Dissertationes Forestales 34

# Developing spatial optimization in forest planning

Tero Heinonen

Faculty of Forestry
University of Joensuu

Academic dissertation

To be presented with permission of the Faculty of Forestry, University of Joensuu,
for public criticism in Auditorium BOR 155 of the University, Yliopistokatu 7, Joensuu,
on February 23th 2007, at 12 o'clock noon.

Title:
Developing spatial optimization in forest planning

Author:
Tero Heinonen

Series title and issue number:
Dissertationes Forestales 34

Thesis supervisor:
Prof. Timo Pukkala, Faculty of Forestry, University of Joensuu

Pre-examiners:
Prof. Howard M. Hoganson, University of Minnesota, Dept. of Forest Resources, St. Paul, MN, USA

Prof. Ljusk-Ola Eriksson, Swedish University of Agricultural Sciences, Dept. of Forest Resource Management and Geomatics, Umeå, Sweden

Opponent:
Prof. Jose Guilherme Borges, Department of Forestry Engineering, Instituto Superior de Agronomia, Lisbon, Portugal

ISSN 1795-7389
ISBN 978-951-651-156-9 (PDF)

Paper copy printed:
University of Joensuu, 2007

Publishers:
The Finnish Society of Forest Science
Finnish Forest Research Institute
Faculty of Agriculture and Forestry of the University of Helsinki
Faculty of Forestry of the University of Joensuu

Editorial Office:
The Finnish Society of Forest Science
Unioninkatu 40A, 00170 Helsinki, Finland
http://www.metla.fi/dissertationes

Heinonen Tero 2007. Developing spatial optimization in forest planning.
University of Joensuu, Faculty of Forestry

## ABSTRACT

Forest management planning problems typically includes many objectives and several planning periods, and the planning area consists of numerous forest stands. The solution space can be enormous and numerical methods are needed to solve problems. When there are spatial management objectives, problem becomes combinatorial in nature. The aim of this thesis is to develop methods to improve the performance of heuristic optimization methods in spatial forest planning problems and to compare the ability of different heuristics to solve different problems. Another aim is also to develop improved methods to solve spatial forest planning problems.

Traditional local search heuristic, when applied to forest planning, consider one stand at the time and change its treatment if it improves the solution. In this thesis treatment was changed treatment simultaneously in two stands which enlarges the solution space. This clearly improved the spatial layout of desired features and led to better objective function values especially with simple heuristics. The performance of local search methods is highly dependent on the parameters controlling the search. In this thesis an automated procedure to look for optimal parameters was developed. The method of Hooke & Jeeves for nonlinear programming was adopted in the search process. The method was able to find logical and efficient parameters for local search methods when the search time was limited.

Stand borders are traditionally subjectively drawn and fixed, and individual stands are assumed homogeneous in terms of forest characteristics. This can restrict the efficient use of forest resources. The interest in the use of fine-grained forest data is increasing the prospects of obtaining reliable data with remote sensing tools. This thesis deals with the implications and possibilities of using raster cells in forest planning. By using spatial objectives these cells were aggregated into dynamic treatment units. Spatial optimization and raster data produced more old forest area with the same timber production level than the approach based on predefined stands.

The computational burden of large planning problems can be reduced using decentralized computing methods. Instead of controlling the whole system with one objective function, a cellular automaton and a spatial application of the reduced costs method were used for decentralized optimization. The decentralized approaches reduced the solution space into a small fraction of the solution space of local search heuristics and decreased the time consumption of spatial optimization. The quality of the solutions also improved.

Keywords: cellular automata, dual price, fine-grained data, Hooke & Jeeves, local search heuristics, n-opt

## ACKNOWLEDGEMENTS

The study was partly funded by the Ministry of Environment (YM13/221/2004, grant to J. Kouki). I like also thank the Faculty of Forestry for providing facilities and funding.

My greatest gratitude goes to my supervisor Prof. Timo Pukkala. This thesis would'n be possible without him. I hope that I have adopted at least a part of his commitment and energy, and hopefully knowledge too. I want to thank Prof. Jari Kouki for taking me in his research group, and making this thesis possible. I wish to thank Docent Mikko Kurttila for his support and ideas. Also I would like to thank Dr. Jukka Matero, Dr. Olli-Pekka Tikkanen and Prof. Timo Tokola.

I want also thank the two pre-examiners, Prof. Ljusk Ola Erikson and Prof. Howard M. Hoganson, for their valuable comments.

Many other people have affected positively my studies and being here in Joensuu and at the university. Especially, I would like to thank Teemu, Kalle and Juhana for their help and friendship. Warm thanks go to my parents for providing a safe home base during the years. Finally, I want to thank my most love ones, Susanna and Emmi.

Joensuu, January 2007

Tero Heinonen

## LIST OF ORIGINAL ARTICLES

This thesis is a summary of the following Papers, which are referred to in the text by the Roman numerals I – V:

I Heinonen, T. & Pukkala, T. 2004. A comparison of one- and two- compartment neighbourhoods in heuristic search with spatial forest management goals. Silva Fennica 38(3):321-332.

II Pukkala, T. & Heinonen, T. 2006. Optimizing heuristic search in forest planning. Nonlinear Analysis: Real World Applications 7:1284-1297.

III Heinonen, T., Kurttila, M. & Pukkala, T. 2007. Possibilities to aggregate raster cells through spatial optimization in forest planning. Submitted manuscript.

IV Heinonen, T. & Pukkala, T. 2007. The use of cellular automaton approach in forest planning Submitted manuscript.

V Pukkala, T., Heinonen, T. & Kurttila, M. 2007. An application of the reduced costs approach in spatial forest planning. Submitted manuscript.

Heinonen was responsible for all the analyses in all studies. Heinonen was the main writer in Papers III and IV.

TABLE OF CONTENTS

# 1 INTRODUCTION

## 1.1 Background

The purpose of forest planning is to maximize the benefit of forest owner. Forest owner can have many conflicting objectives concerning the use and desired outputs of the forest. In addition, forest plans are usually produced to comprise many time periods and have constraints concerning the production and sustainable use of resources. Planning model is such a formulation of the planning problem that it can be solved with numerical optimization methods. The planning model must simultaneously consider all stands of the forest. The growing interest in ecological aspects in forest management has brought a wide range of objectives many of which are related to the relative locations of forest treatments or characteristics. These situations create very complex problems to solve (Weintraub et al. 2000).

Forest planning in Finland usually includes two stages. In the first, different treatment alternatives are produced for individual stands. Stands are homogenous forest areas that differ from adjacent areas in site or stand characteristics. Forest is a combination of stands. In the second stage the best treatment schedule is chosen for every stand, usually according to objective set at the level of whole forest. Treatment schedules are sequences of treatments over a planning horizon. In tactical forest planning individual stands must be treated according to one treatment schedule. This leads to a huge number of different combinations of stands' treatment schedules, even in a small forest holding (Pukkala 2002). Numerical methods are needed to go through these combinations and to evaluate different solutions in order to find the best solution according to given objectives.

Numerical optimization methods are used for determining the optimal allocation of scarce resources. They solve problems presented in a numerical model efficiently and reliably with a computer. The used optimization algorithm determines the formulation of the forest planning problem. For example, linear programming, integer programming, dynamic programming or heuristics can be adopted.

## 1.2 Optimization

Due to the nature of forest planning problems, only constrained optimization problems are considered in this thesis. An *optimization problem* is one where the value of a given function $R^n \rightarrow R$ is to be maximized or minimized over a given set $D \subset R^n$. The function $f$ is called the *objective function*, and the set $D$ the *constraint set*. These problems can be presented by

$$\text{Maximize } f(x) \text{ subject to } x \in D, \tag{1}$$

or

$$\text{Minimize } f(x) \text{ subject to } x \in D, \tag{2}$$

respectively. Problems of the first kind are called *maximization problems* and those of the second kind are called *minimization problems*. A solution to the problem $\max\{ f(x) \mid x \in D\}$ is a point $x$ in $D$ such that

$$f(x) \geq f(y) \quad \text{for all } y \in D. \tag{3}$$

In this case $f$ attains a maximum on $D$ at $x$. Similarly a solution to the problem $\min\{ f(x) \mid x \in D\}$ is a point $z$ in $D$ such that

$$f(z) \leq f(y) \quad \text{for all } y \in D. \tag{4}$$

In this case $f$ attains a minimum on $D$ at $z$. In the following text only maximization problems are considered.

The domain $D$ of $f$ is called the *search space* (feasible region), while the elements of $D$ are called *candidate solutions* or *feasible solutions*. A feasible solution that maximizes the objective function is called an *optimal solution*. Solution may fail to exist and there can exist more than one solution. When the *search space* or the objective function of the problem does not present convexity, there may be several local minima and maxima. A point $x \in D$ is a local (or relative) maximum of a function $f$ if there exists some $\varepsilon > 0$ such that $f(x) \geq f(y)$ for all $y \in D$ with $|x-y| < \varepsilon$.

## 1.3 Numerical methods

### *1.3.1 Exact methods*

There are two main ways to solve optimization problems, namely exact methods and heuristics. The most used way to model problems with exact method has been linear programming (LP). A linear program is the problem formulation for minimizing or maximizing a linear objective function in the decision variables, $x_1,\ldots,x_n$, subject to linear equality or inequality constraints on the $x_i$'s. A general LP problem presented in standard form is as follows:

Maximize

$$\sum_{j=1}^{n} c_j x_j \tag{5}$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \qquad \text{for } i=1,\ldots,m \tag{6}$$

$$x_j \ge 0 \qquad \text{for } j=1,\ldots,n, \tag{7}$$

where $a_{ij}$ is a matrix of known coefficients, and $c_j$ and $b_i$ are vectors of known coefficients. The above LP problem is called *primal*, and every LP problem can be presented as *dual* problem. The d*ual* problem can be stated as follows.

Minimize

$$\sum_{i=1}^{m} b_i y_i \tag{8}$$

subject to

$$\sum_{j=1}^{n} y_i a_{ij} \le c_j \qquad \text{for } j=1,\ldots,n \tag{9}$$

$$y_i \ge 0 \qquad \text{for } i=1,\ldots,m, \tag{10}$$

where $y_i$ are *dual (shadow) prices*. A *dual price* is the change in the objective function value of the optimal solution of an optimization problem obtained by relaxing the right hand side of the constraint by one unit. It is also referred to as a *dual variable*. In the *dual* problem, the objective function is a linear combination of the values of the $m$ constraints of the primary problem. There are $n$ dual constraints, each of which places a lower bound on a linear combination of $m$ *dual variables*. Each constraint in *primal* problem has a *dual price*. With every $x_j$ in any solution of the *primal* is associated a quantity known as the *reduced cost*. The *reduced cost* of a decision variable ($x_j$) is the amount by which the profit contribution of the variable must be improved before the variable in question would have positive value in an optimal solution.

