The use of cellular automaton approach in forest planning

Tero Heinonen and Timo Pukkala

Abstract: This study presents an optimization method based on cellular automaton (CA) for solving spatial forest planning problems. The CA maximizes stand-level and neighbourhood objectives locally, i.e., separately for different stands or raster cells. Global objectives are dealt with by adding a global part to the objective function and gradually increasing its weight until the global targets are met to a required degree. The method was tested in an area that consisted of 2500 (50 × 50) hexagons 1 ha in size. The CA was used with both parallel and sequential state-updating rules. The method was compared with linear programming (LP) in four nonspatial forest planning problems where net present value (NPV) was maximized subject to harvest constraints. The CA solutions were within 99.6% of the LP solutions in three problems and 97.9% in the fourth problem. The CA was compared with simulated annealing (SA) in three spatial problems where a multiobjective utility function was maximized subject to periodical harvest and ending volume constraints. The nonspatial goal was the NPV and the spatial goals were old forest and cutting area aggregation as well as dispersion of regeneration cuttings. The CA produced higher objective function values than SA in all problems. Especially, the spatial objective variables were better in the CA solutions, whereas differences in NPV were small. There were no major differences in the performance of the parallel and sequential cell state-updating modes of the CA.

Résumé : Cet article présente une méthode d'optimisation basée sur un automate cellulaire (AC) pour résoudre des problèmes de planification forestière dans l'espace. L'AC maximise localement les objectifs pour le peuplement et le voisinage, c.-à-d., séparément pour différents peuplements ou trames de cellules. Les objectifs globaux sont pris en compte en ajoutant une partie globale à la fonction objectif et en augmentant progressivement sa pondération jusqu'à ce que le niveau requis des cibles globales soit atteint. La méthode a été testée pour une superficie constituée de 2500 (50 × 50) hexagones de 1 ha. L'AC a été utilisé à la fois avec des règles de mise à jour parallèle et séquentielle. La méthode a été comparée à la programmation linéaire (PL) pour quatre problèmes de planification forestière non spatiale où la valeur actualisée nette (VAN) assujettie à des contraintes de récolte était maximisée. Les solutions de l'AC correspondaient à au moins 99,6 % des solutions obtenues avec la PL pour trois des problèmes et à 97,9 % pour le quatrième problème. L'AC a été comparé au recuit simulé (RS) pour trois problèmes à caractère spatial pour lesquelles une fonction d'utilité à objectifs multiples était maximisée en respectant des contraintes de récolte périodique et de volume final. L'objectif de nature non spatiale était la VAN et les objectifs de nature spatiale étaient le regroupement des vieilles forêts et des aires de coupe, ainsi que la dispersion des coupes de régénération. Pour tous les problèmes, le résultat de la fonction objectif avait une valeur plus élevée avec l'AC qu'avec le RS. En particulier, les variables liées aux objectifs de nature spatiale étaient meilleures dans les solutions de l'AC alors que les différences de la VAN étaient minimes. Il n'y avait pas de différence importante entre la performance des règles de mise à jour parallèle et séquentielle de l'état de la cellule utilisées par l'AC.

[Traduit par la Rédaction]

Introduction

The purpose of forestry planning is to maximize the objective function specified by the forest owner. One step of planning is to prepare a situation-specific model that describes the production potential of the forest, on one hand, and the preferences of the forest owner, on the other hand. The success of planning depends, among other things, on how precisely the planning model represents the goals and expectations of the forest owner and the relationships between different inputs and outputs of the production system. Management objectives of the forest may be related to individual forest stands, aggregates of stands, or all of the stands. Therefore, the planning model should be able to accommodate objectives representing any of these three geographical scales.

Forest planning aims at finding such treatment schedules for the stands that maximize the objective function of the forest owner. Numerical optimization methods are commonly employed in this task. Stands are homogenous forest areas that differ from adjacent areas in site or stand characteristics. Forest is a combination of stands. In stand-level planning, the analyses are conducted one stand at a time, and in the absence of forest-level goals or constraints, stand-level optima also result in the forest-level optimum. Stand-level management objectives have traditionally been related to economic benefits such as net present value (NPV). Maximal wood production is also a typical stand-level objective, but stand-level objectives may also be noneconomic such as stand diversity or amenity.

The importance of including spatial aspects in forest planning is well established (e.g., Murray and Church 1995; Bas-

Received 6 June 2006. Accepted 30 April 2007. Published on the NRC Research Press Web site at cjfr.nrc.ca on 20 November 2007.

T. Heinonen[1] and T. Pukkala. Faculty of Forestry, University of Joensuu, P.O. Box 111, 80101 Joensuu, Finland.

[1]Corresponding author (e-mail: tero.heinonen@joensuu.fi).

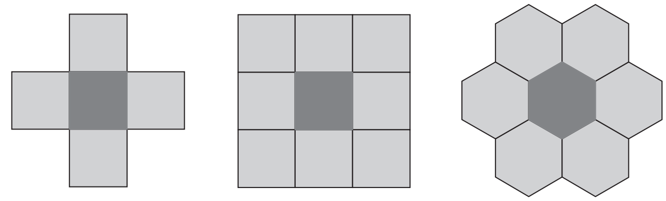
kent and Jordan 2002), mainly because of increasing importance of ecological objectives. A common spatial objective has been to decrease the fragmentation of the forest (Öhman 2000; Öhman and Eriksson 2002). In North America, so-called green-up constraints have frequently been included in optimization problems (e.g., Brumelle et al. 1997; Boston and Bettinger 2001). Cutting area aggregation aiming at scale benefits is an example of economic spatial objectives (Öhman and Lämås 2003). Spatial objectives require consideration of the relative locations of stands in optimization calculations. Spatial objectives make the ranking of the treatment alternatives of a stand dependent on the characteristics and treatments of the neighbouring stands. This means that the analyses for a stand must include the neighbouring stands as well.

The third category of management objectives requires the planning model to simultaneously consider all stands of the forest. The requirement for even flow of timber is an example of these objectives. Other examples are the requirement for a minimum or maximum total harvest, maximum total regeneration area, and a minimum growing stock volume or asset value at the end of the planning period. Examples of ecological forest-level objectives are a certain total area or percentage of an important habitat type (Öhman 2000; Kurttila et al. 2002). With forest-level objectives, the task of planning is to find such a combination of treatment schedules for stands that meets the forest-level objectives.

The most frequently used numerical techniques in stand-level optimization have been nonlinear programming (e.g., Roise 1986; Pukkala and Miina 1997) and dynamic programming (e.g., Torres-Rojo and Brodie 1990; Bettinger et al. 2005). Problems with forest-level objectives have traditionally been solved with linear programming (LP). If individual stands must not be partitioned, integer programming and mixed integer programming can be used. However, spatial objectives often make the use of LP impractical. Integer programming and mixed integer programming can be used to solve small spatial problems, but real-world problems can be very complex and large (Murray and Church 1995; Bettinger et al. 1999). Heuristic techniques can deal with nonlinear relationships and a large amount of decision variables, but they cannot guarantee the global optimum. The most commonly used heuristics applied to spatial forest planning have been simulated annealing (SA) (Lockwood and Moore 1993; Baskent and Jordan 2002), tabu search (Bettinger et al. 1997), and genetic algorithms (Bettinger et al. 2002). LP can be used in combination with the heuristics (Tarp and Helles 1997; Öhman and Eriksson 2002), and hybrid methods can also be composed of different heuristics (Boston and Bettinger 2002; Nalle et al. 2002). Dynamic programming has also been applied to solve forest level problems that involve adjacency constraints (Hoganson and Borges 1998).

One method that has not yet been applied much to spatial forest planning is cellular automaton (CA) (Strange et al. 2002, Mathey et al. 2005, 2007). CA are decentralized computing methods based on self-organizing systems capable of describing complex systems with simple rules. CA were first proposed by von Neumann and Ulam in the 1950s (Von Neumann 1966). CA typically consist of square-shaped cells forming a regular grid or tessellation, each cell having a finite number of possible states. Usually the state set is the same for all cells. If cells can have different state sets, the CA is called polygeneous (Sarkar 2000). A single cell changes its state following a rule (local rule) that depends on the neighbourhood of the cell. The neighbourhood of a cell is usually a set of cells that interact with the given cell. The most common neighbourhoods used with two-dimensional grids are the von Neumann and Moore neighbourhoods (Wolfram 1984). The von Neumann neighbourhood consists of adjacent cells on the right, left, above, and below. The Moore neighbourhood comprises also the diagonal cells. If the CA is composed of hexagons, the neighbourhood is called a hexagon neighbourhood and it consists of six immediate adjacent cells (Fig. 1).

Fig. 1. The most common cell neighbourhoods in cellular automata. From the left: von Neumann neighbourhood, Moore neighbourhood, and hexagonal neighbourhood.



