Available online at www.sciencedirect.com
ScienceDirect
Forest Ecology and Management 239 (2007) 45–56
Forest Ecology and Management
www.elsevier.com/locate/foreco

Forest planning using co-evolutionary cellular automata

Anne-Hélène Mathey [a,b,*], Emina Krcmar [a], David Tait [b], Ilan Vertinsky [a], John Innes [b]

[a] *Forest Economics and Policy Analysis (FEPA) Research Unit, University of British Columbia, Forest Sciences Centre, 2424 Main Mall, Vancouver, BC, Canada V6T 1Z4*

[b] *Forest Resources Management, Faculty of Forestry, University of British Columbia, Forest Sciences Centre, 2424 Main Mall, Vancouver, BC, Canada V6T 1Z4*

Received 4 October 2005; received in revised form 14 November 2006; accepted 14 November 2006

Abstract

The spatial distribution of forest management activities has become increasingly important with, most notably, rising concerns for biodiversity. Addressing both timber production and non-timber goals requires planning tools that support spatially explicit decision-making. The paper examines the capability of a co-evolutionary cellular automata (CA) approach to address forest planning objectives that are both spatial and temporal with global constraints. In this decentralized self-organizing planning framework, each forest stand and its associated management treatment over the planning horizon is represented as a cellular automaton. The landscape management goals are achieved through a co-evolutionary decision process between interdependent stands. A novel, computationally efficient CA algorithm for asynchronous updating of stand states is developed. The specific problem considered in the paper is maximization of cumulative harvest volume and amount of clustered late-seral forest. The global constraints considered are stable harvest flow and minimum amount of late-seral stands in each period of the planning horizon. Applied to a test area from the Northeastern forest region of Ontario, Canada, the model demonstrates short computation time and consistent results from multiple runs. It also compares favorably with outputs from a simulated annealing search. The CA-based algorithm developed in the paper successfully identifies sustainable forest outputs over the planning horizon. It shows sensitivity to both local constraints, strategic goals and strategic constraints and generates spatially explicit forest plans.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Cellular automata; Decentralized planning; Evolutionary game; Forest planning; Heuristics; Multiple scales; Late-seral forest; Self-organization; Spatial analysis; Tradeoffs

1. Introduction

Sustainable forest management recognizes and seeks to maintain a wide array of ecological as well as economic and social forest functions, both locally and globally (UNCED, 1992). The challenge in forest management planning is to accommodate timber production with other, non-timber goals such as the protection of biodiversity and ecosystem health. To this end, traditional stand-level cost-benefit-type analyses need to be combined with forest or regional-level analyses in order to adequately select among forest management alternatives. As forest management interacts with ecological processes at multiple spatial and temporal scales, combined analyses can be difficult to conduct (Martell et al., 1998; Nelson, 2003). These issues are usually handled through either top-down or bottom-up planning approaches (Shands et al., 1990).

Top-down planning is the most frequent approach to reconcile processes and goals at different spatial and temporal levels. This approach is often associated with centralized procedures that track the global performance of decision combinations to select the best alternative. Centralized top-down procedures have traditionally been favored in strategic planning with an extensive use of large non-spatial mathematical programming models (Martell et al., 1998). However, an increasing number of location-specific concerns (e.g., environmental buffers, proximity to road network, adjacency, size and distribution of reserve patches) have made necessary the inclusion of spatial considerations in the planning process. For strategic plans to be useful, their spatial implementation must be feasible. Spatial details typically generate a large number of search combinations.

* Corresponding author at: Forest Resources Management, Faculty of Forestry, University of British Columbia, 2424 Main Mall, Vancouver, BC, Canada V6T 1Z4. Tel.: +1 604 822 3450; fax: +1 604 822 9106.

E-mail address: mathey@interchange.ubc.ca (A.-H. Mathey).

0378-1127/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.foreco.2006.11.007

The use of heuristic methods has greatly facilitated solving the resulting computationally difficult planning problems. The most frequently applied heuristics are simulated annealing (Lockwood and Moore, 1993), Monte Carlo search (Clements et al., 1990; Boston and Bettinger, 1999), tabu search (Murray and Church, 1995; Bettinger et al., 1997; Richards and Gunn, 2003), genetic algorithms (Lu and Eriksson, 2000) or combinations of several heuristics (Boston and Bettinger, 2002). Despite the increasing effectiveness of heuristic methods, it remains difficult to formalize spatial objectives such as clustering, connectivity and continuity of set-aside forestland throughout the planning horizon with centralized procedures (Bettinger et al., 2002; Nelson, 2003; Pukkala and Kurttila, 2005). The heuristics used in centralized procedures are based on the evaluation of the global objective values for different management plans. If the global objective value is not satisfactory for a given plan, lower-level decisions are changed until an acceptable objective value is obtained. Another way of finding a satisfactory management plan is to build information from the lower levels (i.e., individual stands).

Bottom-up planning offers the advantage of directly addressing local spatial goals and constraints. In order to achieve an acceptable level of the global objective, some coordination of lower-level decisions is required. In spatial systems decisions taken at nearby locations can affect each other's contribution to the global objective more than decisions taken at distant locations (Strange et al., 2001; Hoganson and Borges, 1998; Hoganson et al., 1998). This implies that the solution space could be explored in parallel by evaluating local decisions and taking into account their interactions. Given the increasing complexity of forest systems, a decentralized approach based on local-level decisions is a natural way to address strategic forest planning. A decentralized bottom-up framework may address the spatial goals of forest management computationally more efficiently than a centralized framework.

Cellular automata (CA) modeling is a decentralized framework capable of representing discrete dynamical systems whose behavior is specified in terms of local relations (Toffoli and Margolus, 1987). This modeling tool has been almost exclusively used for process simulation and for the exploration of complex systems in physics, geography and biology. If CA models are to be used for planning purposes, the incorporation of guiding rules is needed to ensure that not only local (stand-level) objectives, but also global (landscape-level) goals are met. Strange et al. (2001) developed a planning tool centered on a CA-based evolutionary optimization algorithm, which solves spatial problems involving one-time afforestation decisions (Strange et al., 2002). Mathey et al. (2005) extended the work on using CA for planning by designing an evolutionary CA algorithm to address inter-temporal aspects of forest planning in addition to spatial issues. The newly developed co-evolutionary optimization algorithm was applied to solving spatial multi-period planning problems. The current paper addresses global constraints within this decentralized optimization framework and proposes a modification of both the transition rules and updating method that is capable of guiding the decentralized optimization process toward meeting global objectives while also satisfying landscape scale (global) constraints.

This study describes the co-evolutionary algorithm for both an unconstrained and a constrained forest planning problems and compares its performance with a simulated annealing algorithm. The specific forest planning problem considered is the maximization of a weighted combination of the cumulative harvest volume and a measure of the clustering of old forest stands, while keeping stable harvest flow over time and maintaining a minimum amount of the forested area in late-seral stage. Volume maximization and stable harvest flow reflect timber-related objectives of forest planning. Ecological objectives of forest management are expressed by both maintaining a minimum amount of late-seral forest stands and at the same time promoting the clustering of these late-seral stands. Maintaining clusters of late-seral forest is important for several reasons. First, late-seral forests are most severely impacted by timber management, both through direct degradation and through fragmentation (Harris, 1984). Second, late-seral forests constitute a specific habitat that a number of plant and animal species depend on, partially or entirely. Preserving habitat for these species requires continuity of areas of stands of old forests over the planning horizon (Seymour and Hunter, 1999; Ohman, 2000). Both forest ecologists and managers acknowledge the importance of conserving clusters of old forests (Spies and Franklin, 1996) even if the question of their location, size and distribution is still in contentions (Shafer, 2001; Schwartz, 1999).

The following section describes how cellular automata can be used in forest planning. We detail the co-evolutionary CA algorithm of Mathey et al. (2005) for an unconstrained forest planning problem after which, we present the solution approach to constrained forest planning problems. The model and solution approaches are then illustrated by an empirical study using a forest in northeast Ontario. The tradeoffs in the model are analyzed and a comparison with a simulated annealing search algorithm is proposed.