With LP all the variables are continuous, but if the unknown variables are all required to be integers, then the problem is called an integer programming (IP) problem. 0-1 integer programming is the special case of integer programming where the variables are required to be 0 or 1 (rather than arbitrary integers). This can be handled with extra constraints as follows:

$$x_j \in \{0,1\} \qquad \text{for } j=1,\ldots,n \tag{11}$$

If only some of the unknown variables are required to be integers, the problem is called a mixed integer programming (MIP) problem. The simplex method is a efficient tool for

solving LP problems. If the problem formulation requires integer values, e.g. the branch and bound can be applied.

### *1.3.2 Heuristics*

A heuristic is a technique which seeks good (i.e. near-optimal) solutions at a reasonable computational cost without being able to guarantee either feasibility or optimality, or even in many cases to state how close to optimality a particular feasible solution is (Reeves & Beasley 1993).

One widely used branch of heuristics is local search (Aarts & Lenstra 1997). Local search methods are mainly used to solve combinatorial optimization problems. Combinatorial optimization is the process of finding one or more best (optimal) solutions in a well defined discrete problem space. An instance of combinatorial optimization problem is a pair $(D, f)$, where the objective function $f$ is a mapping $f: D \rightarrow R$. The problem is to find a globally optimal solution, i.e., an $i^* \in D$ such that $f(i^*) \geq f(i)$ for all $i \in D$. 0-1 integer programming problems can be considered as combinatorial optimization problems.

In theory the optimal solution could be obtained by total enumeration, i.e. calculating $f(s)$ for every $s \in D$. Small size problems can also be solved with IP solvers. Real-life problems can unfortunately be very large and impractical to solve with total enumeration or IP. The local search methods search only a small portion of solution space. They search the neighbourhood $N(s)$ of the current solution $s$ making small changes (moves) in current solution. If the change improves the objective function value, it is accepted. Local search optimization proceeds as follows (Dowsland 1993):

Select a starting solution $s_o \in D$;
Repeat
    Select $s$ such that $f(s) > f(s_o)$ using a suitable method;
    Replace $s_o$ by $s$;
Until $f(s) < f(s_o)$ for all $s \in N(s_o)$.
$s_o$ is the approximation to the optimal solution.

This simplest version of local search is called iterative improvement, and it will converge to the nearest local optimum. A poor local optimum can be avoided by letting the search process accept, every now and then, also moves which lead to inferior objective function values. With simulated annealing (Kirkpatrick et al. 1983) this is done by accepting inferior solutions with decreasing probability during the optimization. Threshold accepting (Dueck & Scheuer 1990) uses deterministic thresholds which are gradually lowered to 0. Tabu search (Glover 1989) accepts the best candidate move whether it is improving or deteriorating the solution, and prevents oscillation between neighbouring

solutions by using tabu-lists. All the problems that can be solved with exact methods can also be solved also with heuristic methods.

If the number of elements in the solutions of combinatorial problem are fixed and known, a neighbourhood can be defined as the set of solutions obtained by swapping a fixed number of elements in the current solution for the same number of non-solution elements. If $k$ items are swapped, these neighbourhoods are often referred to as $k$-neighbourhoods (also called $k$-opt), and the solutions obtained by using $k$-neighbourhoods are called $k$-optimal solutions. With large values of $k$ the chance to attain global optimum increases with increasing computational costs.

## 1.4 Forest planning and optimization

### *1.4.1 Non-spatial optimization*

One step of forest planning is to prepare a situation-specific model, which 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 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 (Pukkala 2002). Forest owner can have objectives at both the stand-level and the forest-level. Forest planning problems usually include forest-level goals, which make the problems difficult to solve.

Non-spatial problems are efficiently modelled as LP problems. Non-spatiality means that a treatment of a stand has no effect on the optimal treatment of other stands due to the location of stands. The requirement for even flow of timber is an example of traditional non-spatial forest-level objectives. Other forest-level 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. When using LP the forest planning problem can be modelled as follows:

Maximize

$$z = \sum_{j=1}^{N} \sum_{i=1}^{n_j} c_{ij} x_{ij} \tag{12}$$

subject to

$$\sum_{j=1}^{N} \sum_{i=1}^{n_j} a_{ijk} x_{ij} \geq / = / \leq B_k \qquad \text{for } k=1,\ldots,K_l \tag{13}$$

$$\sum_{i=1}^{n_j} x_{ij} = A_j \qquad \text{for } j=1,\ldots,n \tag{14}$$

$$x_{ij} \geq 0 \qquad \forall\, ij \tag{15}$$

where $z$ is objective function, $n$ is the number of stands, $n_j$ is the number of alternative treatment schedules in stand $j$, $x_{ij}$ is the area of stand $j$ treated according to schedule $i$, $B_k$ are constraints, $K$ is the number of constraints, and $A_j$ is the area of stand $j$. Coefficient $c_{ij}$ tells how much one hectare of stand $j$ treated according to schedule $i$ produces or consumes the objective variable, and coefficient $a_{ijk}$ indicates how much one hectare of schedule $i$ in stand $j$ produces or consumes constraining variable $k$. The decision (unknown) variables are the $x_{ij}$, i.e. stand areas treated with different methods.

If the dual prices of the dual problem for the above LP problem were known, the dual problem could be written as follows:

Maximize

$$w = \sum_{j=1}^{N} A_j u_j \tag{16}$$

Subject to

$$u_j \le c_{ij} - \sum_{k=1}^{K} a_{ijk} v_k \qquad \forall ij \tag{17}$$

$$u_j,\ v_k \text{ unsigned} \tag{18}$$

where $v_k$ is the dual price of constraining variable $k$ and $u_j$ is the dual price of stand $j$. With this problem formulation the forest-level planning problem reduces to a set of simple stand-level problems. The method of Hoganson and Rose (1984) is based on this fact. Dual prices are changed iteratively depending on the achieved level of forest-level constraints, and the stand-level problems are solved after every change. With the correct dual prices the problem can be solved optimally (Hoganson & Rose 1984).

If stands must not be partitioned, the LP problem is formulated so that the $x_{ij}$ are specified as 0 or 1 binary variables, and constraints (14) are converted into

$$\sum_{i=1}^{n_j} x_{ij} = 1 \qquad \text{for } j=1,\ldots,n \tag{19}$$

In this formulation, the decision variables are the proportions of stands treated according to different schedules, and the problem becomes a 0-1 integer programming problem. This modification requires that the coefficients $c_{ij}$ and $a_{ijk}$ are converted from per-hectare values into per-stand values. The problem becomes a combinatorial optimization problem, and IP or MIP solvers must be used to solve the problem optimally.

### *1.4.2 Spatial optimization*

Spatial aspects are getting more and more attention in forest planning, mainly because of increasing importance of ecological issues in forestry. In modern spatial optimization information about the neighbourhood relations of stands is integrated into the planning model, and the aspired spatial layout of resources is produced during optimization. The desired spatial layout is obtained by using spatial objectives or constraints. Spatial objectives require consideration of the relative locations of stands in optimization calculations. The approach of including spatial objectives in optimization represents endogenous planning (Kurttila 2001). When spatial considerations are imposed before optimization we are talking about exogenous approach. This thesis deals only with the endogenous approach.

Spatial problems in forest planning can roughly be separated into two categories: dispersing and connectivity problems (Öhman 2001). In dispersing problems the aim is to keep stands with certain characteristics apart from each other. The basic dispersing problem includes restrictions for clear cutting area and for the timing of clear cuttings (Brumelle et al.1997, Boston & Bettinger 2001); clear cuttings of certain size must not happen in adjacent stands in the set time frame. The unit restriction model (URM) and the area restriction model (ARM) have been developed to model the problem (e.g. Murray & Church 1995, Brumelle et al. 1997, Boston & Bettinger 2001). In URM the stand size corresponds to clear cut area restriction while in ARM the average stand size is clearly smaller than the allowable clear cut area. These are so called adjacency problems. URM can be modelled by using the following constraint (Borges et al. 2002).

$$x_{ij} + x_{km} \leq 1 \qquad \forall i, j, \forall k \in A_i, \forall m \in B_{i_j k} \tag{20}$$

where $A_i$ = set of all stands that have an adjacency relationship with stand $i$; $B_{i_j k}$ = set of options for stand $k$ that have an adjacency conflict with option $j$ for stand $i$. Clear cuttings could be dispersed also for example by minimizing the boundary between adjacent clear cut stands in the objective function.

In connectivity problems the aim is to aggregate stands with certain characteristics. A common connectivity problem in the Nordic Countries is to decrease the fragmentation of the forest (Öhman 2000, Öhman & Eriksson 2002). From the ecological point of view it would be advantageous to aggregate old forests. This could be achieved for example by minimizing the difference between the total old forest area and the total old forest core area (Öhman 2000) or by maximizing the boundary between adjacent old forest stands. Generation of continuous habitats for certain species has also received attention in spatial forest planning studies (Bettinger et al. 1997, Boston & Bettinger 2001, Kurttila et al. 2002, Hurme et al. 2006). Cutting area aggregation aiming at scale benefits is an example of economic spatial aggregation objectives (Öhman & Lämås 2003).

When spatial objectives are included in optimization, the objective function is no longer a linear combination of decision variables, and the stands must be treated only according to one treatment schedule. This makes spatial forest planning problems combinatorial in nature. IP and MIP solvers can be applied to small spatial problems, but real world problems can be very complex and large (Murray & Church 1995, Bettinger et al. 1999). The problem of large size can be alleviated e.g. by manipulating the structure of the constraints or by using heuristics (Torres-Rojo & Brodie 1990, Murray 1999). Heuristic techniques can deal with objectives with non-linear relationships and large amount of decision variables. The initial solution for heuristic search process is usually produced by selecting a random treatment schedule for individual stands. The heuristic algorithm then continues by changing randomly the treatment schedule of one or more (n-opt) stands until no improvements to the solution can be detected.

When heuristics are used to solve multi-objective spatial planning problems, the spatial component can be included in the objective function by using penalty functions (e.g. Lockwood & Moore 1993, Baskent & Jordan 2002, Öhman & Eriksson 2002). With the penalty function it is possible to measure the deviation of additional objective variable from their target level. The penalty function has the same unit as the objective variable. Multi-attribute utility function can also be applied to model spatial forest planning problems solved with heuristics (Pukkala & Kangas 1993, Kurttila et al. 2002). Sub-utility functions used within the utility function (objective function) transform the absolute value of the variable measured in its own units into a relative sub-utility value. Therefore variables included in the objective function do not have to have the same unit.

