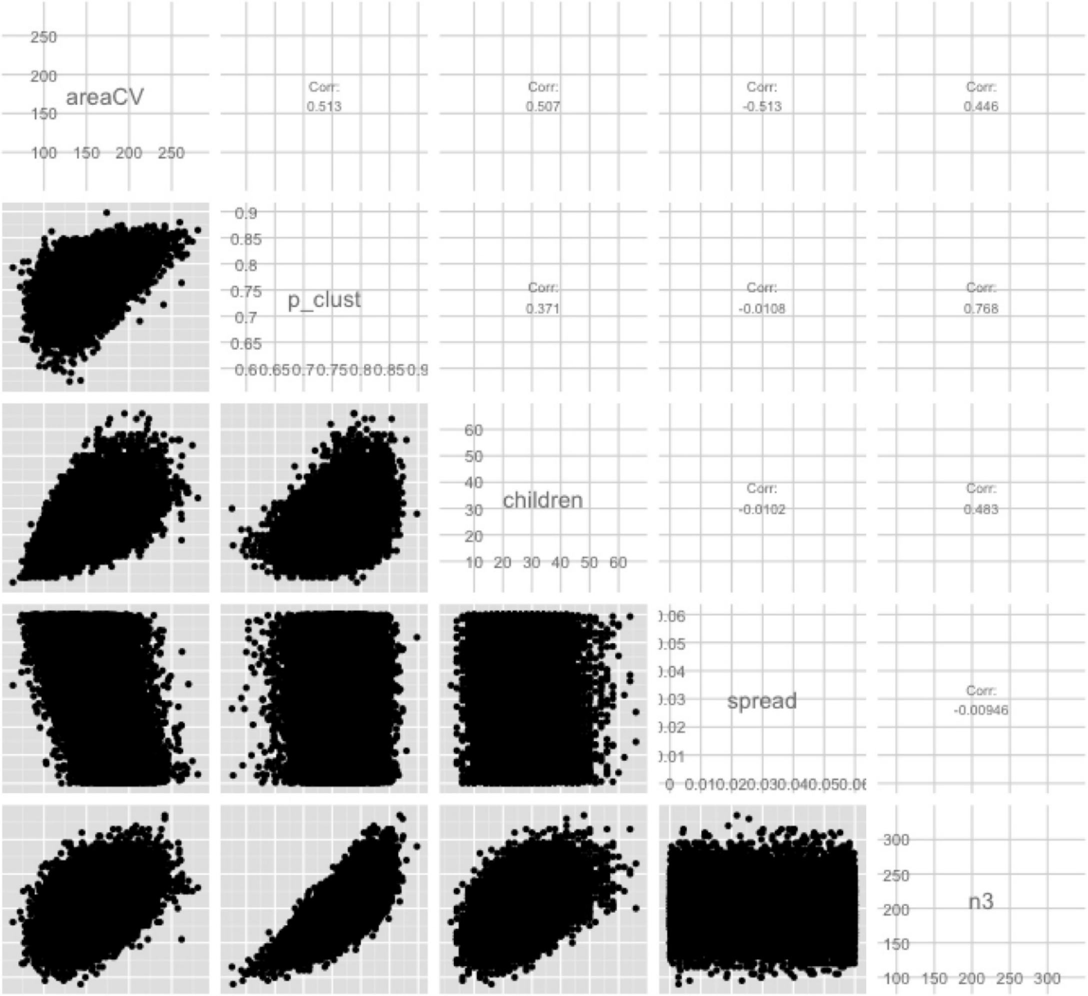


Table 2. Coefficients for estimating p_{lat}.

Index	Parameter	Coefficient (β value)
0	Intercept	0.00189
1	CV	0.02423
2	μ_{d}	0.34489
3	σ_{d}	0.79758
4	a	−0.12852
5	CV×μ_{d}	−0.01221
6	CV×σ_{d}	−0.02720
7	μ_{d}×σ_{d}	−0.21347
8	a×CV	0.00476
9	a×μ_{d}	0.03068
10	a×σ_{d}	0.08906
11	a×μ_{d}×σ_{d}	0.00693
12	a×CV×μ_{d}	−0.00092
13	a×CV×σ_{d}	−0.00330
14	a×μ_{d}×σ_{d}	−0.01927
15	a×CV×μ_{d}×σ_{d}	0.00066

Note: The predictors are not normalized, so the ranges of predictors must be considered when determining effect size. For example, CV usually ranges from 30 to 150, whereas μ_{d} ranges between 3 and 6, so the 0.024 coefficient for CV has a greater effect than the 0.345 for μ_{d}.