The dynamics of a CA are generated by repeatedly applying the local rule to all of the cells on the grid. This can be carried out in a number of different ways. With the classical, synchronous, or parallel updating method, all cells are evaluated and they change state simultaneously. With asynchronous or sequential updating, the cells are evaluated one after another. Asynchronous updating requires the specification of the order in which the cells are considered. The simplest asynchronous updating method is to put the cells of the grid in a predefined, fixed order to form a sequence. The cells can be evaluated for example line by line (Schönfisch and de Roos 1999). Asynchronous updating can also be implemented in random ordering. Asynchronous CA are not CA in the classical (von Neumann) sense.

CA were originally developed as models of self-reproducing organisms, and computation-theoretic questions were considered important. Since those days, the CA have gained interest in many different fields. A review of topics related to computer science and mathematics can be found in Sarkar (2000). The most widely explored application of CA is modelling different physical systems such as spin systems (Creutz 1986). Pattern recognition has also been a popular field of study (Raghavan 1993). In landscape planning, CA have been used to forecast the effect of urban growth on habitat pattern (Syphard et al. 2005), simulate landscape dynamics in an Amazonian colonization area (Soares-Filho et al. 2002), and simulate the spread of urban areas (Barredo et al. 2003; Ward et al. 2003). CA have also been used in ecology to study the functioning and structure of ecological systems (Keymer et al. 1998). Fire spread (Hargrove et al. 2000; Li and Magill 2001) and road traffic (Maerivoet and De Moor 2005) have also been modelled with CA. The above examples are just a small sample of the broad list of different application areas of CA.

Strange et al. (2001, 2002) were the first to apply CA in spatial land allocation problems in forestry. They used CA to optimize land use in afforestation areas in Denmark. Meilby et al. (2001) included global constraints in the problem when allocating recreational and forest areas. Mathey et al. (2005) presented a simple CA capable of solving landscape management problems with soft global constraints. In Mathey et al. (2007), even-flow constraints were taken into account when maximizing the cumulative harvest volume and the amount of clustered late-seral forest. The CA approach has also been used to study the forest stand structure and different indices describing it (Lett et al. 1999; Pommerening 2006). Sprott et al. (2002) modelled patterns of natural forest and predicted its evolution using CA. The formation of banded vegetation patterns in Sub-Antarctic forest was simulated in the study of Puigdefábregas et al. (1999).

CA have many features that make them suitable for stand-level and spatial optimization. CA are based on localized computing units and neighbourhood relationships. The implication of this property is that when CA are used in forest planning, stand-level problems are solved instead of a forest-level problem, but the stand-level optima may depend on the neighbouring stands. If the planning problem has forest-level objectives, additional methods must be developed to tie the stand-level problems together.

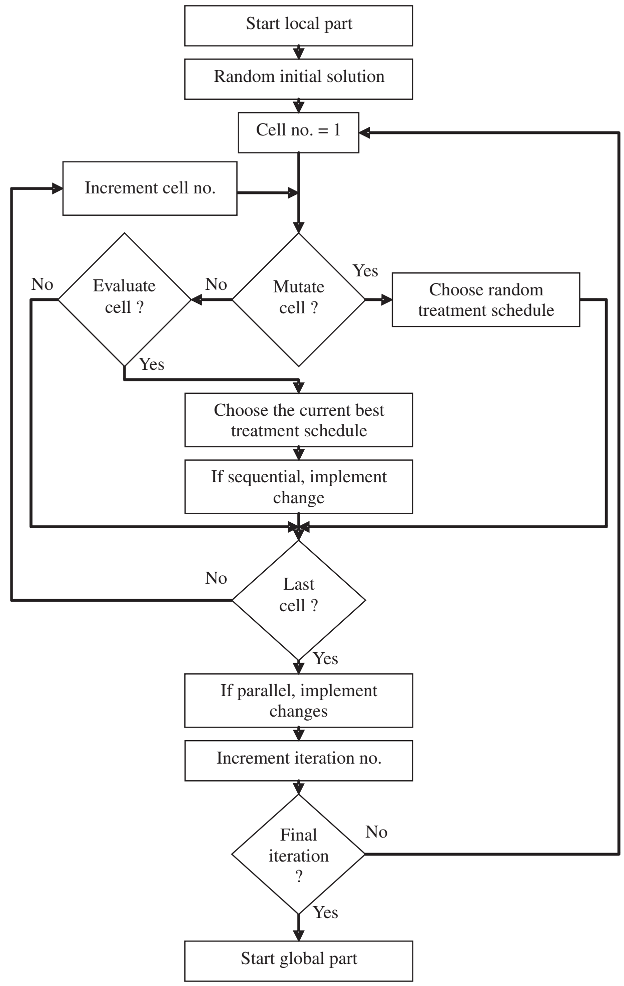
The aim of the study was to develop and test a two-step CA heuristic in forest planning. This method should be able to deal with cell- or stand-level goals, neighbourhood-related goals, and global goals. The first step of the algorithm maximizes stand-level objective functions. The second step forces the solution to meet the forest-level goals or constraints. The method should be competitive with the traditional "centralized" heuristics and it should work with both cell- and vector-based stand data.

The next section introduces the reader to the CA used and developed in this study. The Results section begins with the presentation of the optimal parameters of the automaton. Then, the performance of the optimized automaton is compared with LP in nonspatial problems and with SA in spatial problems. The final section discusses the results and provides some ideas for further modification and improvements of the automaton.

Description of the system

The starting point of our algorithm was the one proposed by Strange et al. (2002) for optimizing the combination of land uses of raster cells. In the algorithm of Strange et al. (2002), mutations and innovations occurring with decreasing probability change the solution. In our case, a treatment schedule of a stand or a cell corresponds to a land use alternative of a cell in Strange et al. (2002). The treatment schedules are produced prior to optimization. First, a random treatment schedule is selected for every cell (or stand), from among the set of previously produced schedules (Fig. 2). Then the first cell is considered and a random number ($U(0,1)$) is drawn from a uniform distribution. If the random number is smaller than the current mutation probability, a random schedule of the same cell replaces the current schedule, i.e., a mutation occurs. If there is no mutation, another uniform random number is generated and compared with the current innovation probability. If innovation occurs, the best (i.e., locally optimal) schedule is searched for the cell, and it will replace the current schedule. Then the next cell is inspected in the same way. Once all cells have been inspected, an iteration

Fig. 2. Flowchart describing the local part of the cellular automaton.



is completed, and a new iteration is begun with the first cell. Mutation and innovation probabilities are updated before starting the new iteration. This is repeated for a predefined number of iterations or until no changes can be identified during an iteration or a certain number of iterations.

In the algorithm of Strange et al. (2002), the cells' treatments change in a parallel way. When an innovation seeks for the best alternative for a cell, it does not know how the neighbouring cells are going to change. We implemented the algorithm in both parallel and sequential ways. In the parallel mode, all of the mutations and innovations that are identified during iteration are made simultaneously at the end of iteration, whereas the sequential mode executes the changes immediately with a consequence that, when a certain cell is inspected, it is known how the previous cells would change during the same iteration.

Parameters

The search process of our CA is controlled through six parameters: initial mutation probability, a change parameter for mutation probability, initial innovation probability, a change parameter for innovation probability, total number of iterations, and search mode (parallel versus sequential). The probability of mutation depends on the initial probability, total number of iterations, and current iteration number:

$$P_M = P_M^0 (1 - t/T)^{\tau_M} \quad [1]$$

where P_M^0 is the initial probability of mutation, t is the current iteration number, T is the total number of iterations, and τ_M is an exponent greater than or equal to zero. The probability of innovation is calculated in the same way:

$$P_I = P_I^0 (1 - t/T)^{\tau_I} \quad [2]$$

where P_I^0 is the initial probability of innovation and τ_I is an exponent greater than or equal to zero.

Local objective function

When innovation occurs, the best treatment schedule is selected for the cell (or stand). Alternative schedules of cell k are ranked with the following objective function:

$$U_{jk} = \sum_{i=1}^{I} w_i u_i (q_{ijk}), j = 1, ..., n_k \quad [3]$$

where U_{jk} is the objective function value of schedule j of cell k, I is the number of objectives, w_i is the weight of objective i, u_i is a priority function for objective i, q_{ijk} is the quantity of objective variable i in schedule j of cell k, and n_k is the number of alternative management schedules in cell k. The priority functions scale the objective variables between 0 and 1:

$$u_{ijk} = \frac{q_{ijk} - q_i^{\min}}{q_i^{\max} - q_i^{\min}} \quad [4]$$

where $q_i^{\min}$is the smallest and $q_i^{\max}$ the largest value of the objective variable among all schedules of all cells. If the smallest possible value is zero, which is often the case, the priority is simply $u_{ijk} = q_{ijk}/q_i^{\max}$.