The most commonly used heuristics in spatial forest planning have been simulated annealing (Lockwood & Moore 1993, Baskent & Jordan 2002), tabu search (Bettinger et al. 1997, Richards & Gunn 2003) and genetic algorithms (Bettinger et al. 2002, Pukkala & Kurttila 2005). LP can be used in combination with the heuristics (Tarp & Helles 1997, Öhman & Eriksson 2002), and hybrid methods can also be composed of different heuristics (Boston & Bettinger 2002, Nalle et al. 2002). Dynamic programming has also been applied to solve forest level problems that involve adjacency constraints (Hoganson & Borges 1998). Sequential quenching and tempering is another method developed by Falcão & Borges (2002) for combining random and systematic modifications to the solution. Bettinger & Zhu (2006) presented a method that allow infeasibilities in a controlled manner.

## 1.5 Challenges in spatial forest planning

A proper functioning of heuristic local search methods highly depends on the parameters used. The quality of the solutions is sensitive to the parameter settings, which are often situation specific (Baskent & Jordan 2002). Traditional heuristics are also time-consuming. The result is often a compromise between solution time and solution quality. Very little has been done in the past to get more detailed and precise knowledge about the parameters

used. Usually parameters are determined by tedious tests (Bettinger & Zhu 2006). Knowledge about the magnitude of the effect of a single move on the objective function can be utilized (e.g. Öhman 2000). In addition, the quality of the parameters can not be easily evaluated. If a systematic and reliable process to seek for optimal parameters could be developed, much of the uncertainty concerning parameters could be avoided, and better quality solutions achieved.

The performance of heuristic local search methods could be improved also by enlarging the search neighbourhood, e.g. from 1-opt to 2-opt. With 2-opt moves it is possible to have smaller changes in the objective function and constraining variables, and thus allow more freedom for optimization near the border of feasible region (Bettinger et al. 1999, Bettinger et al. 2002). Moves with 2-opt neighbourhood for instance makes it possible to find improvements in the spatial layout without deteriorations in the non-spatial objectives and constraints. Search in a larger solution space unfortunately increases computational costs.

The ocular compartment inventory provides quite inaccurate data about the forest structure. Inventory is also expensive. Both the delineation and inventory of stands are subjective leading to large differences in data quality between surveyors (Haara 2003). The tendency is towards the use of fine-grained data. Remote sensing methods are being adopted to produce forest data for forest planning purposes. One promising method, laser scanning, is nowadays able to produce at least as good or even better quality data on some forest variables as the traditional method (Næsset 2002, Næsset et al. 2004).

When fine-grained data, like raster cells or micro-segments, are used, the problem arises of how to aggregate these small units into practical treatment units. One possibility is to treat the small units in optimization calculations in the same manner as traditional stands, and use spatial objectives to reach a desired landscape structure. The use of these so called dynamic treatment units leads to the abandonment of the traditional subjectively drawn stand borders. A raster approach should result in a more flexible and efficient utilization of the production potential of the forest (Holmgren & Thuresson 1997, Lu & Eriksson 2000).

Unfortunately the use of fine-grained data leads to a larger solution space and greater computational costs. To face this problem, new approaches to solve these large problems must be developed. One solution would be to adopt decentralized computing methods like cellular automata (CA) (Von Neumann 1966). CA are computing methods that are based on self-organizing systems capable of describing complex systems with simple rules. 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. 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, which interact with the given cell.

The dynamics of a cellular automaton is generated by repeatedly applying the local rule to all the cells on the grid. This can be done 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. CA reduces the solution space due to localized computing, and are therefore

more efficient computationally (Strange et al. 2002). In forest planning this means that optimization is performed at the stand level. CA have characteristics which make them suitable for spatial optimization (Strange et al. 2002, Ward et al. 2003, Mathey et al. 2005). The problem with localized computing is how to integrate forest-level objectives and constraints such as even-flow requirement to the calculations. Global objectives could be dealt with by adding a global objective component to the local objective function, and gradually increasing the weight of the global part until the global targets are met to a required degree. It is also possible to utilize the dual theory of linear programming at the stand level to tie the stand-level problems together and to meet the forest-level goals. The decentralized methods have often few parameters which makes them easy to use.

## 1.6 Aims of this thesis

The general aim of this thesis is to develop heuristics for spatial forest planning. The developed methods should be able to deal with multi-objective dynamic forest management problems including objectives that are related to the structure of forest landscape. Traditional heuristics are tested and improved to cope with complex spatial planning situations. Methods to deal with fine-grained forest data are developed, and alternative approaches to solve time consuming spatial forest planning problems with less computational costs are compared.

The specific objectives of the five studies of this dissertation were to:

i compare one- and two-compartment neighbourhoods in spatial utility maximization problems, using heuristics that are based on local neighbourhood searches (random ascent, Hero, simulated annealing and tabu search);

ii develop a method that can be used to optimize the search process of a heuristic algorithm (simulated annealing, threshold accepting, and tabu search) in a non-spatial and a spatial forest planning problem taking into account the solution time; inspect how sensitive the performance of the algorithm is to small changes in parameters, and how the optimal parameters are related to the problem size;

iii test alternative approaches to generate operative treatment units and aggregated old forest patches from raster cell data without predefined compartment boundaries; compare the efficiency of the traditional stand approach and the dynamic treatment unit approach;

iv develop and test a two-step cellular automaton heuristic in tactical forest planning; compare sequential and parallel updating methods in the developed cellular automaton;

compare the developed automaton to traditional methods (LP, simulated annealing);

v develop a cellular automaton method based on the dual price approach to manage spatial forest planning problems;
test the developed method with both polygon and raster data; and
compare the developed method to a cellular automaton and a traditional heuristic (simulated annealing).

# 2 MATERIALS AND METHODS

## 2.1 Test forests

In Study I the heuristic methods were tested in five test forests, four of which were real landscapes and one an artificial raster forest. The real forests are located in North Karelia, Finland. Their areas range from 700 to 981 hectares, and they consist of 608 to 803 compartments. The age class distribution of the stands is rather uniform with a small peak in 20–40-year-old stands. The raster forest consisted of 900 one-hectare square-shaped compartments, arranged in a grid of 30 rows and columns. The site and growing stock data of the compartments of the raster forest were taken from two real landscapes.

The raster forest was also used as a test forest in Study II which aimed at optimizing the heuristic search. To analyse the relationship between problem size and optimal parameters of heuristics, forests of 100, 500, or 1800 compartments were also used. These forests were created by deleting random compartments from the raster forest or taking copies of random compartments.

In Study III, a raster forest and a compartment forest of the same area were compared in a planning area located in North Karelia, Finland. The test area was 333.7 hectares, which included 10.1 ha of non-forest land. About 83 ha were young stands under 20 years of age and 112 ha carried mature forest over 80 years of age. The Forest Centre of North Karelia surveyed the forest applying ocular stand inventory. The surveyor divided the forest into 242 stand compartments. The raster forest was constructed by dividing the compartment forest into hexagon-shaped cells 721 $m^2$ in size (perimeter 100 m). The size of the hexagon was corresponded to typical cell sizes in raster-based forest inventory (e.g. Lind 2000; Lu and Eriksson 2000; Tuominen & Haakana 2005). Hexagons were used instead of squares to avoid single points of contact between neighbouring cells so as to make the determination of adjacent cells unambiguous. The resulting planning data included 4612 hexagons. The forest data for the hexagons were derived from the compartment inventory data by intersecting the compartments and the cell centres in a GIS application. The centre of the hexagon therefore determined the data source of a border cell.

Hexagons were used also in Study IV. The test forest was a hexagonal grid formed by a tessellation of regular hexagon cells. The grid consisted of 2500 cells, 50 cells in a row and

column. Each cell was one hectare in size. The forest data for the cells were derived randomly from actual forest stands locating in North Karelia, eastern Finland.

The test area in Study V was a 1134 ha forest area in southern Finland owned by UPM Kymmene. The forest consisted of 408 stands, and was inventoried with traditional stand-level inventory. The mean stand volume was 156.7 $m^3$/ha with standard deviation of 120.4 $m^3$/ha. About 30 % of the forest was younger than 20 years, and only 6 % carried mature forest over 80 years of age.

The same area was also used for a raster based comparison. The study area was split into 17798 cells (25 m x 25 m), 1112 ha altogether. Forest data were generated for the raster forest using k-nn method with Landsat7 ETM+ data (Tokola et al. 1996). Altogether 637 reference points (field plots) were used in estimation. The field data consisted of 472 measured sample plots from stands with volume over 100 $m^3$/ha and 165 artificial plots generated for young stands from the forest stand database using polygon centroid points as plot locations. Ten field plots with the shortest euclidean spectral distance were used in estimation. The age distribution of the raster forest was even more emphasized on young stands, 82 % being younger than 40 years old and only 2 % older than 60 years.

## 2.2 Search space

Monsu's automatic simulation tool was used to produce alternative treatment schedules for the stands or raster cells for 30- (Study II) or 60-year (Studies I, III, IV and V) planning period consisting of three 10- or 20-year time-periods. The simulation model was instructed to schedule a regeneration cut, accompanied by the necessary post-cutting treatments when the stand age reached the minimum regeneration age or the mean diameter reached the minimum 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 time-periods. The simulations were based on the silvicultural guidelines of the Forestry Development Centre Tapio (Luonnonläheinen … 1994), but the timing of regeneration cuttings was varied in order to obtain more than one treatment schedule per calculation unit. In addition, one simulated treatment alternative for mature stands was always the "no treatment" option.

The total number of treatment schedules (decision variables) in Study I ranged from 2986 to 4274 for the compartment forest, and was 4773 for the raster forest. In Study II the total number of treatment schedules was 4596. The total number of schedules in Study III was 1054 for the compartment forest and 15171 for the raster forest. The total number of schedules in Study IV was 12133, and in Study V 1866 for the compartment forest and 102075 for the raster forest. The number of schedules per stand or raster cell varied in all the studies. Some young stands had only one management option while dense stands approaching to maturity often had eight distinctly different management options.