2. Application of cellular automata in forest planning

CA models are generally constructed as a collection of cells that form a lattice (Toffoli and Margolus, 1987; Wolfram, 1994). Each cell is characterized as being in a particular state. Transition rules applied to a cell are used to compute the cell's state for the next iteration. The transition rules are a function of the cell's own state and the states of its neighbors. CA models progress by discrete steps (a time interval or iteration) during which transition functions are applied to all or a subset of the lattice's cells.

Two properties of CA are particularly relevant to forest planning: scale-integration and self-organization. The integration of processes and objectives from different spatial and temporal scales is a long-standing issue in forest planning: some of the management concerns are local or stand-level (e.g., timber yield, stand structure) while others are tied to the surrounding environment (e.g., landscape patterns, economics and aesthetics). Self-organization in dynamic systems refers to the spontaneous emergence of global coherence out of local interactions (Krugman, 1996). Self-organization is particularly relevant to forests where landscape patterns result from the interactions of natural disturbances, succession dynamics, and human-induced stand management decisions. The ability of CA to develop emergent characteristics enables the modeling of spatial and temporal interactions within the forest landscape, such as gap dynamics (Hubbell and Foster, 1986), fire spread (Green, 1989) and species interactions (Colasanti and Grime, 1993). CA models are also well suited to forest management allocation problems since any land allocation or management decision reflects the suitability of a local cell for a specific activity and also the suitability of this activity within the neighborhood of the cell.

The details of an algorithm for multi-period forest planning with constraints, which fully employs the self-organizing and scale-integration properties of CA, are presented. Contextualized in a CA framework, the forest is characterized as a raster system with a set of uniformly sized, square stands. Each stand, f, has an associated stand type and stand age at the beginning of the planning horizon. Each stand also has an associated set of possible treatment schedules. In terms of cellular automata, the forest is the lattice or grid, the stand is a cell and the treatment schedule for a stand represents the state of the cell. The stand-specific treatment schedules reflect the stand type associated with the raster cell together with the alternative possible schedules of treatments (silviculture and harvest) that could meaningfully be applied to that given stand type. For example, the set of treatment schedules available for a raster cell that was initially a young birch stand would differ from the set of treatment schedules that would be available for a late-seral pine stand.

Cellular automata models are iterative: a CA models begins with an assignment of initial states to all of the cells and progresses by updating cell states according to a state transition function. Formally, a plan $P_i\{s_i(f){:}f \in F\}$, is the configuration of the states of all cells at iteration i. As the cells represent stands, and the states represent schedules, P_i is therefore a management plan.

The state $s_i(f)$ is the treatment schedule associated with stand f in the ith iteration. Treatment schedules range from a state that represents 'no treatment' with no actions scheduled for the length of the planning horizon through to specific schedules involving multiple harvests and silvicultural regimes applied over the planning horizon.

The lattice F is the forest represented as the set of stands. Each stand, f, has an associated location, set of neighbors, and initial conditions (initial stand type and stand age). In the above formalism, f and s are both indexing systems. f is an index into the set of all stands and s is an implicit index into the set of all possible treatment schedules.

A state transition function determines a new (possibly the same) state for a stand as a function of its current stand state and its neighbors' states. The transition function φ is applied to each stand f and its neighborhood to generate the new stand state:

$$s_{i+1}(f) = \varphi(S_i(f))$$

where $s_{i+1}(f)$ is the state of stand f in iteration $i+1$; $\varphi(\cdot)$ the state transition function; $s_i(f)$ is a vector representing the states, in iteration i, of stand f and its neighbors.

The state transition function in our algorithm chooses the stand schedule that maximizes an objective function.

The last element of a CA model is the updating method that defines, for each iteration i, the set of stands to which the transition function is applied.

The forest plan evolves iteratively by generating new schedules (states) and applying the state transition function to individual stands as determined by the updating scheme. The forest plan P_i from iteration i changes into a new plan $P_{i+1} = \{s_{i+1}(f){:}f \in F\}$ in iteration $i+1$. The evolution of the plan will depend on both the transition rules and updating method formulations while the process duration is determined by stopping rules.

3. Unconstrained forest planning

We begin by formulating an unconstrained multi-period forest planning problem and the co-evolutionary optimization method developed to solve it. This algorithm was sketched in Mathey et al. (2005). This is the first detailed description and rationale presented for this algorithm. We further extend the algorithm to apply to a constrained multi-period forest planning problem.

3.1. Model formulation

In this model, the state transition function evaluates all the treatment schedules for a stand and chooses the stand's next state to be the treatment schedule that generates the highest value. This evaluation is a function of the individual stand and reflects both the stand type and the initial stand inventory. The evaluation of each treatment schedule is separated into two components.

The first component represents context-independent values. In general, this context-independent component is a site-specific evaluation that generates a value. This value can depend on a rich set of stand-specific criteria that will include factors that depend on the site type and stand location. In our model we have identified these as being proportional to the total harvest for the treatment schedule over the planning horizon.

The second component represents context-dependent values. In general, we model this as site-specific values that can be modified or augmented as a function of the states of their neighboring cells. In our forest model, the context-dependent value is the conservation value associated with a stand being in late-seral stages. In any given time period, late-seral stands have a conservation value that is enhanced if the neighboring stands are also late-seral at the same time. The context-dependent and context-independent values are both normalized relative to each stand's potential.

Formally, the context-independent value is formulated as

$$I(s(f)) = \frac{\sum_t \mathrm{HV}_t(s(f))}{\mathrm{HV}_f} \quad (1)$$

where $I(s(f))$ is the context-independent value associated with treatment schedule $s(f)$ for stand f. Although $I(s(f))$ could take into consideration factors such as cost and represent a net present value, in our model we only considered the total volume harvested over the planning horizon. In what follows, the context-independent value is referred to as the harvest value. $\mathrm{HV}_t(s(f))$ is the harvest volume in period t generated by treatment schedule $s(f)$. HV_f is a scaling factor that ensures that I ranges between 0 and 1. As a scaling factor, HV_f is the maximum possible total harvest that could be generated by a stand type similar to that of f over the planning horizon T.

The harvest value for a treatment schedule is thus an index that ranges between 0 and 1. A harvest value of 0 corresponds to a treatment schedule that had no harvesting and a harvest value of 1 corresponds to a treatment schedule that achieved the maximum possible total harvest volume for all stands of that type over the planning horizon T.

Formally, the context-dependent value is formulated as

$$D(s(f)) = \frac{\sum_t [\mathrm{LS}_t(s(f)) + \rho \times \mathrm{LSN}_t(f)]}{\mathrm{LS}_f} \tag{2}$$

where $D(s(f))$ is the context-dependent value associated with treatment schedule $s(f)$, $\mathrm{LS}_t(s(f))$ and $\mathrm{LSN}_t(f)$ are the late-seral conservation values of a stand and its neighborhood, respectively, and LS_f is a scaling factor. In this case, $D(s(f))$ considers the stand's conservation value, which is further enhanced by the proportion of the stand's neighbors that are simultaneously scheduled to remain late-seral. $\mathrm{LS}_t(s(f))$ is either 1 or 0 and indicates whether the stand f is late-seral or not in each planning period t of the planning horizon T under treatment schedule $s(f)$. Late-seral forest, in this context, represents a stand that is over a specific age usually associated with the onset of late-seral characteristics. The actual age varies for different stand types and species associations.

$\mathrm{LSN}_t(f)$ is a factor between 0 and 1 that is equal to the proportion of stand f's neighbors that are also late-seral in planning period t. ρ is a scaling factor that can be used to modify the degree to which late-seral neighbors influence the late-seral conservation value of a stand. $\rho \times \mathrm{LSN}_t(f)$ represents the enhancement of the stand's late-seral conservation value that is due to the simultaneous existence of neighboring late-seral stands. We set ρ to be 1, which gives equal importance to the neighboring late-seral forest as to the stand late-seral state itself. The stand's neighborhood consists of the eight adjacent stands in the raster grid. To limit a bias caused by missing parts of the neighborhood for the cells located at the edge of the grid, we assume that the unknown part of their neighborhood is proportionally similar to the known part of the neighborhood for the edge cells. If the stand is on an edge, we only use the five adjacent stands. Corner neighbors consist of the three adjacent stands.