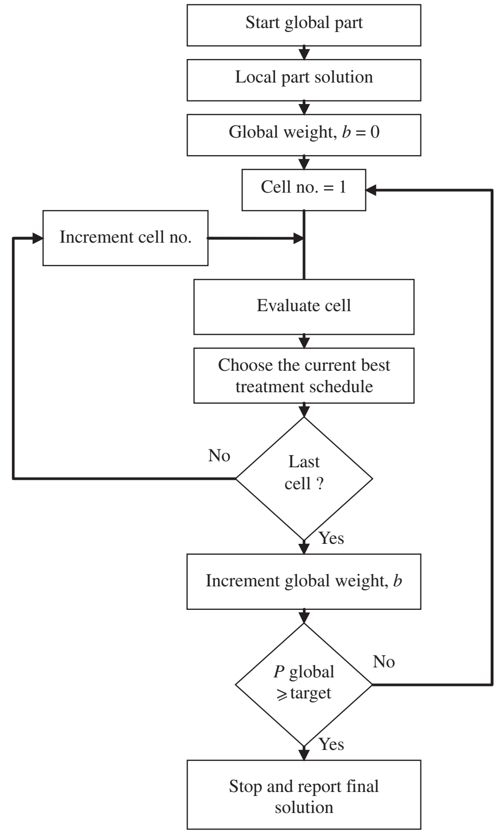
Managing global objectives and constraints

The values of local objective variables depend on the cell only or the cell and its neighbourhood. In forestry planning, there are often goals and constraints that cannot be met by allowing the cells or stands to optimize, even when the cell-level optima depend on adjacent cells or other neighbourhoods. Therefore, an additional phase was added to the algorithm, the purpose of which is to guarantee that the global objectives and constraints are met to a required degree. In this part of the algorithm, alternative schedules are evaluated using a function that has both a local and a global component:

$$R_{jk} = \frac{a_k}{A} U_{jk} + bP, j = 1, ..., n_k \quad [5]$$

where R_{jk} is the total priority of alternative j of cell (or stand) k, a_k is the area of cell k, A is the total area of cells, b is the weight of the global priority function, and P is the global priority of the combination of the cells' treatment schedules. If stands are used instead of cells, multiplier a_k/A makes the relative weight of local objectives dependent on stand area, which is a useful property. The global priority function measures the priority of the combination in terms of global objectives and constraints such as the total harvest of a period or total habitat area. The global priority function is as follows:

Fig. 3. Flowchart describing the global part of the cellular automaton.



$$P = \sum_{l=1}^{L} v_l p_l (g_l) \quad [6]$$

where P is the global priority, L is the number of globally evaluated objectives, v_l is the weight of global objective l ($\Sigma v_l = 1$), p_l is a priority function for objective l, and g_l is the quantity of objective variable l. The priority functions convert the values of criterion variables into priority values ranging from 0 to 1. As the weights (v_l) are scaled so that their sum equals 1, the maximum possible global priority is also 1. Therefore, a global priority equal to 1 means that all the global objectives and constraints are fully met.

The global part of the algorithm begins with the ending solution of the local optimization (Fig. 3). All schedules of all cells are sequentially evaluated for many iterations using eq. 5, and a better schedule always replaces the current schedule. The initial weight of the global priority function is zero, from which it is gradually increased until the global priority reaches a predefined value. The search also stops when a predefined total number of iterations have been completed. This latter stopping criterion is required for cases in which the target value of the global priority is infeasible. The global part of the algorithm makes those changes first, which improve the global priority most without notably impairing the local priority. As a result, the global objectives are achieved with a minimum possible loss in the local objectives. The global part has three parameters: step of changing the weight of global priority, target value for global priority, and maximum number of iterations.

Materials and test problems

Planning area and cell states

The test forest was a hexagonal grid formed by a tessellation of regular hexagon cells. Hexagons were used to avoid single points of contact between neighbouring cells characteristic of square grids. The cell neighbourhood was therefore a hexagonal neighbourhood, and the state of a single cell at each time step was affected only by adjacent cells. For the cells on the edge of the grid, the neighbourhood was not completely defined. The undefined part of the neighbourhood was assumed to have similar properties as the defined part. The grid consisted of 2500 cells, 50 cells in a row and column. Each cell was 1 ha in size. The forest data for the cells were taken from actual forest stands located in North Karelia, eastern Finland. The forest data were imported to the Monsu forest planning software (Pukkala 2004), which was used as the platform for the algorithms used in this study.

Monsu's automatic simulation tool was used to produce alternative treatment schedules for the cells. These treatment schedules represent different states possible for a cell. A regeneration cut with the necessary postcutting treatments was simulated when the stand age reached either the minimum regeneration age or the minimum mean diameter required for a regenerative cut. Thinning was simulated when the stand basal area reached the so-called thinning limit. All cuttings were simulated in the middle of the 20-year time periods. The simulations were based on the official silvicultural guidelines of Finland, but the timing of thinnings and regeneration cuttings was varied to obtain more than one treatment schedule per cell. In addition, one of the simulated treatment alternatives for mature stands was always the "no treatment" option. Obviously, altering the original idea of CA, the state sets of the cells were unequal (polygeneous CA).

Planning problems

The planning period was 60 years divided into three 20-year time periods. Spatial problems were formulated for studying the effect of parameters on the performance of our CA and for comparing the CA with SA. Nonspatial problems were formulated to compare CA and LP.

Three imaginary and increasingly difficult spatial problems were formulated. All of the spatial problems had the same global or forest-level objectives, which were sufficient flow of timber (≥175 000 m^3 for each 20-year time period) and sufficient growing stock volume at the end of the planning period (≥500 000 m^3). All problems also had two common local objectives: maximal NPV and aggregation of cut cells (including all types of cuttings, i.e., thinnings, clear cuts, etc.). The cut cells were aggregated by maximizing the proportion of shared border of adjacent cut cells (Cut). The same value for Cut would be obtained by maximizing the proportion of cut cells in the neighbourhoods of cut cells (instead of the border length between cut cells). When using the proportion of shared border of adjacent cut cells as an objective, instead of the proportion of the number of cut cells, the CA method also works with polygon data. Note also that the mean of the Cut values of cells is equal to the overall proportion of shared border of adjacent cut cells in the whole forest, the latter one being a suitable objective variable for centralized heuristics such as SA.

In Problem 2, another spatial goal was to aggregate old forests. Old-forest cells were aggregated by maximizing the proportion of shared border of adjacent old-growth cells (Old). The goodness of a calculation unit as an old-forest patch was measured with an old growth index, which was a function of tree species, stand age, and growing stock volume. The use of stand volume as a criterion prevented sparse stands of old trees (seed tree stands, retention trees) from having high old-forest index values. All cells in which the index was at least 0.5 were considered old-forest patches. This index value could be achieved for instance with a growing stock volume of 300 m^3·ha^{-1} and stand age of 70 (conifers) or 50 (broadleaved trees) years. In Problem 3, old forests and cut cells were aggregated in the same way as in Problem 2 but regeneration cuttings were simultaneously dispersed by minimizing the proportion of shared border of adjacent regenerated cells (Reg). The spatial problems were therefore as follows, where NPV is net present value (discounting rate 2%), C is total cutting volume during a 20-year time period (I, II, or III), and GS is growing stock volume at the end of the 60-year planning period. Problem 1: maximize NPV and Cut subject to C_{I}, C_{II}, $C_{\mathrm{III}} \geq 175\,000$ m^3 and GS ≥ 500 000 m^3. Problem 2: maximize NPV, Old, and Cut subject to C_{I}, C_{II}, $C_{\mathrm{III}} \geq 175\,000$ m^3 and GS ≥ 500 000 m^3. Problem 3: maximize NPV, Old, and Cut and minimize Reg subject to C_{I}, C_{II}, $C_{\mathrm{III}} \geq 175\,000$ m^3 and GS ≥ 500 000 m^3. NPV, Cut, Old, and Reg were included in the local objective functions (eq. 3), whereas all global objectives (constraints in the above formulation) were taken care of by the global objective function (eq. 6). The local and global objective functions were as follows:

Problem 1:

$$U(\text{local}) = 0.4u_1(\text{NPV}) + 0.6u_2(\text{Cut})$$

$$P(\text{global}) = 0.25p_1(C_{\mathrm{I}}) + 0.25p_2(C_{\mathrm{II}}) + 0.25p_3(C_{\mathrm{III}}) + 0.25p_4(\text{GS})$$

Problem 2:

$$U(\text{local}) = 0.25u_1(\text{NPV}) + 0.5u_2(\text{Cut}) + 0.25u_3(\text{Old})$$

$$P(\text{global}) = 0.25p_1(C_{\mathrm{I}}) + 0.25p_2(C_{\mathrm{II}}) + 0.25p_3(C_{\mathrm{III}}) + 0.25p_4(\text{GS})$$

Problem 3:

$$U(\text{local}) = 0.1u_1(\text{NPV}) + 0.4u_2(\text{Cut}) + 0.25u_3(\text{Old}) + 0.25u_4(\text{Reg})$$

$$P(\text{global}) = 0.25p_1(C_{\mathrm{I}}) + 0.25p_2(C_{\mathrm{II}}) + 0.25p_3(C_{\mathrm{III}}) + 0.25p_4(\text{GS})$$

Initial test runs of the system with the parameters used by Strange et al. (2002) for innovations and mutations gave reasonably good results. To investigate the sensitivity of the system to different parameter settings, different parameter values were tested with Problem 2 starting with parameter values proposed by Strange et al. (2002). Innovation probability and the change parameter for innovation probability were first analysed with a zero mutation probability. Second, the mutation probability and the change parameter of mutation probability were examined using the best innovation parameters found in the first step. The parameters were changed until it was clear that no significant improvement was to be found. The best parameter setting was selected for the final comparison runs.