## 2.3 Description of traditional heuristics used in the thesis

### *2.3.1 Random ascent*

In the random ascent heuristic used in the Study I, a set of initial solutions was produced by selecting a random treatment schedule for each stand from all schedules produced for it. The best random solution was the initial solution of random ascent, which first selects a random stand and then a random treatment schedule for the selected stand (one-compartment neighbourhood). If the selected schedule improved the objective function value it was included in the solution, otherwise it was not. With a two-compartment neighbourhood the algorithm was otherwise similar except that a move consists of changing the treatment schedule simultaneously in two compartments. The search procedure was stopped when the maximum number of trials was reached. To decrease problems arising from getting stuck to local optima, the whole process of generating an initial solution and applying random ascent could be repeated for a user-specified number of times. The parameters of the random ascent heuristic were: number of random searches to produce an initial solution, number of iterations (attempted moves) in random ascent, and number of repeated searches.

### *2.3.2 Hero*

In Hero heuristic (Pukkala & Kangas 1993) used in Study I the initial solution was also the best of a user-defined number of random solutions. Starting from the initial solution, the stands and their treatment schedules were explored sequentially to see whether another treatment schedule would improve the objective function value. If an increase was detected, the treatment schedule that improved the solution replaces the previous one. When all treatment schedules of all stands were examined in this way, the process was repeated until no schedules that would further improve the solution were found. The parameters of Hero are the number of random searches to produce an initial solution and the number of repeated searches. When a two-compartment neighbourhood was used the first compartment in which a change was made was selected in the same way as described above, i.e. sequentially, but the other compartment was selected randomly.

### *2.3.3 Simulated annealing*

Simulated annealing, used in Studies I, II, IV and V, used the best of a set of random combinations of stands' treatment schedules as the initial solution. It differs from the previous techniques in that it may also accept inferior solutions to avoid premature convergence to a local optimum (Dowsland 1993). A candidate move consisted of selecting first a random stand (or two stands if 2-compartment neighbourhood was used) and then a random schedule that would replace the current schedule of the selected stand. Moves that improved the objective function value were always accepted. Non-improving moves were

accepted with a probability of $p=\exp((U_{New} - U_{Old})\ T_i^{-1})$, where $T_i$ is the current "temperature", and $U$ is the objective function value. The "temperature" defines the probability of accepting a candidate solution poorer than the current solution. During the optimization process the temperature was gradually decreased so that at the end of the search the likelihood to accept inferior moves was close to zero. The temperature cooled according to a cooling schedule, which was implemented so that the temperature was multiplied with a multiplier less than one to get the next temperature. A certain number of candidate moves were tested in every temperature. The number of tested moves could change when temperature decreased; it was for instance possible to intensify the search as the process cooled. The search stopped when a user-specified stopping temperature was reached or a certain number of consecutive temperatures (five in the analyses of this study) went without any change in the solution. The parameters of simulated annealing were: number of random searches to produce an initial solution, starting temperature, cooling multiplier, freezing (stopping) temperature, number of iterations (attempted moves) in the initial temperature, and an iteration multiplier to get the number of iterations in the next temperature.

### *2.3.4 Threshold accepting*

Threshold accepting is a simplified version of simulated annealing. Threshold accepting simplifies the decision of whether or not to accept a candidate solution: all moves that produce a candidate equally good as or better than the current objective function value minus a threshold are accepted (Bettinger at al. 2002). The threshold has the same role as the temperature of simulated annealing. When using threshold accepting in Studies II and III the threshold was gradually reduced during the search, and a certain number of moves were tested with every threshold. The process was terminated once the threshold became very small (freezing threshold) or several consecutive temperatures (five in this study) went without any change in the solution. The parameters of threshold accepting were: number of random searches to produce an initial solution, initial threshold, threshold multiplier, freezing (stopping) threshold, number of iterations (attempted moves) with the initial threshold, and an iteration multiplier to get the number of iterations with the next threshold.

### *2.3.5 Tabu search*

Tabu search was used in Studies I and II. It searches the neighbouring solution space before accepting one change in the solution. The production of a set of candidate moves and accepting one of them is repeated for many iterations. Typical of tabu search are also tabu lists. In this thesis only recency-based lists that memorize the most recent moves, and prevent them for some time was used. Schedules that participated in the move were kept in the tabu list for a certain number of iterations. This number was the initial tabu tenure of the schedule. An iteration reduced the tabu tenures of all schedules by one. A schedule could again participate in a move when its tabu tenure had decreased to zero. The best non-tabu

move of the inspected candidates was accepted. If all candidates were in the tabu list the one with the shortest tabu tenure was accepted. If a candidate move would have yielded a solution better than the best obtained so far, it was accepted even if the move was tabu. The initial tabu tenure of a schedule that enters the solution was allowed to be different from the tabu tenure of a leaving schedule. Tabu search was stopped when a certain number of iterations were completed. Tabu search application used in this thesis was controlled through four parameters: number of iterations, number of candidate moves per iteration (given as percent of the total number of treatment schedules of stands), initial tabu tenure of a leaving schedule (exit tabu tenure), and initial tabu tenure of a schedule that enters the solution (entry tabu tenure).

## 2.4 Planning model for traditional heuristics

The Monsu software (Pukkala 2004) was used as the calculation platform in all the studies. All the planning problems solved with local search heuristics in Studies I-V were formulated as utility maximization problems as follows:

Maximize

$$U = \sum_{i=1}^{I} a_i u_i(q_i) \tag{21}$$

subject to

$$q_i = Q_i(\mathbf{x}) \qquad i = 1, \ldots, I \tag{22}$$

$$\sum_{k=1}^{N_n} x_{kn} = 1 \qquad n = 1, \ldots, N \tag{23}$$

$$x_{kn} = \{0,1\} \tag{24}$$

where $U$ is the total utility, $I$ is the number of management objectives, $a_i$ is the importance of management objective $i$, $u_i$ is a sub-utility function for objective $i$, and $q_i$ is the value of objective $i$. $Q_i$ is an operator that calculates the value of objective $i$, $\mathbf{x}$ is a vector of binary decision variables ($x_{kn}$) that indicate whether calculation unit $n$ is treated according to schedule $k$, $N_n$ is the number of alternative treatment schedules for unit $n$, and $N$ is the number of calculation units. With additive utility function all the solutions generated are feasible.

## 2.5 The cellular automaton method (Study IV)

In the cellular automaton developed in Study IV mutations and innovations occurring with decreasing probability changed the solution. First, a random treatment schedule was

selected for every cell (or stand), from among the set of previously produced schedules. Then the first cell was considered and a random number (U(0,1)) was drawn from a uniform distribution. If the random number was smaller than the current mutation probability, a random schedule of the same cell replaced the current schedule, i.e. a mutation occured. If there was no mutation, another uniform random number was generated and compared to the current innovation probability. If innovation occured, the best (locally optimal) schedule was searched for the cell, and it replaced the current schedule. Then, the next cell was inspected in the same way. Once all cells had been inspected, an iteration was completed, and a new iteration was begun with the first cell. Mutation and innovation probabilities were updated before starting the new iteration. This was repeated for a pre-defined number of iterations, or until no changes could be identified during an iteration or a certain number of iterations.

The algorithm was implemented in Study IV in both parallel and sequential ways. In the parallel mode all the mutations and innovations that were identified during iteration were made simultaneously at the end of iteration whereas the sequential mode executes the changes immediately with a consequence that, when a certain cell was inspected, it was known how the previous cells changed during the same iteration. In Study V, CA was used with the parallel updating rule.

The search process of the CA developed in Study IV was 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 vs. sequential). The probability of mutation depended on the initial probability, total number of iterations, and current iteration number:

$$P_M = P_M^0(1 - t/T)^{\tau_M} \tag{25}$$

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

$$P_I = P_I^0(1 - t/T)^{\tau_I} \tag{26}$$

where $P_I^0$ is the initial probability of innovation, and $\tau_I$ is an exponent greater than or equal to zero. When innovation did occur, the best treatment schedule was selected for the cell (or stand). Alternative schedules of cell $k$ were ranked with the following local objective function:

$$U_{jk} = \sum_{i=1}^{I} w_i u_i(q_{ijk}) \qquad j = 1,\ldots.n_k \tag{27}$$

where $U_{jk}$ is the 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 $i$, and $q_{ijk}$ is the quantity of objective variable $i$ in schedule $j$ of cell $k$, and $n_k$ is the number of schedules in cell $k$. The values of local objective variables depended on the cell only, or the cell and its neighbourhood. In forestry planning there are often goals and constraints, which cannot be met by stand-level optimization. Therefore, and additional step was added to the algorithm the purpose of which was to guarantee that the global objectives and constraints were met. In this step alternative schedules were evaluated using a function that has both a local and a global component:

$$R_{jk} = \frac{a_k}{A} U_{jk} + bP \qquad j = 1, \ldots, n_k \tag{28}$$

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. The global priority function was as follows:

$$P = \sum_{l=1}^{L} v_l p_l(g_l) \tag{29}$$

where $P$ is the global priority $L$ is the number of globally evaluated objectives, $v_l$ is the weight of global objective $l$ ($\sum v_l = 1$), $p_l$ is a priority function for objective $l$, and $g_l$ is the quantity of objective variable $l$. The global part of the algorithm begun with the final solution of the local optimisation. All schedules of all cells were sequentially evaluated for many iterations using Equation (28), and a better schedule always replaced the current schedule. The initial weight of the global priority function was zero, from which it was gradually increased, until the global priority reached a pre-defined value. The search also stopped when a predefined total number of iterations had been completed.

## 2.6 The dual price heuristic (Study V)

In Study V, planning problems were formulated for a spatial application of the dual price method (reduced costs method) proposed by Hoganson and Rose (1984) (later referred to as RC) as follows:

$$\max \; z = \sum_{l=1}^{L} w_l u_l(q_{ijl}) - \sum_{k=1}^{K} a_{ijk} v_k \tag{30}$$

where $v_k$ are heuristically up-dated "dual prices". The optimization process was executed at the stand level and had the following steps:

1. produce an initial solution
2. set initial dual prices
3. select the best schedule for every stand using Equation 30
4. calculate the values of forest-level goals (constraining variables)
5. up-date dual prices
6. repeat steps 3-5 until the forest level constraints are met

In spatial problems the selection of the best schedule for a given stand may alter the rankings of the schedules for stands optimized earlier. Because of this, the selection of the best schedules for stands (Step 3) needs to be repeated as many times as there are changes in stands' treatment schedules. When the initial solution always included the same treatment schedules for the stands (e.g. the first alternative of the stand) the method was deterministic.