LS_f is a scaling factor that ensures that D ranges between 0 and 1. LS_f is the maximum possible enhanced late-seral conservation value that could be generated by a stand f over the planning horizon. Numerically, LS_f, is just twice the number of planning periods T.

The late-seral conservation value is thus an index that ranges between 0 and 1. A schedule that results in a stand never reaching the late-seral stage during the planning horizon would have a late-seral conservation value of 0. A schedule that would result in a stand being in the late-seral stage for the entire course of the planning horizon would generate a minimum late-seral conservation value of 0.5, when none of the stand's neighbors ever reach the late-seral stage and a maximum late-seral conservation value of 1, when all of the neighbors are also in late-seral stage in each period.

The value of a treatment schedule, $z(s(f))$, is calculated as a weighted average of the above two components:

$$z(s(f)) = \lambda \times I(s(f)) + (1 - \lambda) \times D(s(f)) \tag{3}$$

where λ is a weighting factor that is between 0 and 1. λ represents the importance attached to timber production relative to late-seral conservation. Also, λ reflects policy and is a weighting factor that influences the behavior of the algorithm. Values of λ that are close to 1 will generate solutions that reflect context-independent values, while values of λ that are close to 0 will favor solutions that reflect context-dependent values.

Here, the state transition function is defined as selecting the treatment schedule $s_{i+1}(f)$ that maximizes the stand value in Eq. (3):

$$s_{i+1}(f) = \varphi(S_i(f)) = \max_s \; z(s_i(f)) \tag{4}$$

The last element of a CA model is the iterative scheme. The iterative scheme clarifies the order that the cells in the automata—the stands in our model, are to be updated. Deterministic synchronous updating (Wolfram, 1994) is the most frequently used iterative scheme. Following this updating method, the transition rules are applied simultaneously to all cells at each iteration. Synchronous updating often leads to oscillatory behavior. A cell changes its state in response to the current states of its neighboring cells. However, its neighboring cells have also, synchronously, changed their states. The new state selected by the state transition function is no longer optimum. In the next iteration, the cell changes its state back to the state that it had in the previous iteration. The algorithm gets stuck with cells flipping back and forth in response to the oscillations of their neighbors. Some modifications of the CA model formulation are required so that local decisions converge towards a stable CA configuration (forest management plan) that has an optimal (or at least satisfactory) value.

4. Co-evolutionary optimization: from local decisions to coordinated management

The state transition function formulated in the previous section searches for the schedule that maximizes the stand's value. Such local (stand-scale) optimization represents a heuristic in that it does not guarantee optimality at the landscape level. The CA literature has suggested a number of methods to relate local, cell level decision-making to the global or higher-level management goals.

One option imposes constraints on the stand self-organizing behavior, so that not all transitions are available to each cell (Ward et al., 1999). Another method consists in coupling CA with an optimization methodology (Garriga and Ratick, 1996): before running the CA simulation, the number of stands that will make a state transition is determined using outcomes of the optimal land allocation problem. Finally, an approach, pursued further in this study consists of manipulating the updating procedure to meet the landscape objective (Strange et al., 2001, 2002).

Asynchronous updating is an alternative to synchronous updating where the transition rule is applied to only one chosen cell for each iteration of the algorithm. These updating schemes represent two extremes of the numerous updating schemes that can be employed. Intermediate schemes, between these two alternatives, would select, either systematically or randomly, a subset of cells to update. All of the iterative schemes involve various degrees of asynchronous updating. Depending on its design, the asynchronous updating will allow a stand's state to influence that stand's neighbors and, in turn, the changes in the environment (neighborhood) can influence further changes in the stand (Dieckmann et al., 2000). Lewontin (1961) refers to such changes as co-evolution. No updating procedure can guarantee landscape-wide optimality of the forest plan. However, asynchronous updating procedures where only a subset of cells is updated at once, may limit oscillatory behavior and foster the development of a stable configuration (final plan) (Huberman and Glance, 1993).

Strange et al. (2001) added randomness by using a probabilistic synchronous updating scheme. In this scheme, each stand, in each iteration, has a probability p of being updated. As a result, on average, only a proportion p of all stands will be updated. The updating of this subset of the stands is done synchronously. And finally, the probability p, of a stand being updated, decreases with each iteration. This results in fewer and fewer stands being updated with each iteration. This updating strategy has proved numerically efficient (Strange et al., 2002).

A similar approach developed in the parallel field of game theory, randomly selects a subset of players (cells) and imposes a random order on this subset. Each individual player is allowed to select his move when his turn comes (state transitions) (Dieckmann et al., 2000). The asynchronous updating provides opportunities for co-evolution towards a global objective by allowing stands to react to each other's decisions and the random order limits potential bias associated with the initial solution or the updating order (Alba and Tomassini, 2002).

To foster co-evolution towards a satisfactory global configuration, we therefore developed an algorithm which introduces a random, asynchronous updating scheme into a CA algorithm for solving multi-period forest planning problems (Mathey et al., 2005). At the beginning of each iteration, all of the stands are randomly ordered. Stands are updated, one at a time, following this random order. An iteration finishes when an updated stand has changed its state. Only one stand is changed per iteration. This process allows other stands in the neighborhood to experience a change and to react to it in turn. After a stand has been updated, its neighborhood or environment is prone to potential changes. If the stand is not updated, the next stand in the order is considered for update. At the onset of the iterative process, an improvement on the value of the stands initial random state is likely to occur and most of the stands selected would change their current states. However, after a number of iterations, fewer and fewer stands would change their state since an improvement of the stand value (i.e., a higher value) would become less likely. This would only be in response to changes in the states of their neighbors. The algorithm continues until either a maximum number of iterations have been reached or no change occurs within a single iteration.

5. Constrained forest planning

In this section, we consider the means of satisfying global constraints together with meeting landscape-wide objective(s). We expand the co-evolutionary optimization algorithm (Mathey et al., 2005) to handle multi-period planning problems with constraints.

As an illustration, we have added global harvest flow constraints together with a global minimum level of late-seral forest area in each period t. These requirements are formulated in the form of landscape-wide constraints:

$$\mathrm{vol}^t_{\min} \leq \mathrm{vol}_t(P) \leq \mathrm{vol}^t_{\max} \quad \text{and} \quad \mathrm{lsa}_t(P) \geq \mathrm{lsa}^t_{\min},$$
$$t = 1, 2, \ldots, T$$

Here $\mathrm{vol}^t_{\min}$ and $\mathrm{vol}^t_{\max}$ denote the respective minimum and maximum harvest volume (m^3) in period t over the planning horizon T, and $\mathrm{lsa}^t_{\min}$ denotes the minimum amount of late-seral forest area as the number of late-seral stands required in period t. The values $\mathrm{vol}^t_{\min}$, $\mathrm{vol}^t_{\max}$, and $\mathrm{lsa}^t_{\min}$ are constants reflecting policy. The values $\mathrm{vol}_t(P)$ and $\mathrm{lsa}_t(P)$ represent the actual volume and number of late-seral stands generated in period t under a forest plan P.

The forest management problem is to maximize the value $Z(P)$ of a forest plan, which is given by the sum of combined values from each stand, $\sum_{f \in F} z(s(f))$, subject to harvest flow constraints and a constraint on maintaining late-seral forest.

Optimizing the stand objective function does not always lead to an improvement of the landscape-level objective function and, especially, does not guarantee satisfying the landscape constraints. Penalty functions are a common means to guide centralized optimization procedures towards fulfilling constraints. However, in the context of a decentralized framework we need to define "penalty functions" that are efficient surrogates for the *global* (landscape) constraints within the *local* (stand) optimization process.