In spatial problems, the CA solutions were compared with solutions generated with SA (Kirkpatrick et al. 1983). SA is a heuristic local search method, commonly known and frequently used for solving spatial forest planning problems. SA used 1-opt moves, i.e., treatment schedule was changed in one cell at the time. The parameters for the SA were searched using the Hooke and Jeeves method described in Pukkala and Heinonen (2006). This article also contains a detailed description of the SA algorithm used. The SA parameters were optimized for the same computing time as required by the CA algorithm. The optimizations were repeated five times for both updating methods (parallel and sequential) of CA and for SA. The best solution was used in the comparison. Both updating methods of the CA had the same parameters for the global optimization. The step of changing the weight of global priority (v_l in eq. 6) was 0.01 and the target value of global priority was 0.99. The maximum number of iterations in the local part was 50.

When SA was used, all objectives were global. For example, the objective function of Problem 1 was

$$P = v_1 p_1(\mathrm{NPV_{SA}}) + v_2 p_2(\mathrm{Cut_{SA}}) + v_3 p_3(C_\mathrm{I}) + v_4 p_4(C_\mathrm{II}) + v_5 p_5(C_\mathrm{III}) + v_6 p_6(\mathrm{GS})$$

in which $\mathrm{NPV_{SA}}$ and $\mathrm{Cut_{SA}}$ are, respectively, the total NPV of all stands and the proportion of shared border of adjacent cut cells in the whole forest.

Four increasingly difficult nonspatial problems were constructed to compare CA with LP. All four problems had the same objective variable, namely NPV calculated with 2% discounting rate. The problems were as follows, where C is the total cutting volume during a 20-year time period (I, II, or III) and RA is the total regeneration area during the 60-year planning period. Problem 1: maximize NPV subject to $C_{\mathrm{I+II+III}} \geq 660\,000$ m^3. Problem 2: maximize NPV subject to C_I, C_II, $C_\mathrm{III} \geq 220\,000$ m^3. Problem 3: maximize NPV subject to $C_\mathrm{I} = C_\mathrm{II} = C_\mathrm{III} = 220\,000$ m^3. Problem 4: maximize NPV subject to $C_\mathrm{I} = C_\mathrm{II} = C_\mathrm{III} = 220\,000$ m^3 and RA ≤ 1000 ha. The LINDO software (e.g., Schrage 1991) was used to solve the LP problems. When formulating the planning problems for CA, the objective functions were as follows:

Problem 1:

$$U(\mathrm{local}) = u_1(\mathrm{NPV})$$

$$P(\mathrm{global}) = p_1(C_{\mathrm{I+II+III}})$$

Problem 2:

$$U(\mathrm{local}) = u_1(\mathrm{NPV})$$

$$P(\mathrm{global}) = 0.333 p_1(C_\mathrm{I}) + 0.333 p_2(C_\mathrm{II}) + 0.333 p_3(C_\mathrm{III})$$

Problem 3:

$$U(\mathrm{local}) = u_1(\mathrm{NPV})$$

$$P(\mathrm{global}) = 0.333 p_1(C_\mathrm{I}) + 0.333 p_2(C_\mathrm{II}) + 0.333 p_3(C_\mathrm{III})$$

Problem 4:

$$U(\mathrm{local}) = u_1(\mathrm{NPV})$$

$$P(\mathrm{global}) = 0.25 p_1(C_\mathrm{I}) + 0.25 p_2(C_\mathrm{II}) + 0.25 p_3(C_\mathrm{III}) + 0.25 p_2(\mathrm{RA})$$

When CA was compared with LP, the sequential mode of CA was used with the same parameters as in the spatial problems. The priority functions for local (u_i) and global objectives (p_i) were linear between the minimum and maximum values, or between the minimum and target values, in both spatial and nonspatial problems. The weights were set and the priority functions formulated in such a way that the global objectives or constraints were always fulfilled. Thus, after reaching the target level of global objectives, additional benefit could be achieved only through local objectives.

When comparing the CA and SA solutions, the core area of old forest was also used as a criterion because it takes into account both the total area and the aggregation of old-forest patches. We used a 30 m buffer, which is the same as in Öhman (2000). All of the computations were conducted with a personal computer that had a Pentium 4 processor (3.06 GHz) and 512 MB of RAM.

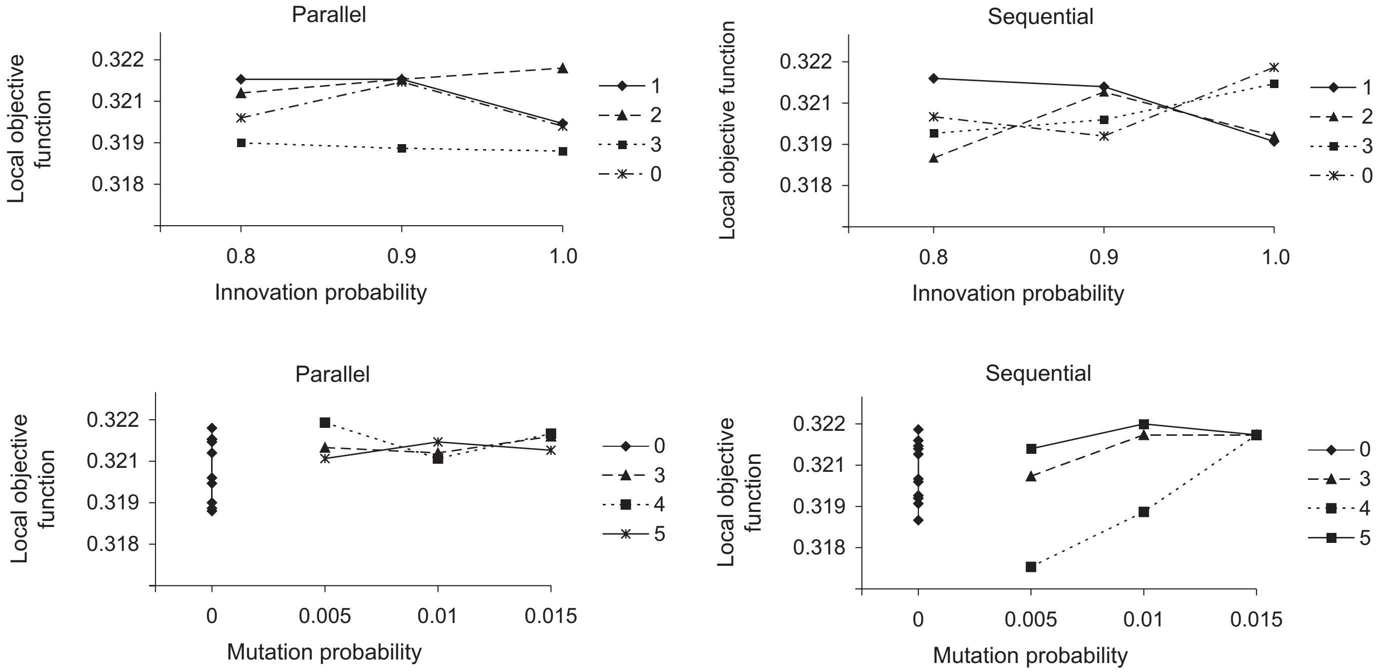
Results

CA parameter values

Tests with different parameter values that control the local part of the algorithm showed that for the sequential mode, the best way for cells to change state was a high initial innovation probability and slow reduction in innovation probability (Fig. 4). The mutation probability had to be low, but otherwise, the mutation probability did not affect the solution quality. Therefore, the initial mutation probability was set to 0 and the initial innovation probability to 1. The change parameter for innovation probability (τ_I in eq. 2) was set to 0. This means that both P_M and P_I were constant with $P_\mathrm{M} = 0$ and $P_\mathrm{I} = 1$ with a consequence that, during every iteration, all of the cells were evaluated and the best treatment schedule was selected.