## 2.7 Optimization of the parameters of heuristics (Study II)

In Study II the planning problem (equations 21 - 24) was solved using different parameters of the heuristic during a Hooke and Jeeves (1961) search. The Hooke and Jeeves is a nonlinear programming method, and it was used to search for the optimal parameter values of simulated annealing, threshold accepting, and tabu search. The Hooke and Jeeves method uses two types of searches. Exploratory search changes one parameter at a time trying to find its optimal value when the other parameters are kept constant. All parameters are inspected in this way. After completing a set of exploratory searches the method proceeds to pattern search in which several parameters are changed simultaneously. The "direction" of the pattern search (how much different parameters are changed) depends on the differences of parameter values in the beginning and at the end of the exploratory search. The step of changing a parameter is gradually decreased during the search. After completing the pattern search, another exploratory search is done, followed by another pattern search. This is continued until the step of changing a parameter is smaller than a user-specified convergence criterion for all parameters. Because there is no certainty that the Hooke and Jeeves method finds the global optimum, the optimization was repeated five times. The method was used to optimize the three heuristics in a non-spatial and a spatial planning problem, and with a short (later referred as quick) and long (later referred as slow) maximum computing time.

The objective function (*OF*) in Hooke and Jeeves optimization was the utility calculated for the solution found by the heuristic (Equation 21), minus time penalty:

$$OF = U - Penalty \tag{31}$$

Two different penalty functions were used for both non-spatial and spatial planning problems. To decrease the harmful effect of stochasticity to Hooke and Jeeves optimization the planning problem was solved ten times with each combination of parameters, and the mean objective function value was passed back to the Hooke and Jeeves algorithm.

## 2.8 Spatial variables used in the thesis

All the developed methods in this thesis were tested in spatial forest planning problems. The spatial objectives used were mainly based on the topology of stands across a landscape and indicate the extent to which they form patches. Aggregation of similar stands (in terms of treatments or stand features) was pursued by using the lengths of boundaries between adjacent similar stands as follows (Baskent & Jordan 1995):

$$Agr = \frac{total\ length\ of\ similar\ boundaries}{total\ length\ of\ boundaries} \tag{32}$$

A high ratio indicated aggregation across the landscape and low value indicated highly fragmented landscape. Maximization of equation (32) in the objective function aggregated stands of desirable characteristics and minimization did the opposite.
Spatial objectives based on similar stand boundary length used in this thesis were to aggregate cuttings (CCB, cut-cut boundary), flying squirrel habitats (FFB, flying squirrel-flying squirrel boundary) and old forests (OOB, old-old boundary). Cuttings were aggregated also by minimizing the total length of cut-uncut (CUCB) boundaries. Regeneration cuttings were dispersed by minimizing the boundary between adjacent stands or cells both cut during the same time period (Reg, regeneration cut-regeneration cut boundary). Spatial objectives used in different Studies are in Table 1.

**Table 1**. Spatial objectives used in different Studies (CCB = cut-cut boundary, FFB = flying squirrel-flying squirrel boundary, OOB = old-old boundary, CUCB = cut-uncut boundary and Reg = regeneration cut-regeneration cut boundary, MNM = mean of the neighbourhood minima).

| | Metric | | | | | |
|---|---|---|---|---|---|---|
| Study | CCB | OOB | CUCB | FFB | Reg | MNM |
| I | x | | | | | |
| II | x | | | x | | |
| III | x | x | x | | | x |
| IV | x | x | | | x | |
| V | x | x | x | | | |

The quality of a stand as a flying squirrel habitat patch was evaluated with a special habitat suitability index (Kurttila et al. 2002). The goodness of a stand 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 ($m^3ha^{-1}$). 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. Stands or cells with index value $\geq 0.5$ were considered old. This index value could be achieved for instance with a growing stock volume of 300 $m^3ha^{-1}$ and stand age of 70 (conifers) or 50 (broadleaved trees) years.

In Study III the functioning of the OOB objective was compared to another type of landscape metric. The purpose of this spatial objective, called as the mean of the neighbourhood minima (MNM), was to aggregate old forests. MNM was calculated so that a calculation unit first got the lowest old forest index value in its neighbourhood. The neighbourhood included the calculation unit itself and all adjacent units (i.e., those having common border with it). After this, calculation of the mean of the old forest index values produced the metric. Maximization of this metric aggregated old forests.

The sizes and shapes of landscape patches were measured after optimization using the Area Weighted Mean Shape Index (AWMSI) (McGarigal and Marks 1995) in Study III. When using the AWMSI, a circle represents the ideal shape for which the AWMSI gets a minimum value of 1. In Studies IV and V the core area of old forest was also used as a criterion because it takes into account both the total area and aggregation of old forest patches. 30-m buffer was used, which is the same as in Öhman (2000).

In all studies the planning problems also had non-spatial objectives. Non-spatial objectives aimed mainly at even-flow of timber and sustainability of cuttings. The weights ($a_i$) and sub-utility functions ($u_i$) of goals were formulated in such a way that the solutions differed only with respect to the spatial objective variables. This was achieved by setting high enough weights and low enough target levels for the non-spatial goals, and a high enough target level and a low enough weight for the spatial goal variable. Values higher than the target level no longer increased sub-utility, with the result that the values of non-spatial objective variables were almost exactly equal to the target level. The same principle was used with CA and RC in such a way that the solutions differed only with respect to the local objective variables.

# 3 RESULTS

## 3.1 A comparison of one- and two- compartment neighbourhoods in heuristic search with spatial forest management goals (Study I)

In Study I, the mean objective function values of the 20-repeated optimisations per heuristic method and planning problem were systematically higher with a two-compartment neighbourhood (Table 2). The standard deviations between repeated runs were very small compared to the differences between means, indicating that the differences between one- and two-compartment neighbourhoods were significant and not caused by the stochasticity of the optimisation algorithms. Of the four tested heuristics, simulated annealing and tabu search found better objective function values than random ascent and Hero, especially with a one-compartment neighbourhood. The improvements with a two-compartment neighbourhood were greatest with the simplest heuristics, random ascent and Hero. Thus, with a two-compartment neighbourhood the differences between methods were smaller than with a one-compartment neighbourhood.

**Table 3**. Mean and standard deviation of objective function value (Equation 21) in 20 repeated optimisations for different heuristics with one- and two-compartment neighbourhood, and the average time required for convergence.

| Variable | Random ascent | | Hero | | Simulated Annealing | | Tabu search | |
|---|---|---|---|---|---|---|---|---|
| | One | Two | One | Two | One | Two | One | Two |
| | | | | Area 1 | | | | |
| Mean | 0.9417 | 0.9465 | 0.9445 | 0.9548 | 0.9568 | 0.9594 | 0.9539 | 0.9582 |
| Standard dev. | 0.0021 | 0.0014 | 0.0013 | 0.0010 | 0.0010 | 0.0014 | 0.0016 | 0.0013 |
| Time, s | 45 | 76 | 71 | 270 | 54 | 106 | 61 | 341 |
| | | | | Area 2 | | | | |
| Mean | 0.9247 | 0.9240 | 0.9235 | 0.9345 | 0.9378 | 0.9390 | 0.9353 | 0.9389 |
| Standard dev. | 0.0011 | 0.0012 | 0.0014 | 0.0008 | 0.0008 | 0.0009 | 0.0014 | 0.0009 |
| Time, s | 34 | 55 | 52 | 143 | 40 | 82 | 50 | 290 |
| | | | | Area 3 | | | | |
| Mean | 0.9466 | 0.9548 | 0.9565 | 0.9675 | 0.9664 | 0.9689 | 0.9629 | 0.9692 |
| Standard dev. | 0.0021 | 0.0013 | 0.0020 | 0.0009 | 0.0016 | 0.0018 | 0.0024 | 0.0015 |
| Time, s | 34 | 56 | 55 | 283 | 44 | 86 | 52 | 238 |
| | | | | Area 4 | | | | |
| Mean | 0.9482 | 0.9560 | 0.9487 | 0.9673 | 0.9673 | 0.9704 | 0.9619 | 0.9694 |
| Standard dev. | 0.0016 | 0.0015 | 0.0024 | 0.0013 | 0.0018 | 0.0016 | 0.0022 | 0.0015 |
| Time, s | 56 | 89 | 113 | 423 | 64 | 122 | 70 | 452 |
| | | | | Raster forest | | | | |
| Mean | 0.9397 | 0.9443 | 0.9404 | 0.9556 | 0.9588 | 0.9612 | 0.9567 | 0.9590 |
| Standard dev. | 0.0014 | 0.0007 | 0.0012 | 0.0006 | 0.0006 | 0.0008 | 0.0010 | 0.0009 |
| Time, s | 68 | 107 | 119 | 452 | 79 | 137 | 86 | 560 |

The use of a two-compartment neighbourhood always lengthened the optimisation time (Table 2), because the evaluation of the effect of changes in two compartments takes more time than evaluating only one change. Another reason is that because a two-compartment neighbourhood is better in finding improving moves, optimisation continues longer. Generally, a one-compartment neighbourhood improved the objective function value faster in the beginning of the search, but improvements continued for much longer with a two-compartment neighbourhood.

Another problem formulation in Study I, covering only one 20-year period, supported the results obtained with the first objective function: the use of a two-compartment neighbourhood improves the attainment of the spatial objective. However, the benefit depended on the problem formulation. It seems that the benefit of using two-compartment neighbourhood increases when the problem becomes more difficult.

## 3.2 Optimizing heuristic search in forest planning (Study II)

### *3.2.1 Optimal parameters of simulated annealing*

The three best quick solutions of simulated annealing for the non-spatial planning problem were close to each other in terms of the number of initial random searches, initial temperature, and the number of iterations in the initial temperature (Table 3). The three best solutions for the spatial problems were reasonably close to each other in terms of the initial and freezing temperatures, initial random searches, and number of iterations in the initial temperature. The cooling and iteration multipliers affect most the duration of the search of simulated annealing. The optimal cooling multiplier varied from 0.898 to 0.968, a high multiplier resulting in slow cooling. The iteration multiplier varied from 1.018 to 1.087, and suggests that it is optimal to intensify the search as a function of cooling, by having 1.8 to 8.7 % more iterations in every new temperature.

A comparison of the three best solutions for the same problem with the same time penalty function reveals (quick or slow in Table 3) that slow cooling and intensive search are alternative strategies: if the cooling multiplier is decreased the iteration multiplier or the number of initial iterations increases, and vice versa. It seems that, in the non-spatial problem, it is optimal to increase the iteration multiplier when the available search time increases. In the spatial problem, on the other hand, slower cooling was often used when the allowed search time increased. In simulated annealing the optimal freezing temperature decreased with increasing number of compartments but the initial temperature remained the same. Cooling was faster when the problem size increased but the solution time was not continued. The number of initial iterations and the iteration multiplier were not systematically related to the problem size.