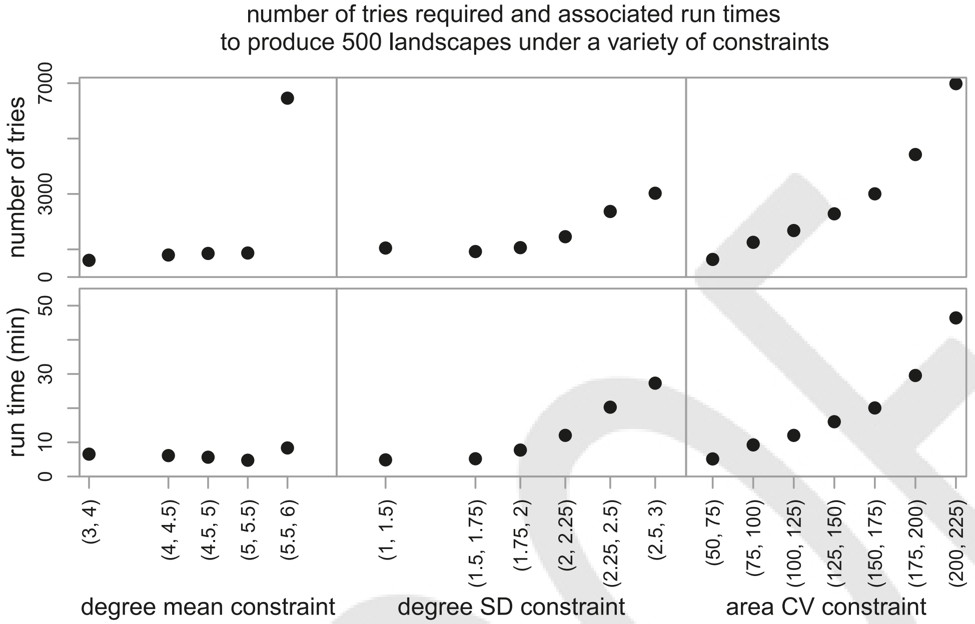
Table 3. Resulting statistics of 2000 landscapes of about 100 polygons each created by giving rlandscape the same input parameters.

Characteristic	Mean	SD
μ_{d}	4.86	0.11
σ_{d}	1.64	0.12
CV	69.1	8.0

Note: Not shown is the output n: of the 2000 trials, 1958 have n = 100 and the range is from 97 to 101.

are divided into equally sized stands such that the area CV = 0. Bypassing rland's automatic parameter selection, an intelligent human user can use rlandscape to mimic such a landscape by exclusive use of the lattice point process, which will drastically reduce the flexibility of the characteristic space — and care must be taken to specify appropriate control parameters. As a way around this, areas could be assigned post hoc using only the adjacencies from the generated landscape. The downside of this approach would be the lack of visual representation of the landscape. The current regression methods for choosing control parameters are sufficient, but using GAMs might yield more robust fits resulting in greater efficiency for producing landscapes of prespecified characteristics. For highly irregular landscapes, such as might occur in riparian areas or at a site with other geo-

Fig. 8. This shows the results of testing the time and number of tries necessary for rland to produce 500 landscapes under varying specifications. The number of polygons in these landscapes was held between 95 and 105, and other characteristics were set as indicated on the horizontal axis. Only one specification was imposed at a time; when μ_d was constrained, σ_d and CV were allowed to vary freely.



graphic features influencing the delineation of land into management units, we encourage users to again bypass rland and experiment selecting their own control parameters for rlandscape. Using the cluster process and adjusting the cluster spread toward 0 can result in landscapes with arbitrarily large CV, while values of p_H and p_M that are closer to 1 produce patchy landscapes with some very oddly shaped management units. While rlandscape can produce unconnected landscapes if p_H is high enough, users can construct their own by combining two or more smaller connected landscapes. These methods can produce a broad variety of landscapes, which we hope will be fully explored.