Also, the parallel mode was insensitive to parameter values controlling the local part if the innovation probability was rather high and the mutation probability low. However, the innovation probability must be lower than 1 if the change parameter (τ_I in eq. 2) is 0 because otherwise, the cells may oscillate between two states. In the analyses of this study, the initial innovation probability was set to 0.8 and the change parameter to 1. The initial mutation probability was set to 0.005 and the change parameter for muta-

Fig. 4. Local objective function values for different parameter values of the cellular automaton. The legend shows the value of the change parameter of innovation or mutation probability (exponent of eq. 1 or 2).

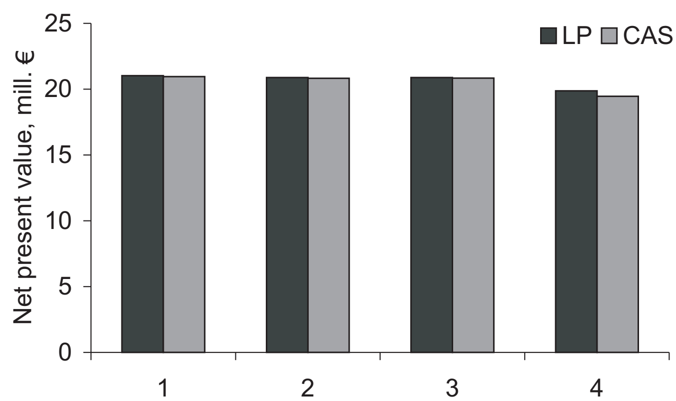


tion probability to 4. In general, there were no significant differences between the performance of different parameter settings, and none of the parameter settings was clearly better than the others.

Comparison with LP

In all of the four nonspatial problems, the constraints were fulfilled with both LP and CA. Thus, differences between the methods emerged only in NPV. In Problems 1, 2, and 3, the NPV obtained with CA was within 99.6% of the maximum and 97.9% in Problem 4 (Fig. 5). Differences in solutions can partly be explained by the integer nature of the CA solutions and partly by the inability of CA to locate the global optimum. However, the NPVs of the CA solutions were very close to those of LP solutions, indicating a good performance of the CA.

Fig. 5. Net present value obtained with the sequential cellular automaton and linear programming in four nonspatial problems.



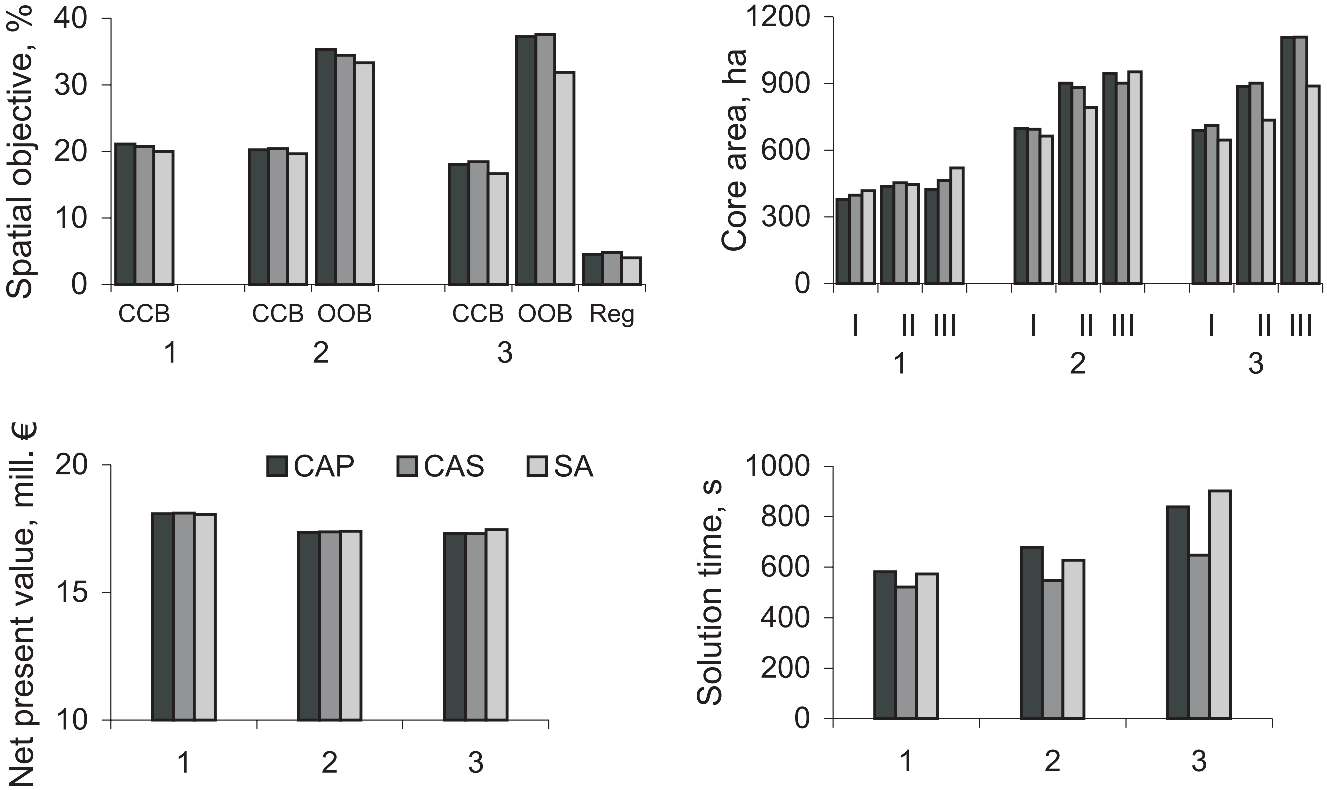
Comparison with SA

Comparisons of CA and SA showed that in a computing time sufficient for CA, SA was not able to produce as good solutions as CA. In Problem 1, SA produced an objective function value of 0.780, whereas the value of the CA solution was 0.791 (sequential mode) or 0.792 (parallel mode) when measured with the objective function of SA. The superiority of CA was nearly the same in Problems 2 and 3. The proportion of shared boundary of adjacent cut cells was on the average about 6% lower with SA (Fig. 6). When old forests were aggregated (Problems 2 and 3), the proportion of old–old border of SA solutions was on the average 96% of CA in Problem 2 and 85% in Problem 3. The old-forest core area of SA was on the average 91% of CA. In Problem 2, the core area was larger in the SA solution at the end of the third time period. In Problem 3, the proportion of shared border of adjacent regeneration cells, which was minimized to disperse regenerative cuts, was about 15% lower in SA, which might be a result from more dispersed cuttings in general in the SA solution. The NPVs of SA and CA solutions were within 1% in all problems and neither SA nor CA was systematically better than the other. The differences between SA and CA solutions in the spatial objectives are visually obvious from Figs. 7 and 8. The solution times of CA varied slightly more than in SA.

Comparison of parallel and sequential modes

CA with sequential (CAS) or parallel (CAP) state-updating modes ended up in very similar solutions in spatial problems (Fig. 9). CAS reached slightly better NPVs in Problems 1 and 2, but CAP was better in aggregating cuttings in Problem 2 (Fig. 6). In Problem 2, the total old-forest core area was larger with CAP for all time periods, but CAS

Fig. 6. Variables describing the cellular automaton (CAP, parallel cell state updating; CAS, sequential updating) and simulated annealing (SA) solutions for spatial Problems 1–3 in terms of three spatial objectives: Cut (CCB), Old (OOB), and Reg. Cut (percentage of shared border of two adjacent cut cells) and Old (percentage of shared border of two adjacent old-forest cells) were maximized and Reg (percentage of shared border of two adjacent regenerated cells) was minimized. I, II, and III refer to a 20-year time period.



was better in Problem 3. The parallel mode dispersed regeneration cuttings more than the sequential mode. The largest difference between the two modes was in the solution time; CAS required at most 90% of the solution time of CAP.

The similarity of the solutions of CAS and CAP are also clearly shown in the spatial layout of old forests and cuttings. Despite different updating of cell states, the two modes produced rather similar spatial arrangements of cut and old-forest cells in Problems 2 (Fig. 7) and 3 (Fig. 8). Most probably the same cells or groups of cells acted as attractors with both modes to aggregate desired features. However, there was a clear difference in the temporal development of the solution between the two modes (Fig. 10). After the first iteration, the CAS solution was clearly more aggregated. This is due to the fact that the sequential mode evaluated all cells during an iteration and changed cell states immediately. Both modes had little variation in the solution quality between five repeated runs (Fig. 9). The parallel mode seemed to work slightly better when the complexity of the problem increased.