We incorporate violations of global level constraints as penalties or incentives that influence cell level choices. Let α_t, β_t, and γ_t measure the degree to which each of the constraints is violated in each of the planning periods. The factors α_t, β_t, and γ_t are derived from the forest wide levels of constraint satisfaction. In our model we have defined α_t, β_t, and γ_t to be

$$\alpha_t = \begin{cases} \dfrac{\mathrm{vol}^t_{\min}}{\mathrm{vol}_t(P)} & \text{if } \mathrm{vol}_t(P) \le \mathrm{vol}^t_{\min} \\ 0 & \text{otherwise} \end{cases},$$

$$\beta_t = \begin{cases} \dfrac{\mathrm{vol}_t(P)}{\mathrm{vol}^t_{\max}} & \text{if } \mathrm{vol}_t(P) \ge \mathrm{vol}^t_{\max} \\ 0 & \text{otherwise} \end{cases},$$

$$\gamma_t = \begin{cases} \dfrac{\mathrm{lsa}^t_{\min}}{\mathrm{lsa}_t(P)} & \text{if } \mathrm{lsa}_t(P) \le \mathrm{lsa}^t_{\min} \\ 0 & \text{otherwise} \end{cases}$$

These measures are used as period adjustment factors, which will discourage local level decisions that would contribute to a continuing violation of the global constraints. They act as penalty rates, in the case of β_t, and as incentive rates, in the cases of α_t and γ_t to modify the stand-level objective function. Eq. (3), the sum of the context-independent and context-dependent values associated with a stand's treatment schedule, becomes

$$z(s(f)) = \sum_{t=1}^{T}\left[(\lambda_t + \alpha_t - \beta_t) \times \frac{\mathrm{HV}_t(s(f))}{\mathrm{HV}_f} + (1 - \lambda_t + \gamma_t) \times \frac{\mathrm{LS}_t(s(f)) + \rho \times \mathrm{LSN}_t(f)}{\mathrm{LS}_f}\right] \tag{5}$$

The period adjustment factors augment the value of a treatment schedule in proportion to the amount that the schedule decreases a constraint violation.

The modified equation (5) is used for the evaluation of the stand value and thus modifies the initial transition function (Eq. (3)). The weighing factor λ (Eq. (3)) is internalized in each period objective value and becomes λ_t. The inclusion of the period adjustment factors modifies the period-specific values λ_t. From this perspective, a specific set of values for α_t, β_t, and γ_t modify the expression of the relative values of harvested timber versus late-seral forest area and late-seral clusters. For any combination of α_t, β_t, or γ_t, the CA, as an iterative scheme, co-evolves a plan. Although the cells make local level decisions, their decisions influence their neighbors' decisions and in turn their neighbors' neighbors. The CA requires a number of iterations to co-evolve a configuration of treatment schedules that represent an effective global solution for any given set of weights and adjustment factors. If penalty parameters change too often, stands do not have the opportunity to co-evolve a forest plan. This may potentially generate a forest plan far from the globally optimal plan. On the other hand, if penalty parameters change rarely, it is hard to find a feasible solution.

In recognition of this behavior, we further modify the constrained CA algorithm with an *ad hoc* iterative scheme that periodically recalculates the incentive/penalty rates. The intent is to freeze the policy environment and allow the algorithm to adjust to the set of penalties and incentives by running through a number of iterations. We group the iterations into successive blocks. Within a block of iterations, we can set the frequency at which we will recalculate our incentive and penalty parameters. In our implementation, we can adjust both the size of each block and the frequency of recalculation within each block. As an example, one block of iterations could be from iteration 2000 to iteration 4000. Within this block we would recalculate our measures of the constraint violations every 400 iterations. The frequency of recalculation would increase with successive blocks until, eventually, the expression of constraints α_t, β_t, and γ_t would be recalculated at every iteration.

The flowchart for the algorithm is presented in Fig. 1. Note that this algorithm can also be used for solving unconstrained planning problems as its special case.

6. Problem instance

The forest planning framework described above was applied to a test area from a forest located in the Northeastern forest region of Ontario, Canada. In this boreal region, the main forest type is dominated by black spruce (*Picea mariana*), but other distinct forest types and associations include poplar (*Populus* spp.), balsam fir (*Abies balsamea*), white birch (*Betula papyrifera*) and jack pine (*Pinus banksiana*), depending on the site. The test area is represented by an 18×27 grid of square stands where each stand is 9 ha with a total of 486 stands. The small size of the problem presented here allowed the generation of numerous outputs in a relatively short time and further allowed a comparison with other algorithms which would have been difficult with large-size problems. The resolution used in CA models and how it is adapted to the landscape greatly influence the results (Ménard and Marceau, 2005). However, Chen and Mynett (2003) found that the bias introduced by cell size in the final outputs was only problematic if the cell size was not relevant to the scale of decision variables. The 9 ha stand size was therefore selected because it is a functional size for representing silviculture operations in the study area since stands smaller than 10 ha typically get amalgamated to surrounding stands under the current inventory system of the area.

The unconstrained problem consists of generating a forest plan that maximizes the cumulative harvest volume while preserving continuous areas of late-seral forest over a 100-year planning horizon divided into 10-year periods. In the constrained problem, a stable harvest flow and a minimum amount of late-seral stands over time are also required. The forest plan consists of management treatments (states) assigned to all stands that make up the forest.

Each stand has an associated forest type and an initial age. Harvested stands can be treated with any one of four silvicultural regimes: no-management, extensive, basic, and intensive. The details and consequences associated with these different silvicultural regimes are a function of the stand type and the initial age that is associated with the stand. For each stand, the alternative treatment schedules consist of all possible feasible planning periods in combination with silvicultural regimes. Feasibility, in this case, reflects the harvest operability