**Table 3**. Parameter values of the simulated annealing heuristic in three best solutions (I, II, III) for a non-spatial and a spatial planning problem with a quick or slow convergence requirement.

| Parameter of the heuristic method | | Non-spatial | | Spatial | |
|---|---|---|---|---|---|
| | | Quick | Slow | Quick | Slow |
| Initial random searches | I | 2 | 1 | 33 | 162 |
| | II | 1 | 100 | 21 | 47 |
| | III | 1 | 142 | 27 | 121 |
| Initial temperature | I | 0.0019649 | 0.0024651 | 0.0003954 | 0.0028192 |
| | II | 0.0018990 | 0.0028230 | 0.0011115 | 0.0013606 |
| | III | 0.0022806 | 0.0013985 | 0.0019068 | 0.0013934 |
| Cooling multiplier | I | 0.920 | 0.938 | 0.947 | 0.919 |
| | II | 0.953 | 0.937 | 0.919 | 0.947 |
| | III | 0.938 | 0.920 | 0.898 | 0.968 |
| Freezing temperature | I | 0.0000070 | 0.0000153 | 0.0000277 | 0.0000177 |
| | II | 0.0000001 | 0.0000161 | 0.0000177 | 0.0000286 |
| | III | 0.0000001 | 0.0000045 | 0.0000242 | 0.0000316 |
| Initial iterations | I | 191 | 218 | 118 | 257 |
| | II | 157 | 215 | 117 | 118 |
| | III | 113 | 70 | 262 | 122 |
| Iteration multiplier | I | 1.047 | 1.066 | 1.087 | 1.087 |
| | II | 1.018 | 1.064 | 1.087 | 1.087 |
| | III | 1.028 | 1.097 | 1.082 | 1.045 |

### *3.2.2 Optimal parameters of threshold accepting*

In Study II, the optimal initial threshold of threshold accepting was not related to problem type and solution time, but was often near 0.001 (Table 4). The freezing threshold was also reasonably constant, of a magnitude of 4 % of the initial threshold. The optimal threshold multiplier varied little being mostly about 0.95 and in a few cases somewhat higher. The results indicate that the optimal sequence of thresholds is not strongly correlated with the problem type and requirements concerning the convergence time. Instead, the search should be intensified when more time is allowed.

**Table 4**. Parameter values of the threshold accepting heuristic in three best solutions (I, II, III) for a non-spatial and a spatial planning problem with a quick or slow convergence requirement.

| Parameter of the heuristic method | | Non-spatial | | Spatial | |
|---|---|---|---|---|---|
| | | Quick | Slow | Quick | Slow |
| Initial random searches | I | 13 | 54 | 2 | 84 |
| | II | 1 | 24 | 58 | 91 |
| | III | 60 | 63 | 1 | 63 |
| Initial threshold | I | 0.0009258 | 0.0015482 | 0.0012789 | 0.0013650 |
| | II | 0.0003016 | 0.0005264 | 0.0011361 | 0.0005106 |
| | III | 0.0012866 | 0.0017226 | 0.0010024 | 0.0009225 |
| Threshold multiplier | I | 0.965 | 0.944 | 0.957 | 0.952 |
| | II | 0.978 | 0.957 | 0.953 | 0.976 |
| | III | 0.949 | 0.941 | 0.946 | 0.963 |
| Freezing threshold | I | 0.0000422 | 0.0000293 | 0.0000385 | 0.0000428 |
| | II | 0.0000245 | 0.0000276 | 0.0000404 | 0.0000486 |
| | III | 0.0000454 | 0.0000001 | 0.0000761 | 0.0000172 |
| Initial iterations | I | 79 | 109 | 61 | 85 |
| | II | 101 | 136 | 25 | 60 |
| | III | 76 | 163 | 124 | 49 |
| Iteration multiplier | I | 1.046 | 1.087 | 1.055 | 1.093 |
| | II | 1.029 | 1.089 | 1.081 | 1.066 |
| | III | 1.057 | 1.044 | 1.094 | 1.060 |

Compared to the optimal parameters of simulated annealing, threshold accepting used fewer initial iterations but had nearly same iteration multiplier. The optimal threshold multiplier of threshold accepting was a bit higher than the optimal cooling multiplier of simulated annealing. The initial threshold of threshold accepting did not depend systematically on the problem size. The freezing temperature, on the other hand, decreased systematically as the number of compartments increased. The number of initial random searches and the threshold multiplier tended to decrease when the number of compartments increased.

### *3.2.3 Optimal parameters of tabu search*

In tabu search, the maximum number of iterations varied from 893 to 2907 (Table 5). It was mostly around 1000, and not systematically related to the problem type or allowed search time. The number of candidates inspected per iteration was about 1 % of the neighbourhood in quick optimization (in two cases only 0.1 %) but 4 to 11 % if more time was available.

**Table 5**. Parameter values of the tabu search heuristics in three best solutions (I, II, III) for a non-spatial and a spatial planning problem with a quick or slow convergence requirement of the heuristic.

| Parameter of the heuristic method | | Non-spatial | | Spatial | |
|---|---|---|---|---|---|
| | | Quick | Slow | Quick | Slow |
| Iterations | I | 1299 | 1492 | 1608 | 2907 |
| | II | 1053 | 999 | 946 | 1732 |
| | III | 1313 | 893 | 985 | 1025 |
| Percent of candidates | I | 1.141 | 7.045 | 0.100 | 3.653 |
| | II | 1.432 | 10.000 | 1.692 | 5.644 |
| | III | 0.100 | 11.610 | 1.582 | 10.412 |
| Exit tabu tenure | I | 41 | 37 | 58 | 116 |
| | II | 49 | 59 | 13 | 1 |
| | III | 60 | 80 | 1 | 34 |
| Entry tabu tenure | I | 29 | 27 | 6 | 9 |
| | II | 36 | 25 | 1 | 42 |
| | III | 39 | 42 | 27 | 34 |

The optimal tabu tenure for a schedule that is removed from the solution was often about 50 iterations. Around 50 % shorter tabu tenures should be used for schedules that enter the solution. Out of the four parameters the number of candidate moves was most systematically related to the convergence time requirement.

In tabu search, the first way to keep the same solution time with increasing number of compartments was to decrease the percentage of candidate moves. When this could no longer be done the number of iterations was decreased. Both the exit and entry tabu tenures seemed to increase as a function of problem size. The optimal entry tabu tenure was always clearly shorter than the optimal initial tabu tenure of a leaving schedule if the problem was not very small.

### 3.3 Possibilities to aggregate raster cells through spatial optimization in forest planning (Study III)

In Study III, the area of old forest was larger in the raster forest than in the compartment forest in all the plans except one for the first and second 20-year time-period (Figure 1). The mean size of old forest patches was larger in the compartment forest in all the plans because there were some very small patches in the raster forest. Taking into account that the existence of additional small patches is not a deficiency, it is noteworthy to mention that e.g. in the plan where OOB was used for the raster forest the total area of 11 (number of patches in the corresponding plan for the compartment forest) largest old forest patches was

33 % larger and their mean size 34 % higher than in the compartment forest. Therefore, the existence of some very small old forest patches affects much the mean patch size statistic for the raster forest.

The AWMSI values were lower (better) for the raster forest in all the plans except the plan where only cuttings were aggregated at the end of the third time-period. In both forests the total area and the mean size of old forest patches were at their highest when the OOB was used as the spatial objective. The use of the OOB as an objective resulted in clearly larger areas of old forest than the use of MNM. On the other hand, the use of the MNM led to slightly lower shape index values (AWMSI) in the raster forest and thus in old forest patches that were more compact in shape.

The old forest index in the raster forest revealed clear differences in the functioning of the OOB and MNM metrics. In all plans the area of old forest (OFI > 0.8) increased from the initial situation. The difference between the OOB and MNM objectives was that while the OOB produced more old forest (OFI > 0.8) the MNM resulted in less abrupt edges: old forests patches were often surrounded by areas with only slightly lower OFI values when MNM was used. This means that the MNM produced a smoother landscape.

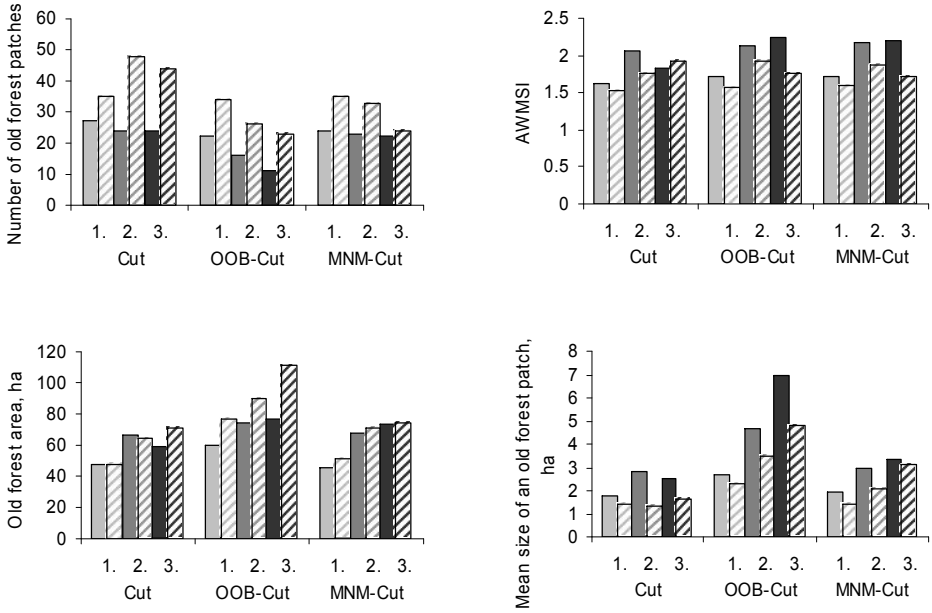


**Figure 1**. Variables describing old forest patches in different plans. Solid shadings represent compartment forest and line patterns represent raster forest. Numbers below the x-axis indicate the time-period (Cut = CCB and CUCB used; OOB-Cut = CCB, CUCB and OOB used; MNM-Cut = CCB, CUCB and MNM used).

The total number of cutting areas and their mean size did not depend systematically on the use of compartment vs. raster data (Figure 2). The AWMSI was lower for the compartment forest during the first time-period in the plans where OOB was used and where only cuttings were aggregated. Otherwise the AWMSI was slightly lower for the raster forest. Maximizing CCB and minimizing CUCB worked better in combination with the MNM objective than with the OOB objective: the total number of cutting areas was smaller and the mean size of the cutting area larger. This can be explained by the nature of the MNM objective, which (in addition to clustering old forest) also attempts to cluster poor stands in terms of old forest index, i.e. cutting areas.