Discussion

This study developed an optimization method based on a CA to solve spatial forest planning problems. The CA method developed in this study is able to deal with both stand- and forest-level goals as well as neighbourhood-related goals. It is easy to understand, and due to decentralized computing approach, it is computationally efficient. The hypothesis that the method is able to produce high-quality solutions with modest computational costs is shown in the results. The CA solutions were very competitive compared with solutions generated with traditional methods. In nonspatial problems, the CA method was able to produce solutions very close to LP solutions in terms of objective function value. The benefits of the CA method became clear in spatial problems. In the allowed computing time, the CA method generated solutions better than SA solutions in terms of the spatial layout of desired features. The parameters of SA were optimized for the allowed computing time, which means that the performance of SA would not improve with parameter adjustments if the computing time were not continued.

The biggest advantage of the CA method is that spatial optimization is performed at the stand level, the stand optima depending on adjacent stands. Traditional centralized optimization methods like SA calculate, at every move, the effect of a local change on the global variables. The decentralized way of CA to make calculations locally saves computing time. However, the main reason for the superiority of CA is greatly reduced solution space as compared eith centralized heuristics (Hoganson and Rose 1984). In a small forest of 10 stands, each having five treatment schedules,

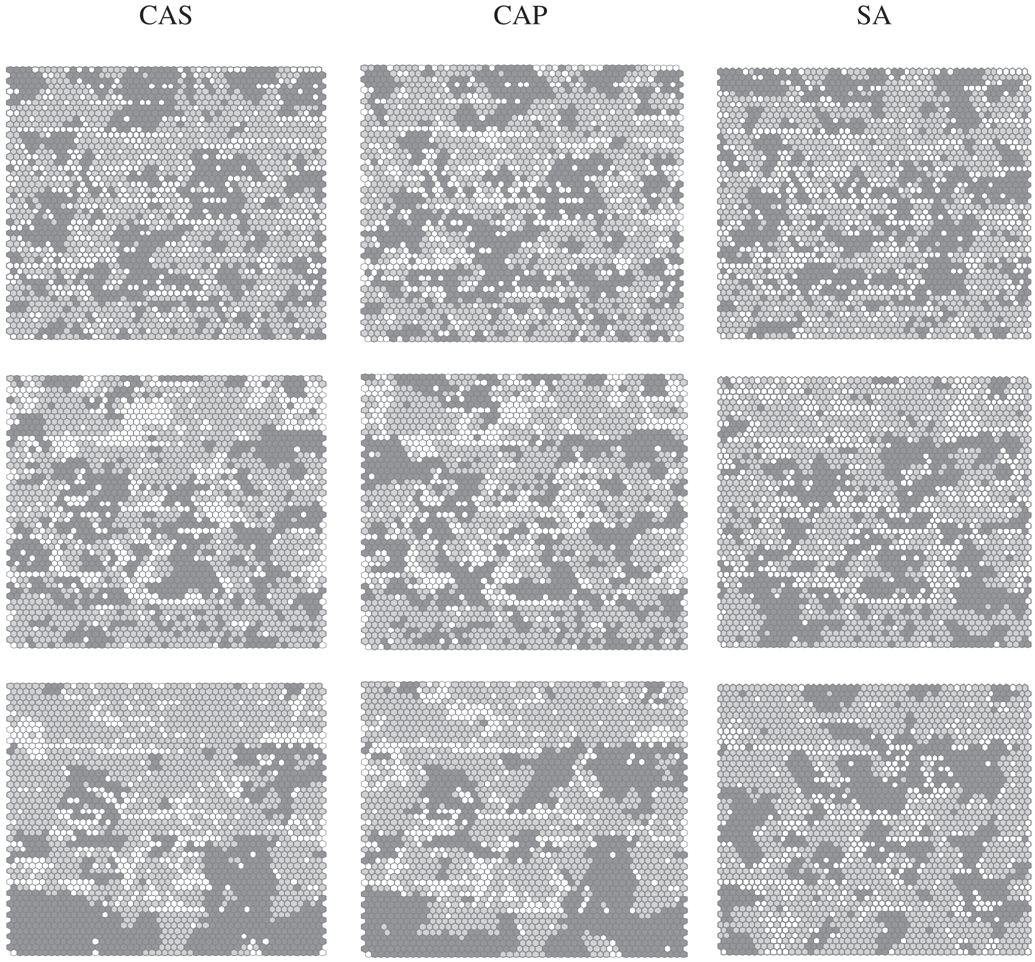


Fig. 7. Solutions of three heuristics in spatial Problem 2. CAS, cellular automaton with sequential updating; CAP, cellular automaton with parallel updating; SA, simulated annealing. From the top: first, second, and third 20-year time period. Light shading, old forest; dark shading, cutting.

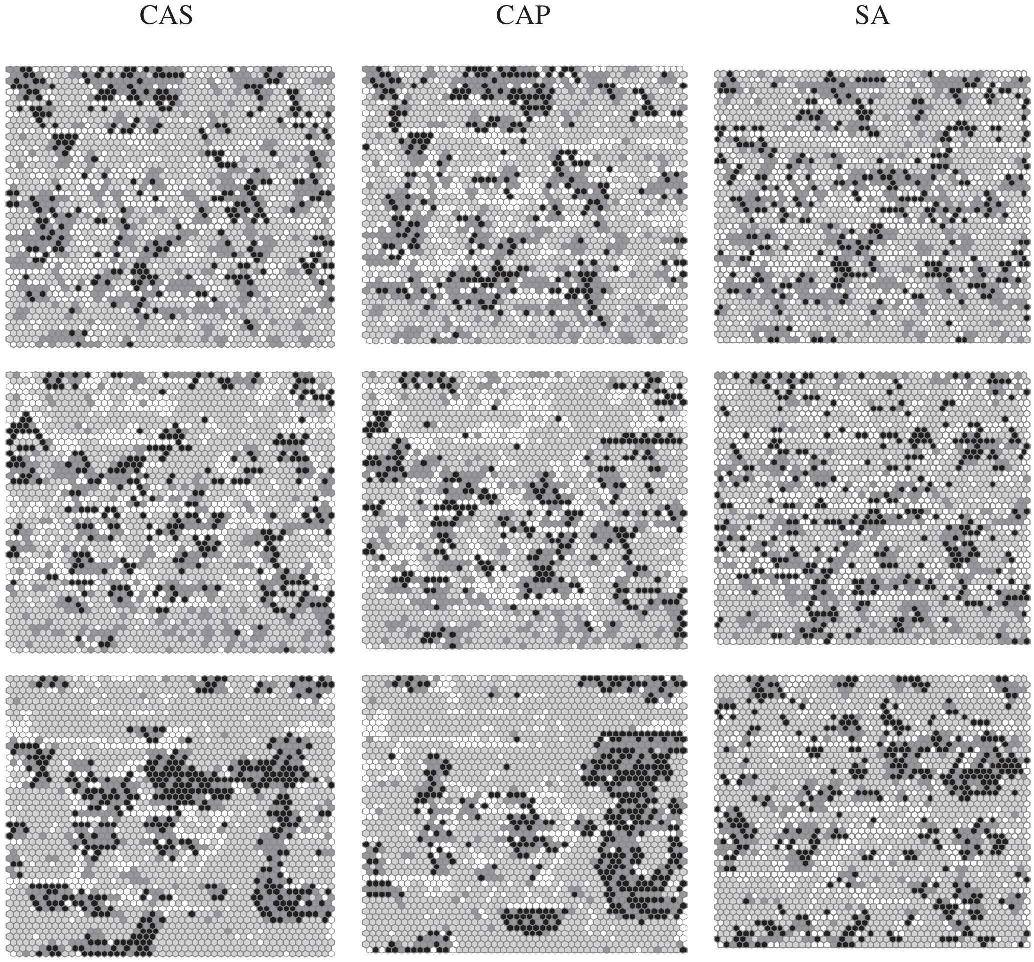
the CA needs to inspect $5 \times 10 = 50$ local alternatives, whereas centralized methods need to search the best combination from among $5^{10} = 9\,765\,625$ alternatives. All local alternatives can be inspected many times in the same time that is needed to evaluate a small fraction of the solution space of centralized methods.

Our CA resembles the Hero method proposed by Pukkala and Kangas (1993) in the sense that all schedules of all stands are systematically evaluated for many iterations. However, the Hero method has been found to be not as good as SA, genetic algorithms, and tabu search in difficult spatial problems (Pukkala and Kurttila 2005). An important difference between our CA and the Hero method is that the CA completes the local optimizations first, after which the solution is gradually modified to meet the global objectives. This corresponds to generating a spatially good but infeasible (in terms of global objectives) solution first, after which the solution is altered until it becomes feasible. Figure 10 shows the ability of the local optimization to reach a good spatial organization from which the solution is further adjusted by extending or shrinking the cutting areas or old-forest patches depending on the global objectives. This strategy seems to be clearly better than working all of the time with feasible solutions.