Acknowledgments

The authors would like to thank the editors and three anonymous referees whose comments greatly improved the manuscript.

References

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

Barrett, T.M. 1997. Voronoi tesselation methods to delineate harvest units for spatial forest planning. Can. J. For. Res. **27**: 903–930. doi:10.1139/x96-214.

Berman, P., and Fürer, M. 1994. Approximating maximum independent set in bounded degree graphs. *In* Proceedings of the Fifth Annual ACM—SIAM Symposium on Discrete Algorithms, SODA ‘94. Society for Industrial and Applied Mathematics, Philadelphia, Pa. pp. 365–371.

Bollobás, B. 1998. Modern graph theory. Springer Science+Business Media, Inc., New York.

Conrad, J.M., and Smith, M.D. 2012. Nonspatial and spatial models in bioeconomics. Nat. Resour. Model. **25**: 52–92. doi:10.1111/j.1939-7445.2011.00102.x.

Constantino, M., Martins, I., and Borges, J.G. 2008. A new mixed-integer programming model for harvest scheduling subject to maximum area restrictions. Oper. Res. **56**: 542–551. doi:10.1287/opre.1070.0472.

Du, Q., Faber, V., and Gunzburger, M. 1999. Centroidal Voronoi tesselations: applications and algorithms. SIAM Rev. **41**: 637–676. doi:10.1137/S0036144599352836.

Erdős, P., and Rényi, A. 1960. On the evolution of random graphs. *In* Publication of the Mathematical Institute of the Hungarian Academy of Sciences, Budapest, Hungary. pp. 17–61.

Goycoolea, M., Murray, A.T., Barahona, F., Epstein, R., and Weintraub, A. 2005. Harvest scheduling subject to maximum area restrictions: exploring exact approaches. Oper. Res. **53**: 490–500. doi:10.1287/opre.1040.0169.

Li, R., Bettinger, P., and Weiskittel, A. 2010. Comparisons of three different methods used to generate forest landscapes for spatial harvest scheduling problems with adjacency restrictions. Math. Comput. For. Nat.-Resour. Sci. Vol. 2: 53–60.

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

McDill, M.E., Rebain, S.A., and Braze, J. 2002. Harvest scheduling with area-based adjacency constraints. For. Sci. **48**: 631–642.

McNaughton, A.J., and Ryan, D. 2008. Adjacency branches used to optimize forest harvesting subject to area restrictions on Clearfell. For. Sci. **54**: 442–454.

Murray, A.T. 1999. Spatial restrictions in harvest scheduling. For. Sci. **45**: 45–52.

Önal, H., and Briers, R.A. 2006. Optimal selection of a connected reserve network. Oper. Res. **54**: 379–388. doi:10.1287/opre.1060.0272.

Petroski, J.R. 2006. Treatment unit delineations for spatial planning on Pennsylvania state forest land. Master of Science, Pennsylvania State University, University Park, Pa.

R Development Core Team. 2011. R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Rebain, S.A., and McDill, M.E. 2003. A mixed-integer formulation of the minimum patch size problem. For. Sci. **49**: 608–618.

Tóth, S.F., and McDill, M.E. 2009. Finding efficient harvest schedules under three conflicting objectives. For. Sci. **55**: 117–131.

Tóth, S.F., McDill, M.E., Könnyű, N., and George, S. 2012. A strengthening procedure for the path formulation of the area-based adjacency problem in harvest scheduling models. Math. Comput. For. Nat.-Resour. Sci. **4**: 1–23.

Turner, R. 2012. deldir: Delaunay triangulation and Dirichlet (Voronoi) tessellation. R package version 0.0-19. Available from http://CRAN.R-project.org/package=deldir.

Wyszomirski, T., and Weiner, J. 2009. Variation in local density results in a positive correlation between plant neighbor sizes. Am. Nat. **173**: 705–708. doi:10.1086/597612.