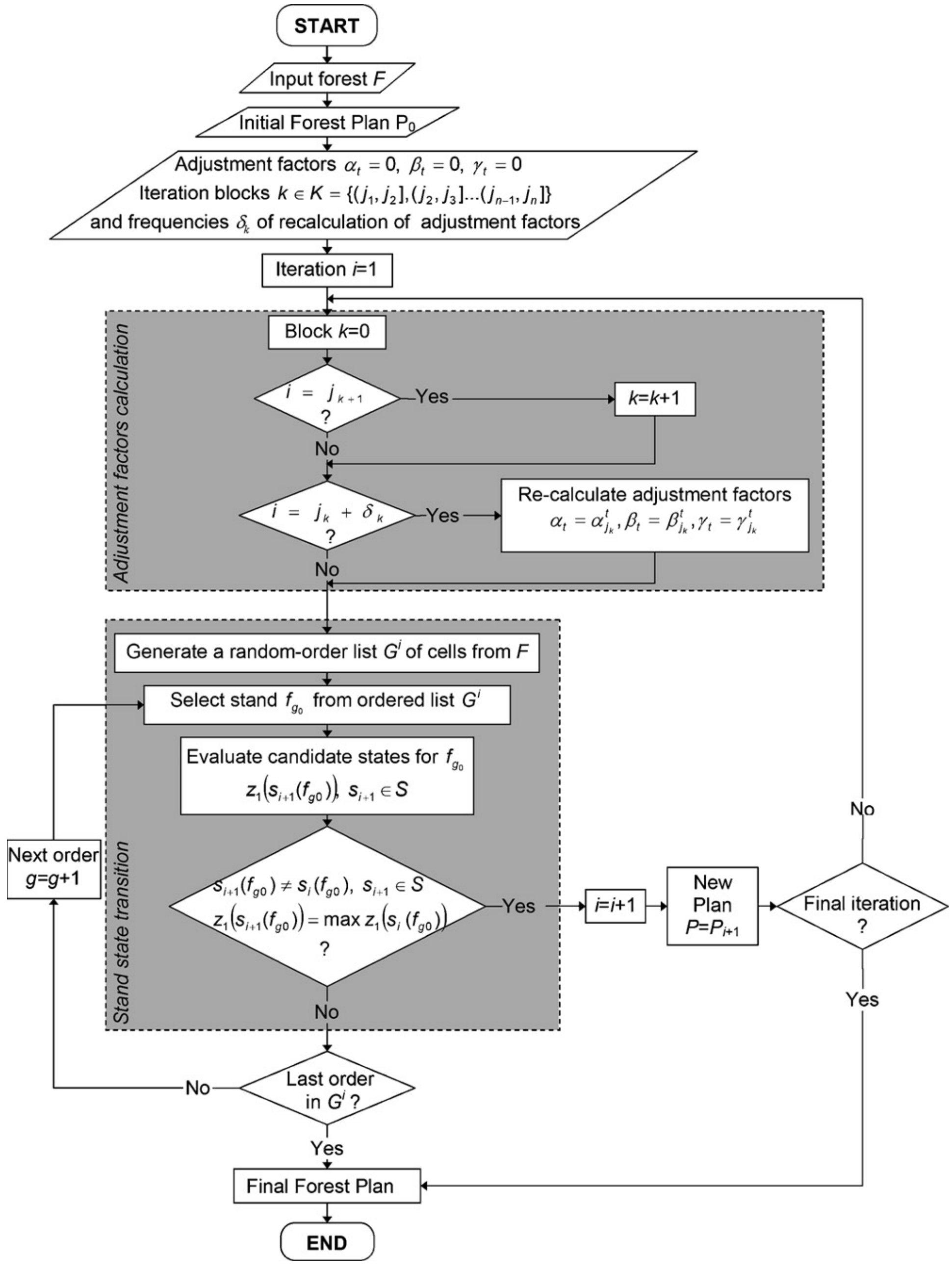

Fig. 1. A flowchart of the co-evolutionary algorithm for a constrained forest planning problem.

of the stand, which is a function of the stands type, initial age class and the applied silvicultural regimes. A stand's state corresponds to any one of the alternative possible combinations of harvest/silvicultural regime schedules. An initial forest plan is created by randomly assigning feasible states $s(\cdot)$ to stands $f \in F$.

The model is built upon the Arc/Info GIS software platform with a number of programs coded in Arc Macro Language (AML) and C++. The initial grid was populated with a forest inventory derived from corporate Natural Resource Values Information System (NRVIS) and Forest Resource Inventory (FRI) maps standardized by the Ontario Ministry of Natural Resources. The forest inventory was rasterized to 300 m × 300 m cells. For each raster dataset, the value assigned to one 9 ha-cell was generated by assigning it the value of the feature that fills the majority of the cell area (majority weighing method). The program was executed on a Personal Computer (PC) with a Pentium(R) 4 processor with a 1.3 GHz CPU speed.

7. Analysis of results for the unconstrained forest planning

In this section, we present the outcomes of the unconstrained problem obtained with the weight $\lambda = 0.4$ associated to the harvest volume. In order to assess the quantitative stability of the solution provided by the algorithm, the process was repeated 250 times, each time with a new randomly generated initial plan.

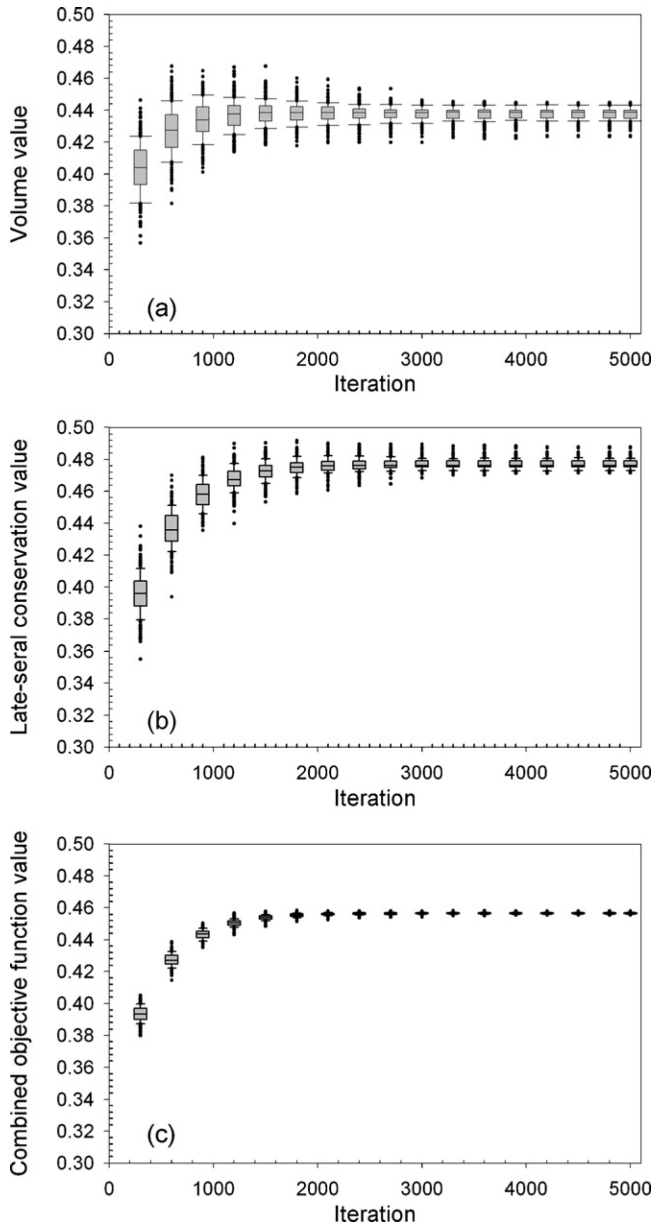

Fig. 2. Progression through iterations of the (a) harvest value, (b) late-seral conservation value and (c) combined objective value for the unconstrained problem. Box plots are produced every 300 iterations and summarize the results of 250 runs. Each box represents the median, 25th and 75th percentiles with the 10th and 90th percentiles as vertical lines. Dots represent the outliers.

Fig. 2a–c presents how the normalized stand volume, Eq. (1), the normalized late-seral conservation value, Eq. (2), and the combined treatment schedule value, Eq. (3), all progress over the algorithm iterations. The results are given for the whole forest as the normalized aggregation of all stands.

The harvest volume index rises until about iteration 1300, and then gradually decreases before converging to the volume generated by the final forest plan (Fig. 2a). The forest late-seral value increases throughout the entire iterative process (Fig. 2b), but its rate of change becomes much slower after iteration 1300. The forest objective function value follows a pattern similar to the forest late-seral conservation value (Fig. 2c).

The variation of the value associated with the plan cumulative volume and of the value associated with the late-seral conservation diminishes towards the end of the iterative process. The lower variation in the combined objective value curve reflects the expected covariance between cumulative harvest volume and late-seral values.

The number of iterations required for the algorithm to reach the highest possible combined value was found to be around 2000 iterations. Beyond 2000 iterations, no noticeable improvement of the combined value was apparent (Fig. 2c). In the initial iterations, the rapid improvement of the combined value is the result of increases in both cumulative harvest volume and OM index values.

Despite the lower weight ($\lambda = 0.4$) placed on harvest value, the harvest value rises until approximately iteration 1300. This behavior results from the initial random assignment of treatments to the stands. Stands with states that favor reaching late-seral stages, i.e. 'no-management' regime or regimes with no scheduled harvest, are in the minority; they are also scattered. The rare and scattered late-seral stands have a weak influence on their neighbors and contribute less to the global objective value. This has the implication that state transitions that increase harvest volume are initially favored. Once a certain amount of late-seral stands is built up in each planning period, the impact of grouping strengthens and it becomes increasingly worthwhile to choose a forest plan that preserves neighboring late-seral forest compared to generating harvest volume.

8. Analysis of results for the constrained forest planning

In this section, we present the model outcomes with constraints imposed on both the harvest flow and the amount of late-seral stands over time. The stable flow constraints require that the total harvest volume in each period be between 12,000 and 15,000 m^3. The late-seral constraint is defined as maintaining at least 41 stand (about 10% of the forested area) in the late-seral stage at all times.