Decentralized computing like CA is the most suitable for solving problems in which the goals are additive, i.e., sums of stand-level values. Although local optimization is seldom sufficient for solving forest-level planning problems, it may quickly produce very good starting points for global optimization. The gain can be particularly significant if raster cells or other fine-grained data are used as calculation units instead of the traditional stand compartments. In addition, the optimization result may be better when local optimization is used in the beginning of optimization. This is because local optimization with neighbourhood functions efficiently finds the natural way for the forest landscape to self-organize. The result may not be feasible in terms of global objectives, but converting it into a feasible one may only need some fine-tuning of the set of locally optimal solutions.

Our CA also resembles the method of Hoganson and Rose (1984) who used heuristically adjusted dual prices of LP constraints to tie the stand-level problems together. The stand-level objective variable was modified on the basis of dual prices and values of constraining variables. Stand-level

Fig. 8. Solutions of three heuristics in spatial Problem 3. CAS, cellular automaton with sequential updating; CAS, cellular automaton with parallel updating: SA, simulated annealing. From the top: first, second, and third 20-year time period. Light shading, old forest; dark shading, thinning; solid, regeneration cutting.

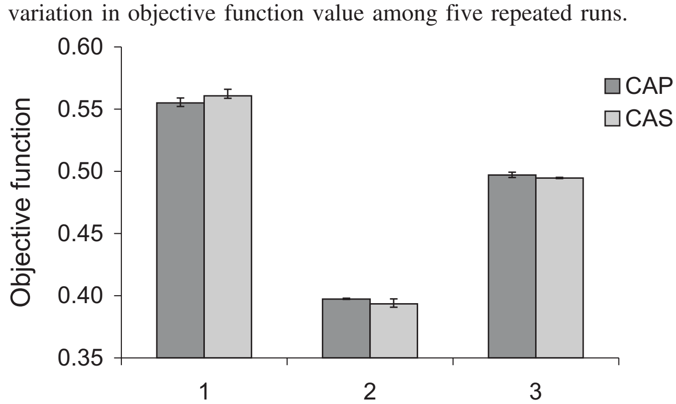


optimizations were repeated after every adjustment, and the optimizations were stopped when the constraining variables reached good enough values. However, the method of Hoganson and Rose (1984) was developed only for nonspatial problems.

The developed CA for spatial forest planning is capable of handling both cell- and vector-based data. The method also allows the user to set the level of how well the global targets should be met. This property gives more freedom for optimization and reduces the solution times when the target levels must not be exactly met. Models and simulations representing real-world include uncertainty, and there are temporal changes in decision makers' preferences, timber prices, etc. Therefore, the requirement for optimality can be relaxed and minor violations of constrains accepted (Hoganson and Rose 1984).

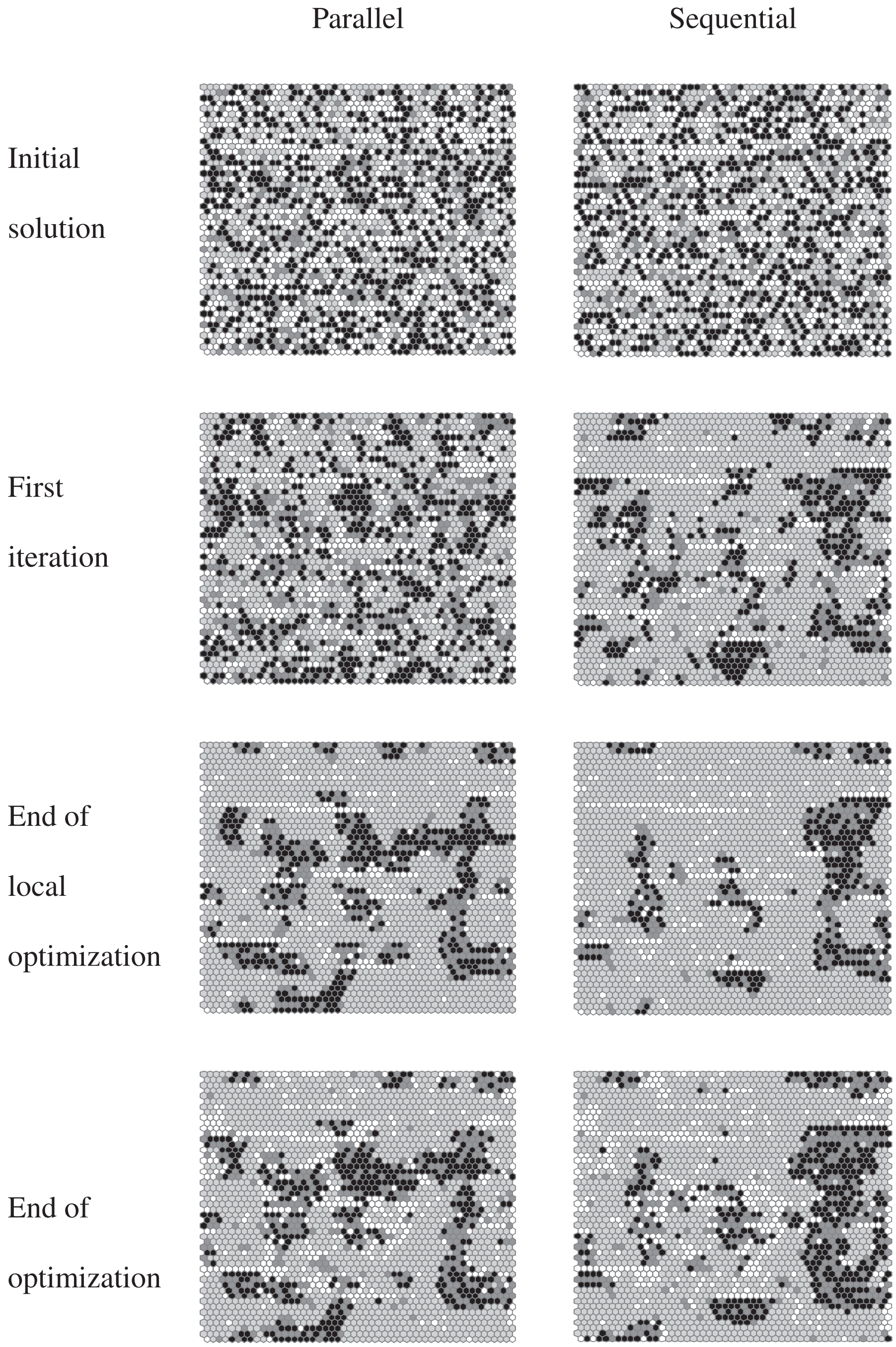
There were no significant differences in the performance of the parallel and sequential modes of the CA. The performance of the CA was rather insensitive to the parameters of the algorithm. The parallel mode needs some kind of evolutionary updating rules to change cell states. Otherwise, a cell may start oscillating between two states. This was also a reason why mutations and innovations were included in the algorithm of Strange et al. (2002). The results of our study suggest that the sequential model is equally good as the parallel one, and a constant high innovation probability with no mutations leads to a good performance. This makes the method very easy to use in forestry practice.

Fig. 9. Mean objective function value and the variation of solutions of the parallel (CAP) and sequential (CAS) modes of cellular automaton in spatial Problems 1–3. The vertical lines show the range of variation in objective function value among five repeated runs.



The algorithm could be further developed and diversified,

Fig. 10. Development of the solution during a run of the cellular automaton in terms of spatial objectives for the third 20-year time period in spatial Problem 3. Light shading, old forest; dark shading, thinning; solid, regeneration cutting.



e.g., by using different kinds of cell neighbourhoods. In forest planning problems, it may be advantageous to use cell neighbourhoods that take the neighbours of immediate neighbours into account also or neighbours inside a specific radius. Also, the distance from important features like roads could be used as a neighbourhood function. These are ways to affect the size and location of the desired features.

References

Barredo, J.I., Kasanko, M., McCormick, N., and Lavalle, C. 2003. Modelling dynamic spatial processes: simulation of urban future scenarios through cellular automata. Landscape Urban Plan. **64**: 145–160.

Baskent, E.Z., and Jordan, G.A. 2002. Forest landscape management modelling using simulated annealing. For. Ecol. Manag. **165**: 29–45.

Bettinger, P., Sessions, J., and Boston, K. 1997. Using tabu search to schedule timber harvests subject to spatial wildlife goals for big game. Ecol. Model. **42**: 111–123.