Visual examination of the plans revealed that neighbouring raster cells formed feasible cutting areas (Figure 3). The cutting areas did not always follow the compartment boundaries, even though the forest data of raster cells were derived from the compartment forest. The old forest patches formed from raster cells were good in size and shape and frequently in different places than in the compartment forest. The raster approach frequently generated old forest patches that did not follow the compartments boundaries.

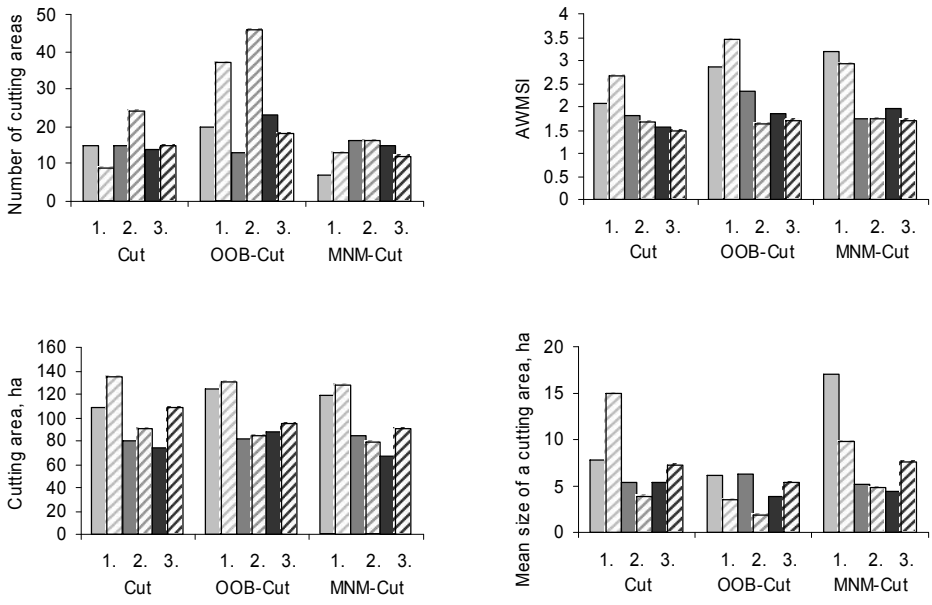


**Figure 2**. Variables describing cutting areas in different plans. Solid shadings represent compartment forest and line patterns represent raster forest. Numbers below the x-axis indicate the time-period (Cut = CCB and CUCB used; OOB-Cut = CCB, CUCB and OOB used; MNM-Cut = CCB, CUCB and MNM used).

**Figure 3**. Old forest patches at the end of the 60-year planning period in the raster forest (dark grey) and cutting patches during the third 20-year time-period (light grey). CCB, CUCB and OOB used on the left and CCB, CUCB and MNM used on the right.

The trade-off curve calculated in Study III for harvested volume and old forest area confirmed, together with other results, that the raster approach was able to generate more old forest than the compartment approach. At all cutting levels the old forest area was larger in the raster forest. This verifies that the raster approach enables a more efficient joint production of old forest and harvested timber. At the end of the 60-year planning period the raster approach produced at most about 25 hectares (14%) more old forest than the compartment forest with the same total harvest.

### 3.4 The use of cellular automaton approach in forest planning (Study IV)

Study IV first tested the effect of parameter values on the performance of the cellular automaton (CA). The tests 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. The mutation probability had to be low, but otherwise the mutation probability did not affect the solution quality. Also the parallel mode was insensitive to parameter values 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 is zero because otherwise the cells may oscillate between two states. 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. Based on the sensitivity analyses, parameters in Table 6 were used in subsequent analyses.

Study IV then compared the performance of CA to linear programming (LP) in four non-spatial problems and to simulated annealing (SA) in three spatial problems. In all the four non-spatial problems the constraints were fulfilled with both LP and CA. Thus differences between the methods emerged only in the net present value which was the objective variable of the LP problems (and one objective variable in the CA problem formulations). In three problems the net present value obtained with CA was within 99.6 % of the maximum, and 97.9 % in the most difficult problem. The differences can be partly 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.

The CA was superior to SA in all spatial problems when computing time was limited (Figure 4). Comparisons of CA and SA showed that in a computing time sufficient for CA, SA was not able to produce as good solutions in terms of spatial objectives. The proportion of cut-cut border in three different problems was on the average about 6 % lower with SA. When also old forests were aggregated the proportion of old-old border of SA solutions was on the average 96 % ( CCB and OOB used) and 85 % ( CCB, OOB and Reg used) of CA. The old forest core area of SA was on the average 91 % of CA. The proportion of border between two regeneration cells, which was minimized to disperse regenerative cuts, was about 15 % lower in SA. The net present values of SA and CA solutions were within 2 % in all problems and neither SA nor CA was systematically better than the other. The solution times of CA varied more than in SA.

**Table 6**. Cellular automaton parameter values.

| Parameter | Parallel mode | Sequential mode |
|---|---|---|
| Initial mutation probability | 0.005 | 0 |
| Change parameter for mutation probability | 4 | 0 |
| Initial innovation probability | 0.8 | 1 |
| Change parameter for innovation probability | 1 | 0 |

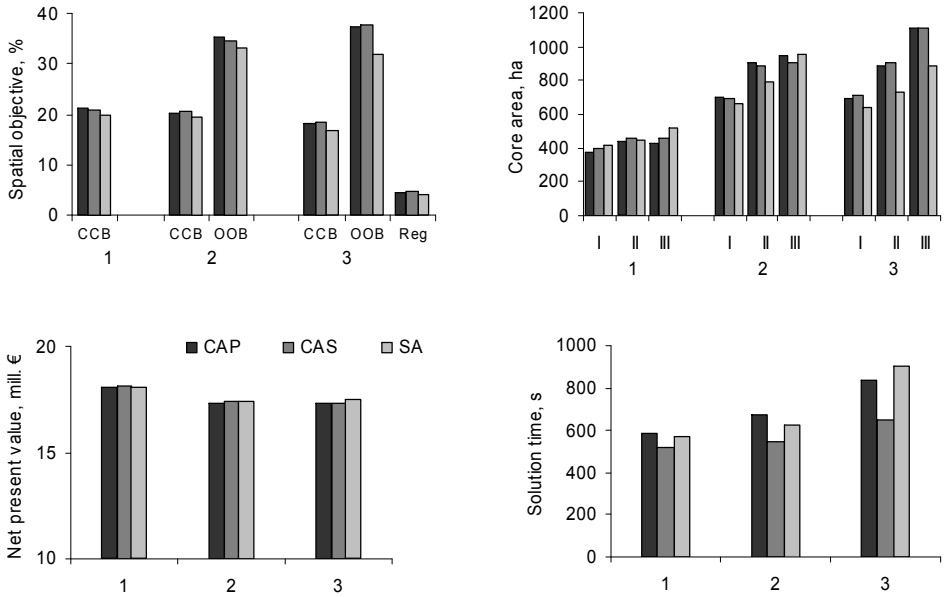


**Figure 4**. Variables describing the cellular automaton (CAP = parallel cell state updating, CAS = sequential updating) and simulated annealing (SA) solutions for spatial problems 1–3 in terms of three spatial objectives (CCB, OOB and Reg). CCB (percentage of border between two adjacent cut cells) and OOB (percentage of border between two adjacent old-forest cells) were maximized and Reg (percentage of border between two adjacent regeneration-cut cells) was minimized. Roman numbers I, II and III refer to a 20-year time period.

There were no major differences in the performance of the parallel and sequential modes of CA. Cellular automata with sequential (CAS) or parallel (CAP) state-updating modes ended up in very similar solutions in spatial problems. CAS reached slightly better net present values in two of the problems but CAP was better in aggregating cuttings when OOB and CCB were used. In the same problem the total old-forest core area was larger with CAP for all time-periods, but CAS was better in the problem where also regeneration cuttings were dispersed. 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 the most 90 % of the solution time of CAP. Despite different updating of cell states the two modes produced rather similar spatial arrangements of cut and old-forest cells. 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. After the first iteration the CAS solution was clearly more aggregated. This is due to the fact that the sequential mode evaluated all the cells during one iteration, and changed cell states immediately. Both modes had little variation in the solution quality between repeated runs. The parallel mode seemed to work slightly better when the complexity of the problem increased.

### 3.5 The use of dual price approach in spatial forest planning (Study V)

Study V compared the spatial application of the reduced costs heuristic (RC) to a cellular automaton (CA) and simulated annealing (SA) in a compartment forest and raster forest. In the compartment forest RC resulted in 2–3% smaller total net present value than the other heuristics. The ending growing stock volume and the periodical harvests were close to the target values for all methods. The running time of RC was 1/3 of the time required by CA and half of the time used by simulated annealing. To summarise, the three heuristic methods were rather similar in terms of economic profitability and volume targets, and their running times were of the same magnitudes.

The solutions for the compartment forest differed more in terms of spatial objectives. The number of cutting areas is smaller and the mean size of cutting area larger in RC than in CA and SA. This means that RC aggregated cuttings better than the other methods. SA was better than CA in creating cutting aggregations. On the other hand, CA was the best in creating old-forest core areas since the number of old-forest core areas was the smallest and their total area and mean size largest in CA. The second best was RC, and SA was the worst. Also the maps of compartment forest indicate that RC and SA created the largest cutting aggregations and CA the largest old forest aggregations.

The total net present values produce by the final solution in raster forest were nearly the same for all the three heuristics. The volume targets were met nearly equally well in all methods. The computing times were close to each other. RC was by far the best in aggregating cuttings (Fig. 5). The total cutting areas were nearly the same in all methods, but RC resulted in a much smaller number and larger mean size of cutting areas than the two other methods. SA was better than CA in cutting aggregation. RC was also the best method in old forest aggregation (Fig. 5), followed by CA.

With RC deterministic initial solution (the first alternative of the stand) was compared to random initial solution. When the initial solution consisted of schedules in which the cutting was conducted at the first possible moment (immediately when cutting was permissible according to current instructions) the solution resembled the final solution already after the first iteration. Especially the cutting areas of the third time-period and the old-forest areas at the end of the third period were almost the same after the first iteration (with zero dual prices) as in the final solution. When the initial solution consisted of random schedules the process of organizing cuttings and old-forest patches was slower. The running time was also longer when the initial solution consisted of random schedules. However, the final solution was fairly similar with both initial solutions.