The problem is also solved 250 times with $\lambda = 0.4$, for comparison with the unconstrained case above. The iterative process is run for 10,000 iterations. Incentives and penalties are periodically recalculated as described in Table 1.

The total harvest volume, Fig. 3a, and the total number of late-seral stands, Fig. 3b, in the first decade, are used to reveal the behavior of the algorithm. Comparable behavior is generated in the other decades.

Table 1
Incentive and penalty recalculation frequencies

Iteration count at the beginning of the block	Number of iterations between recalculations of incentives and penalties within the block
1	Initial 'burn in' period. Incentives and penalties are fixed at 0 with no recalculation
2000	400
4000	200
6000	100
9000	1

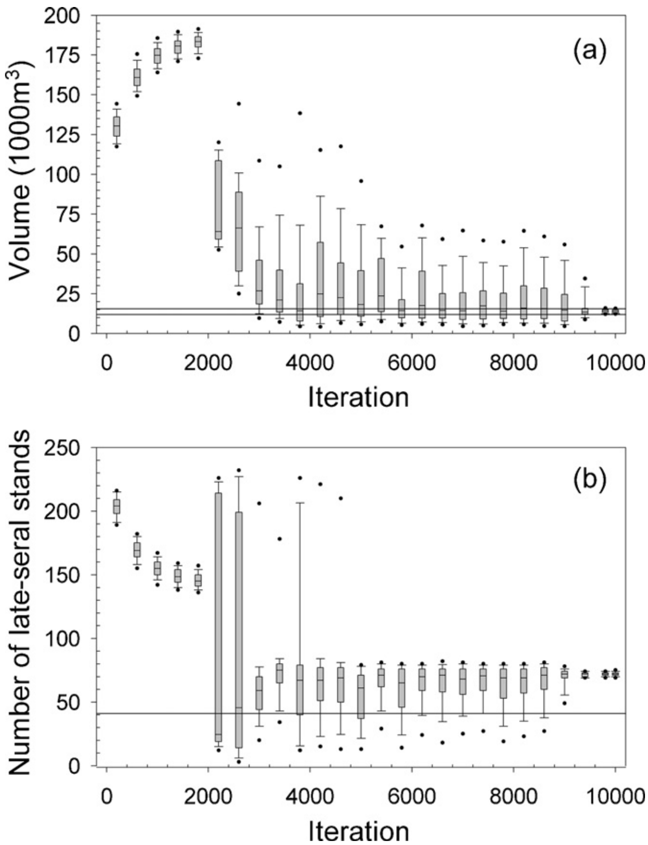

Fig. 3. Progression through iterations of (a) the volume harvested and (b) the amount of late-seral stands in the planning period 1. The lines represent the constraint levels. Box plots are produced every 400 iterations over 250 runs. Each box represents the median, 25th and 75th percentiles with the 10th and 90th percentiles as vertical lines. Dots represent the outliers.

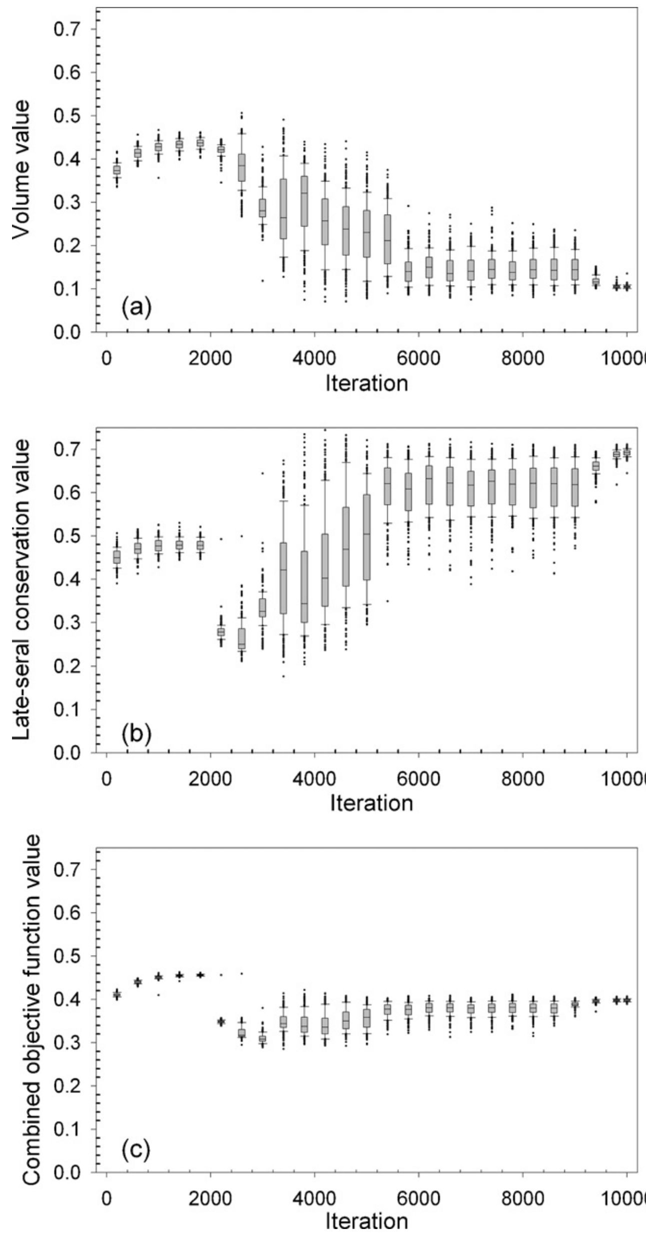

Fig. 4. Progression of (a) the harvest value index, (b) the late-seral conservation value and (c) the stand combined objective value through iterations when constraints are applied. Box plots are produced every 400 iterations over 250 runs. Each box represents the median, 25th and 75th percentiles with the 10th and 90th percentiles as vertical lines. Dots represent the outliers.

By the end of the first block of iterations the volume harvested in the first decade has reached 180,000 m^3. No incentives or penalties were applied during this set of iterations. There is a wide mix of schedules that produce harvest volume and provide late-seral conservation values. The large volume harvested in the first period exceeds the maximum flow constraint (Fig. 3a). This excessive volume generates a large penalty β_1 that will penalize stand decisions that harvest in the first period. On the other hand, at the end of the first block of iterations, the late-seral constraint is satisfied (Fig. 3b). The associated incentive parameter, γ_1, will be equal to zero. There are similar penalties and incentives associated with the other planning periods. Once introduced, the large penalty parameters attached to the harvest volume value will initially dominate the other values in the stand-level objective functions. In this situation the optimum stand-level decisions will generally be ‘no treatment’ and the introduction of the penalty parameter β_1 results in a significant drop in harvest volume over the 2000–4000 block of iterations (Fig. 3a). After 400 further iterations, the incentive/penalty rates are recalculated. Not all of the stands will have been ‘updated’ from their unconstrained solution. Some decades in the plan will have inadequate harvest volumes. Other decades may still have an excess harvest. The actual set of values for the incentive/penalty rates will depend on the initial inventory and the random order in which stands were selected for updating. The wide range of possibilities leads to a wide variation in first period harvest volume and an even wider variation in the number of first period late-seral stands in the plan (Fig. 3a and b).

As the algorithm progresses through the 4000–6000 block of iterations, the stands continue to self-organize in the new environment provided by the inclusion of incentive and penalty parameters. Variations in both the period harvest and number of late-seral stands decrease. By periodically recalculating the penalty parameters, both the volume and late-seral targets are progressively met across planning horizon and the output variations are reduced (Fig. 3a and b).

The progress of the harvest value, late-seral conservation value and combined value for the constrained case through iterations are presented in Fig. 4a–c.