Bettinger, P., Boston, K., and Sessions, J. 1999. Intensifying a heuristic forest harvest scheduling search procedure with 2-opt decision choices. Can. J. For. Res. **29**: 1784–1792. doi:10.1139/cjfr-29-11-1784.

Bettinger, P., Graetz, D., Boston, K., Sessions, J., and Chung, W. 2002. Eight heuristic planning techniques applied to three increasingly difficult wildlife planning problems. Silva Fenn. **36**: 561–584.

Bettinger, P., Graetz, D., and Sessions, J. 2005. A density-dependent stand-level optimization approach for deriving management prescriptions for interior northwest (USA) landscapes. For. Ecol. Manag. **217**: 171–186.

Boston, K., and Bettinger, P. 2001. The economic impact of green-up constraints in the southeastern United States. For. Ecol. Manag. **145**: 191–202.

Boston, K., and Bettinger, P. 2002. Combining tabu search and genetic algorithm heuristic techniques to solve spatial harvest scheduling problems. For. Sci. **48**: 35–46.

Brumelle, S., Granot, D., Halme, M., and Vertinsky, I. 1997. A tabu search algorithm for finding good forest harvest schedules satisfying green-up constraints. Eur. J. Oper. Res. **106**: 408–424.

Creutz, M. 1986. Deterministic Ising dynamics. Ann. Phys. **67**: 62–72.

Hargrove, W.W., Gardner, R.H., Turner, M.G., Romme, W.H., and Despain, D.G. 2000. Simulating fire patterns in heterogeneous landscapes. Ecol. Model. **135**: 243–263. doi:10.1016/S0304-3800(00)00368-9.

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

Hoganson, H.M., and Rose, D.W. 1984. A simulation approach for optimal timber management scheduling. For. Sci. **30**: 220–238.

Keymer, J.E., Marquet, P.A., and Johnson, A.R. 1998. Pattern formation in a patch occupancy metapopulation model: a cellular automata approach. J. Theor. Biol. **194**: 79–90. doi:10.1006/jtbi.1998.0745. PMID:9778426.

Kirkpatrick, S., Gelatt, C.D., Jr., and Vecchi, M.P. 1983. Optimization by simulated annealing. Science (Washington, D.C.), **220**: 671–680. doi:10.1126/science.220.4598.671.

Kurttila, M., Pukkala, T., and Loikkanen, J. 2002. The performance of alternative spatial objective types in forest planning calculations: a case for flying squirrel and moose. For. Ecol. Manag. **166**: 245–260.

Lett, C., Silber, C., and Barret, N. 1999. Comparison of a cellular automata network and an individual-based model for the simulation of forest dynamics. Ecol. Model. **121**: 277–293. doi:10.1016/S0304-3800(99)00090-3.

Li, X., and Magill, W. 2001. Modeling fire spread under environmental influence using cellular automaton approach. Complexity International [serial online]. Vol. 8. Available from www.complexity.org.au

Lockwood, C., and Moore, T. 1993. Harvest scheduling with spatial constraints: a simulated annealing approach. Can. J. For. Res. **23**: 468–478.

Maerivoet, S., and De Moor, B. 2005. Cellular automata models of road traffic. Phys. Rep. **419**: 1–64. doi:10.1016/j.physrep.2005.08.005.

Mathey, A.H., Krcmar, E., and Vertinsky, I. 2005. Re-evaluating our approach to forest management planning: a complex journey. For. Chron. **81**: 359–364.

Mathey, A.H., Krcmar, E., Tait, D., Vertinsky, I., and Innes, J. 2007. Forest planning using co-evolutionary cellular automata. For. Ecol. Manag. **239**: 45–56.

Meilby, H., Strange, N., and Brukas, V. 2001. Decentralised optimisation of the spatial allocation of recreational and forest reserve areas subject to global constraints. Scand. J. For. Econ. **38**: 66–75.

Murray, A.T., and Church, R.L. 1995. Measuring the efficacy of adjacency constraint structure in forest planning models. Can. J. For. Res. **25**: 1416–1424.

Nalle, D.J., Arthur, J.L., and Sessions, J. 2002. Designing compact and contiguous reserve networks with a hybrid heuristic algorithm. For. Sci. **48**: 59–68.

Pommerening, A. 2006. Evaluating structural indices by reversing forest structural analysis. For. Ecol. Manag. **224**: 266–277.

Puigdefábregas, J., Gallard, F., Biaciotto, O., Allogia, M., and del Barrio, G. 1999. Banded vegetation patterning in a subantarctic forest of Tierra del Fuego, as an outcome of the interaction between wind and tree growth. Acta Oecol. **20**: 135–146. doi:10.1016/S1146-609X(99)80027-7.

Pukkala, T. 2004. Monikäytön suunnitteluohjelma Monsu. Ohjelmiston toiminta ja käyttö. Faculty of Forestry, University of Joensuu, Joensuu, Finland. [In Finnish.]

Pukkala, T., and Heinonen, T. 2006. Optimizing heuristic search in forest planning. Nonlinear Anal. Real World Appl. **7**: 1284–1297. doi:10.1016/j.nonrwa.2005.11.011.

Pukkala, T., and Kangas, J. 1993. A heuristic optimization method for forest planning and decision-making. Scand. J. For. Res. **8**: 560–570.

Pukkala, T., and Kurttila, M. 2005. Examining the performance of six heuristic search techniques in different forest planning problems. Silva Fenn. **39**: 67–80.

Pukkala, T., and Miina, J. 1997. A method for stochastic multiobjective optimization of stand management. For. Ecol. Manag. **98**: 189–203.

Raghavan, R. 1993. Cellular automata in pattern recognition. Inf. Sci. **70**: 145–177. doi:10.1016/0020-0255(93)90052-N.

Roise, J.P. 1986. A nonlinear programming approach to stand optimization. For. Sci. **32**: 735–748.

Sarkar, P. 2000. A brief survey of cellular automata. ACM Comput. Surv. **32**: 81–107.

Schrage, L. 1991. LINDO, an optimization modeling system. 4th ed. The Scientific Press, South San Francisco, Calif.

Schönfisch, B., and de Roos, A. 1999. Synchronous and asynchronous updating in cellular automata. Biosystems, **51**: 123–143. doi:10.1016/S0303-2647(99)00025-8. PMID:10530753.

Soares-Filho, B.S., Goutinho Cerqueira, G., and Lopes Pennachin, C. 2002. DINAMICA — a stochastic cellular automata model designed to simulate the landscape dynamics in an Amazonian colonization frontier. Ecol. Model. **154**: 217–235. doi:10.1016/S0304-3800(02)00059-5.

Sprott, J.C., Bolliger, J., and Mladenoff, D.J. 2002. Self-organized criticality in forest-landscape evolution. Phys. Lett. A, **297**: 267–271. doi:10.1016/S0375-9601(02)00052-X.

Strange, N., Meilby, H., and Bogetoft, P. 2001. Land use optimization using self-organizing algorithms. Nat. Resour. Model. **14**: 541–574.

Strange, N., Meilby, H., and Thorsen, J.T. 2002. Optimizing land use in afforestation areas using evolutionary self-organization. For. Sci. **48**: 543–555.

Syphard, A.D., Clarke, K.C., and Franklin, J. 2005. Using a cellular automaton model to forecast the effect of urban growth on habitat pattern in southern California. Ecol. Complexity, **2**: 185–203.

Tarp, P., and Helles, F. 1997. Spatial optimization by simulated annealing and linear programming. Scand. J. For. Res. **12**: 390–402.

Torres-Rojo, J.M., and Brodie, J.D. 1990. Demonstration of benefits from an optimization approach to the economic analysis of natural pine stands in central Mexico. For. Ecol. Manag. **36**: 267–278.

Von Neumann, J. 1966. Theory of self-reproduction automata. *Edited by* A. Burks. University of Illinois Press, Champaign, Ill.

Ward, D.P., Murray, A.T., and Phinn, S.R. 2003. Integrating spatial optimization and cellular automata for evaluating urban change. Ann. Reg. Sci. **37**: 131–148. doi:10.1007/s001680200113.

Wolfram, S. 1984. Universality and complexity in cellular automaton. Physica, **10D**: 1–35.

Öhman, K. 2000. Creating continuous areas of old forest in long-term forest planning. Can. J. For. Res. **30**: 1817–1823. doi:10.1139/cjfr-30-11-1817.

Öhman, K., and Eriksson, L.O. 2002. Allowing for spatial consideration in long-term forest planning by linking linear programming with simulated annealing. For. Ecol. Manag. **161**: 221–230.

Öhman, K., and Lämås, T. 2003. Clustering of harvest activities in multi-objective long-term forest planning. For. Ecol. Manag. **176**: 161–171.