Throughout the first 2000 iterations, the algorithm has not included the constraints and the results (Fig. 4) are identical to those of the unconstrained case (Fig. 2). Before the first introduction of the penalty parameters, the harvest volumes do not satisfy the flow constraints for most of the planning periods. They are either below the lower target or above the upper target. On the other hand, the weight $1 - \lambda$ associated with the late-seral value is sufficient to satisfy the target for old forest conservation in most periods. As in the period 1 discussed above, most of the penalty parameters γ_t associated with the late-seral constraint are therefore set to zero whereas those attached to the period volume (α_t and β_t) are high. The latter penalty parameters strongly influence the performance of the algorithm towards fulfilling volume constraints to the detriment of the value associated with late-seral conservation. This would explain why, over the next block of iterations (iteration 2000–4000) with recalculation of the incentive and penalty rates every 400 iterations, both the average harvest and late-seral conservation values drop below their best levels achieved in the first 2000 iterations.

The stand's harvest value index decreases as the iterative process progresses toward full satisfaction of the harvest flow constraints (Fig. 4a). The decrease in harvest due to the maximum flow constraint leads to the preservation of more late-seral forest. The late-seral conservation value rises way above its value in the unconstrained case (compare Fig. 2b to Fig. 4b).

The introduction of the constraints implies a large reduction of the combined value. There is a slow partial recovery as the algorithm sorts through the alternative plans. As expected, the constrained problem converges to a solution that is lower than the unconstrained plan (Fig. 4c versus Fig. 2c).

9. Analysis of tradeoffs between harvest volume and late-seral forest preservation

The approximate boundary on the set of feasible combinations of harvest volume and late-seral conservation value cumulated over the whole forest is identified and analyzed. This boundary, also known as the production possibility frontier, illustrates the tradeoffs between these two outputs for an efficient forest plan. The boundary shows how much the total harvest volume must be reduced to increase the total late-seral value by some amount and vice versa.

Among several ways to construct the production possibility frontier, we opted to run the model that maximizes the plans value, derived from Eq. (3), for different values of λ, the relative weight associated with the harvest value as opposed to the late-seral conservation value. Results of running both the constrained and unconstrained algorithms for different weights attached to harvest volume are presented in Fig. 5. This figure displays the sum over all forest stands of the late-seral forest conservation value against the total harvest volume for the best plan for a range of 11 values of λ between 0 and 1.

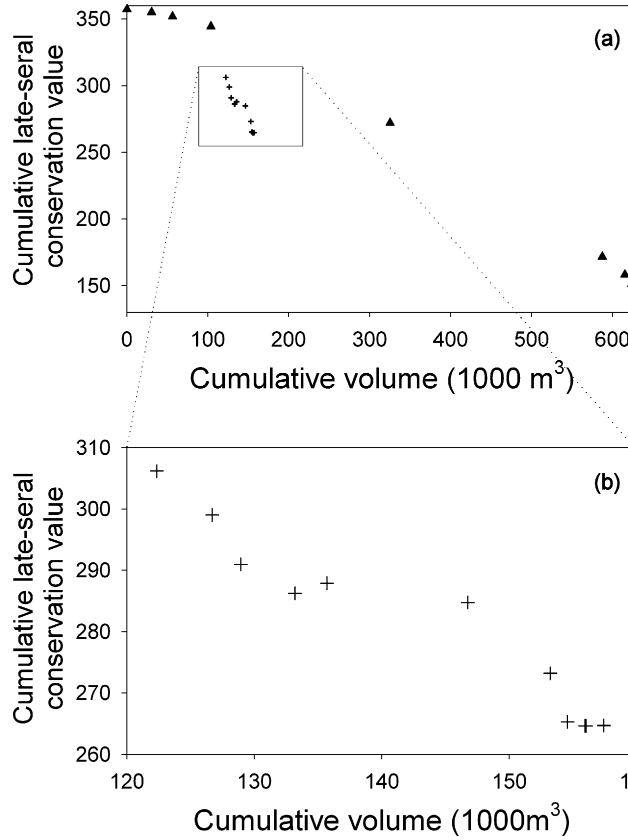

Fig. 5. Tradeoffs between cumulative harvest volume and the cumulative late-seral conservation value for (a) the unconstrained (▲) and constrained (+) co-evolutionary optimization tradeoffs and (b) the constrained problem (a close-up view).

The curve connecting the points for the unconstrained case is concave to the origin. When the weight associated to harvest volume is 1, the cumulative volume reaches 628,000 m^3. The value associated with late-seral forest on the landscape is not null because not all late-seral stands are located in manageable areas. Such stands cannot be cut and therefore contribute to the cumulative late-seral conservation value of the forest. Also, stands that cannot be harvested twice in the planning horizon can be retained in late-seral condition until the last period without a significant loss in harvest volume. Although these stands also contribute to the late-seral conservation value of the plan, this contribution is an artefact of the choice of time length for the planning horizon.

When the weight associated with volume is 0, there is no harvest; the corresponding harvest value is 0 while the cumulative late-seral value approximates 357. The tradeoff results are specific to the study area under consideration: the initial abundance of old forest and a priori exclusion of some areas from harvesting make it possible to satisfy provincial forest regulations and still gain significant harvest volume. Other problem instances may depict the tradeoff curves of different shapes.

When constraints are added to the model, they appear to bound the production possibilities to a roughly piece-wise linear envelope between 120,000 and 150,000 m^3 (Fig. 5). In the first section of this curve, stands are experiencing the low flow constraint in response to lower values of λ that favor late-seral forest conservation. In the upper part of the curve, the stands reach the upper flow constraint but tolerate minor violations in favor of harvest volume values given the high values of λ. In the flat, intermediate portion of the curve, the flow constraints are satisfied and the systems ‘jumps’ from the lower boundary on volume to the upper boundary on volume. Exceedance of the 150,000 m^3 total harvest occurs as a result of small violations of each periods maximum flow constraint that would be associated with the relatively low penalties compared to the relatively high λ. The small deviations from a linear tradeoff are artifacts of the heuristic algorithm and the discrete nature of the problem formulation (10-year periods).

10. Computational analysis and a comparison with simulated annealing

To evaluate the computational efficiency of the co-evolutionary optimization, the constrained planning algorithm presented in the previous section is solved by simulated annealing and the performance of both algorithms is compared. Simulated annealing is a random search technique whose objective is to converge to a steady optimal solution. Simulated annealing is commonly used in solving forest planning problems and has been showed to yield satisfactory results (e.g., Lockwood and Moore, 1993; Boston and Bettinger, 1999; Bettinger et al., 2002). For comparison with our CA algorithm, we run a simulated annealing with the same parameters as described by Strange et al. (2002).

For each algorithm, the solution process was repeated 250 times, using different initial forest plans. Once the simulated annealing algorithm reaches a feasible solution, which takes on average 15,510 iterations, we restrict the number of iterations to 5000, which makes the total of 20,510 iterations. The total number of iterations for the co-evolutionary optimization algorithm is 10,000, starting from the initial infeasible forest plan. The results of this experiment are presented in Table 2.

The comparison between simulated annealing and co-evolutionary optimization algorithm presented in this paper indicates that the latter algorithm can perform better in terms of both the computational speed, number of iterations and solution quality compared to the simulated annealing. The computation time is a function of the implementation framework and therefore may not reflect any significant improvement of the co-evolutionary CA algorithm. However it is interesting to notice its better performance. The improvement likely arises from the differences in how each algorithm treats constraints. The CA algorithm treats global requirements for volume and late-seral forest conservation as soft constraints and allows small deviations from the targets. In its formulation, the simulated annealing algorithm does not allow constraint violation, which may lessen its overall objective function value.

11. Conclusion

The CA-based algorithm developed in this paper successfully estimates some forest values and their long-term sustainability and appears adequate as a tool for long-term forest planning. It also demonstrates sensitivity to both local conditions and constraints as well as to strategic goals and constraints. The outputs compare well to those of simulated annealing, a more standard search technique used in forest management planning. Also, the approach proposed in this study makes the landscape an integral part of the decision-making process, ensures feasibility and generates outputs that meet objectives at different scales (local, focal, global). The spatial evolutionary game aspect of the solution algorithm allows the development of landscape strategies that are consistent with both the stand management objectives and landscape goals.

The application of the CA-based algorithm to a forest planning problem illustrates typical tradeoffs between conservation and timber values: an increase in late-seral forest conservation value makes harvested volume value decrease and vice versa. However, solutions generated over a range of weight combinations showed a non-linear relationship between the two objectives under consideration; a wide range of the cumulative volume may be coupled with similar late-seral conservation values. Producing high conservation values, however, is only possible within a very limited range of harvest volumes. The clustering component of the conservation value is the one that causes non-linearity of tradeoffs. The decentralized framework proposed in this paper provides a way to solve problems involving non-linear relationships.

The co-evolutionary approach offers an integrated perspective on management by addressing tradeoffs that occur across multiple temporal and spatial scales. It can address aspatial as well as spatial objectives across multiple time periods. This spatial aspect of the solving process is particularly interesting since the algorithm can solve not only for locational issues but also for clustering issues. These capabilities offers much potential for further research with a decentralized CA-based planning approach, which includes incorporation of other values and constraints such as net present value of the forest management plan and blocking of forestry operations to achieve economies of scale.

Table 2
Combined values and computation time for the simulated annealing (SA) and co-evolutionary CA (CE CA) optimization algorithms

Algorithm	No. of runs	Combined objective value			Average number of iterations			
		Minimum	Maximum	Average (S.D.)	To reach feasible solution	After feasible solution is reached	Total	Average computation time (min)
SA	250	0.3713	0.3859	0.3794 (0.0023)	15,510	5000	20,510	8.5
CE CA	250	0.3939	0.4055	0.39778 (0.0024)	–	–	10,000	7.1

Acknowledgments

The authors wish to acknowledge research support from the Natural Sciences and Engineering Research Council of Canada and the Sustainable Forest Management Network. The authors want to thank the Canadian Ecology Centre, Mattawa, Ontario and Tembec Inc. for providing data and support for the case study.

References

Alba, E., Tomassini, M., 2002. Parallelism and evolutionary algorithms. IEEE Trans. Evol. Comput. 6 (5), 443–462.

Bettinger, P., Sessions, J., Boston, K., 1997. Using Tabu search to schedule timber harvests subject to spatial wildlife goals for big game. Ecol. Model. 94, 111–123.

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

Boston, K., Bettinger, P., 1999. An analysis of Monte Carlo integer programming, simulated annealing, and tabu search heuristics for solving spatial harvest scheduling problems. For. Sci. 45, 292–301.

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

Chen, Q., Mynett, A.E., 2003. Effects of cell size and configuration in cellular automata based prey–predator modelling. Simul. Model. Pract. Theor. 11, 609–625.

Clements, S.E., Dallain, P.L., Jamnick, M.S., 1990. An operational, spatially constrained harvest scheduling model. Can. J. For. Res. 20, 1438–1447.

Colasanti, R.L., Grime, J.P., 1993. Resource dynamics and vegetation processes—a deterministic model using 2-dimensional cellular automata. Funct. Ecol. 7, 169–176.

Dieckmann, U., Law, R., Metz, J.A.J., 2000. The Geometry of Ecological Interactions: Simplifying Spatial Complexity. Cambridge University Press, New York.

Garriga, H.M., Ratick, S.J., 1996. Addressing the computational complexity of land use allocation in geographical information systems. In: Proceedings of the I International Conference on GeoComputation. September 17–19, 1996. University of Leeds, UK.

Green, D.G., 1989. Simulated effects of fire, dispersal and spatial pattern on competition within forest mosaics. Vegetatio 82, 139–153.

Harris, L.D., 1984. The Fragmented Forest: Island Biogeography Theory and the Preservation of Biotic Diversity. University of Chicago Press, Chicago, IL.

Hoganson, H.M., 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., Borges, J.G., Bradley, D., 1998. Techniques for addressing spatial detail in forest planning. Assessment of biodiversity for improved forest planning. European Forest Institute Proceedings no. 18.

Hubbell, S.P., Foster, R.B., 1986. Biology, chance, and history and the structure of tropical rain forest tree communities. In: Diamond, J., Case, T.J. (Eds.), Community Ecology. Sinauer, Sunderland, MA, pp. 314–329.

Huberman, B.A., Glance, N.S., 1993. Evolutionary games and computer simulations. Proc. Natl. Acad. Sci. U.S.A. 90, 7716–7718.

Krugman, P.R., 1996. The Self-organizing Economy. Blackwell Publishers, Cambridge, MA.

Lewontin, R.C., 1961. Evolution and the theory of games. J. Theor. Biol. 1, 382–403.

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

Lu, F., Eriksson, L.O., 2000. Formation of harvest units with genetic algorithms. For. Ecol. Manage. 130, 57–67.

Martell, D.L., Gunn, E.A., Weintraub, A., 1998. Forest management challenges for operational researchers. Eur. J. Oper. Res. 104, 1–17.

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

Ménard, A., Marceau, D.J., 2005. Exploration of spatial scale sensitivity in geographic cellular automata. Environ. Plann. Plann. Des. 32, 693–714.

Murray, A.T., Church, R.L., 1995. Heuristic solution approaches to operational forest planning problems. OR Spektrum 17, 193–203.

Nelson, J.D., 2003. Forest-level models and challenges for their successful application. Can. J. For. Res. 33, 422–429.

Ohman, K., 2000. Creating continuous areas of old forest in long-term forest planning. Can. J. For. Res. 30, 1817–1823.

Pukkala, T., Kurttila, M., 2005. Examining the performance of six heuristic optimisation techniques in different forest planning problems. Silva Fennica 39, 67–80.

Richards, E.W., Gunn, E.A., 2003. Tabu search design for difficult forest management optimization problems. Can. J. For. Res. 33, 1126–1133.

Schwartz, M.W., 1999. Choosing an appropriate scale for conservation efforts. Annu. Rev. Ecol. Systemat. 30, 83–108.

Seymour, R.S., Hunter Jr., M.L., 1999. Principles of ecological forestry. In: Hunter, Jr., M.L. (Ed.), Maintaining Biodiversity in Forest Ecosystems. Cambridge University Press, Cambridge, UK, pp. 22–61.

Shafer, C.L., 2001. Inter-reserve distance. Biol. Conserv. 100, 215–227.

Shands, W.E., Sample, V.A., LeMaster, D.C., 1990. National Forest planning: searching for a common vision. Critique of Land Management Planning, vol. 2. U.S. Department of Agriculture, Forest Service, Washington, DC.

Spies, T.A., Franklin, J.F., 1996. The diversity and maintenance of old-growth forests. In: Szaro, R.C., Johnston, D.W. (Eds.), Biodiversity in Managed Landscapes: Theory and Practice. Oxford University Press, New York, NY, pp. 296–314.

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

Strange, N., Meilby, H., Thorsen, B.J., 2002. Optimization of land use in afforestation areas using evolutionary self-organization. For. Sci. 48, 543–555.

Toffoli, T., Margolus, N., 1987. Cellular Automata Machines: A New Environment for Modeling. MIT Press, Cambridge, MA.

UNCED (United Nations Conference on Environment and Development), 1992. Agenda 21: the Rio Declaration on Environment and Development, the Statement of Forest Principles, the United Nations Framework Convention on Climate Change and the United Nations Convention on Biological Diversity. UNCED, Rio de Janeiro, June 3–14, 1992. United Nations Commission on Environment and Development Secretariat, Geneva.

Ward, D.P., Murray, A.T., Phinn, S.R., 1999. An optimized cellular automata approach for sustainable urban development in rapidly urbanizing regions. In: Proceedings of the IV International Conference on GeoComputation, Fredericksburg VA, USA, July 25–28.

Wolfram, S., 1994. Cellular Automata and Complexity: Collected Papers. Addison-Wesley, New York.