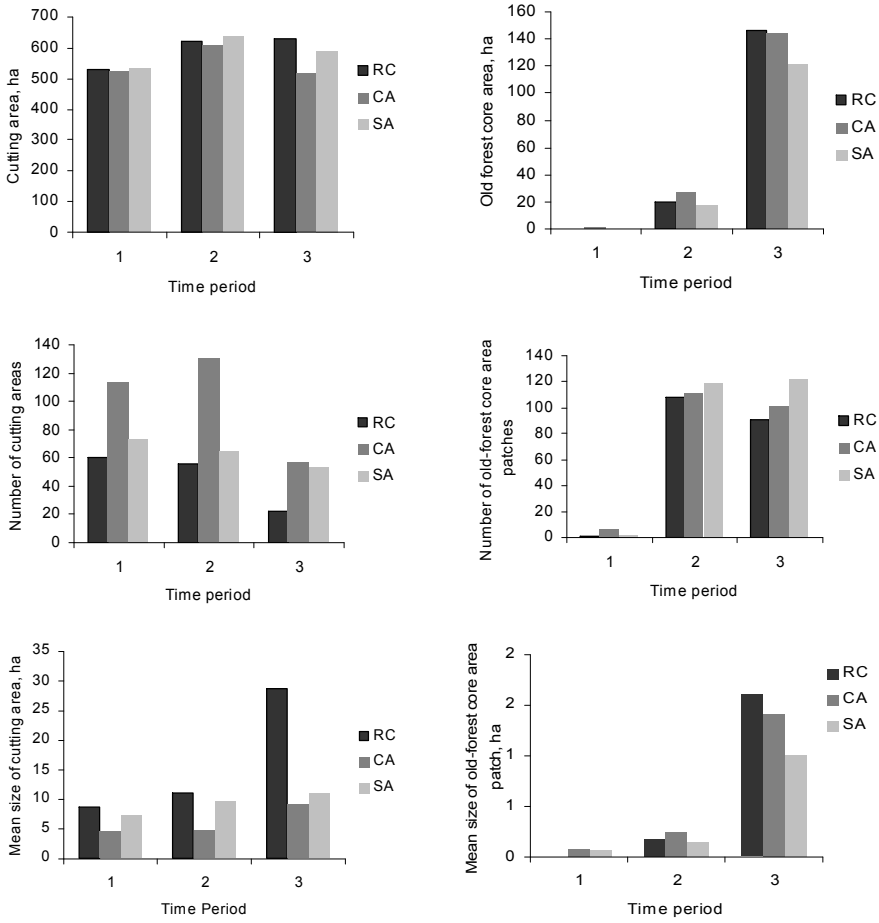


**Figure 5**. Variables describing cutting areas and old forest core area patches in the raster forest.

The maps in Figure 6 reveal the rather clear difference between the decentralised (RC and CA) and centralised (SA) heuristics to aggregate forest features in the raster forest. According to the maps, both cuttings and old forests are much better aggregated in RC and CA than in SA, and both the cutting areas and the old forest patches are slightly more continuous in RC than in CA.

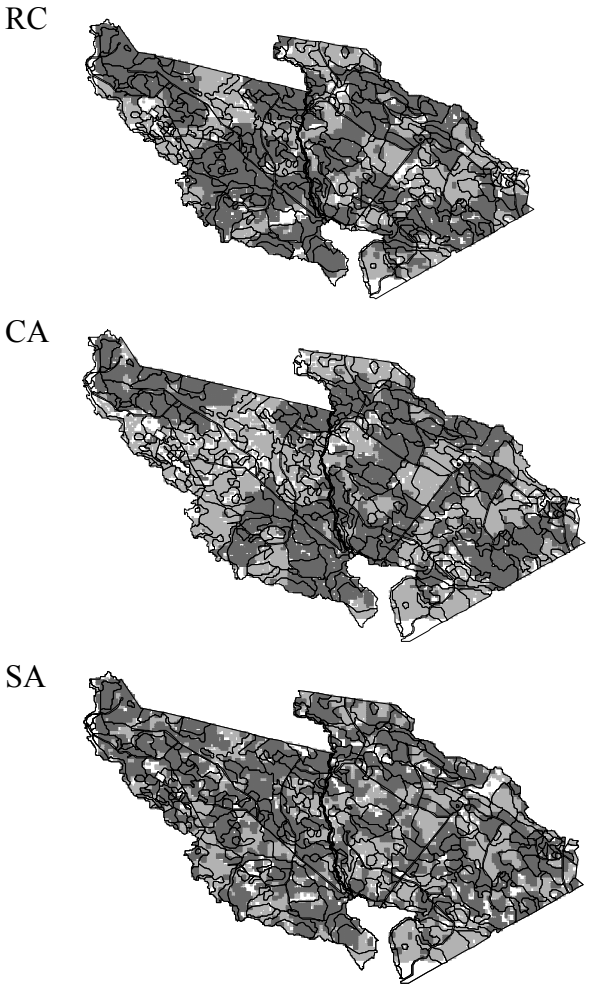


**Figure 6**. Cutting areas (dark grey) during the third time period and old forest patches (light grey) at the end of the third time period in the raster forest.

# 4 DISCUSSION

The problems in this thesis were formulated in a utility theoretic way so that the target levels of objective variables were given trough sub-utility functions. No strict constraints were included in the problems although the use of a utility function as the objective function would not prevent it. In formulations including strict constraints the problem of getting trapped in local optima might be greater than in the utility theoretic formulation in which there is no infeasible region. With problem formulations involving a feasible region

the search may stop once it hits the region boundary although the solution may be far from the global optimum. To alleviate this problem, various means such as strategic oscillation (Reeves 1993) and perturbation (Falcão and Borges 2002) have been proposed. The use of two-compartment moves is an additional possible way of alleviating the problems that the algorithm may have near the border of the feasible region. The results of Bettinger et al. (1999, 2002) in fact show that, at least with tabu search, in problems involving strict constraints the use of two-compartment moves may lead to better solutions than one-compartment moves.

The use of two-compartment moves greatly decreases the differences between heuristics in Study I, indicating that it significantly mitigates the harmful influence of local optima on the search process of the simplest heuristics. Because two-compartment moves seem to be better than one-compartment moves, a logical question is whether three- or four-compartment moves would improve the solutions further. With many simultaneous changes in the solution the method begins to resemble a genetic algorithm in which treatments are changed in several compartments through crossing-over and mutations. Some studies with problems fairly similar to the ones in this study suggest that a genetic algorithm indeed works better with spatial problems than the methods tested in Study I work with a one-compartment neighbourhood (Kurttila and Pukkala 2004, Palahí et al. 2004).

Study II developed a method that can be used to optimize the parameters controlling the search process of a heuristic algorithm. The optimization results were logical and showed that while the optimal values of some parameters were rather constant the others were sensitive to problem type, allowed computing time, or problem size. The objective function value of the forest planning problem was not sensitive to small changes in the parameters of the heuristics. However, because computing time was very sensitive to many parameters, there was not much freedom to set the parameters if both the quality of the solution and speed of the algorithm were maintained.

Because the Hooke and Jeeves method does not require any human intervention the "hybrid method" (non-linear optimization and heuristic search) developed in Study II could be used in forest planning practice. However, although the method is easy to use, the computing times may be very long since the forest planning problem is solved hundreds or thousands of times, making an everyday use of the method impractical. More useful for forest planning practice would be to use the method to find rules for optimal or good parameters of the heuristics. It is logical to think that many parameters of heuristics are related to the number of compartments, or to the size of the neighbourhood (e.g. Park and Kim, 1998). In Study II, most solutions for the parameters of simulated annealing had an initial temperature of magnitude $2/N$, where $N$ is the number of compartments. The freezing temperature was often approximately 1–2 % of the initial temperature, or between $0.01\times1/N$ and $0.02\times1/N$. The number of iterations in the initial temperature was mostly roughly 20 % of the number of compartments. The number of random searches was always less than 20 % of the number of compartments. When longer search times were allowed the optimal initial temperature mostly increased, which is a logical result. The optimal number of initial random searches was less than 100 in all cases.

When the problem size was varied, it was found, that the initial temperature was not related to the number of compartments, but the final temperature decreased as a function of problem size. The cooling multiplier also decreased, which is understandable because the allowed solution time did not increase with the number compartments with the consequence that cooling had to be faster with increasing problem size.

The optimal initial threshold of threshold accepting was not related to problem type and solution time. The optimal threshold multiplier varied surprisingly little, being mostly about 0.95 and in a few cases somewhat higher. The freezing threshold was the smaller the more there were compartments. The threshold multiplier decreased when the problem size increased but the solution time remained the same.

In tabu search, the number of iterations was one to three times the number of compartments, but often it was nearly the same as the number of compartments. It was not strongly and systematically related to the problem type or allowed search time. The number of candidates that were inspected per iteration was about 1 % of the total number of treatment schedules of stands in a quick search, and 4–11 % when more time was available. The optimal tabu tenure for a schedule that is removed from the solution was often about 50 iterations. It was usually optimal to have shorter tabu tenures for entering schedules, which is logical because the whole compartment in which a change is made is tabu for the same time as the entering schedule.

The above examples show that the presented method to optimize heuristic search can be used to derive good guesses or rules of thumb for the parameters, which will ease the practical use of the heuristics. Since the optimal parameters depend on problem size, allowed search time, and type of the problem, a systematic approach might be worthwhile.

In Study III, which compared a compartment forest to a raster forest, the differences between spatial objectives in their ability to aggregate old forests were more clearly shown in the raster forest where optimisation had more freedom to generate landscape patches. The total number of patches was lower and the mean size of the patches larger in the compartment forest. From this viewpoint, the use of predefined compartment boundaries can lead to more complex landscape patch shapes, although this can in part result from the characteristics of the planning area. Of the two landscape metrics that were used to aggregate old forest, the MNM (mean of neighbourhood minima) seems to be slightly more capable than OOB (proportion of old-old border) of generating landscape patches of good shape in terms of AWMSI. The OOB objective is only dependent on the boundary between adjacent old stands and therefore tends to maximise only the area of those calculation units that exceed the OFI threshold value. In the MNM, a calculation unit gets the minimum value of the neighbours. This property of the MNM takes into account the whole range of variation in the OFI values and is not dependent on any threshold value, which also explains the rather clear difference in the total old forest area between the MNM and OOB objectives. The MNM objective also tends to move unfavourable calculation units as far as possible from good calculation units. The OOB objective generates landscapes with more isolated and distinct old forest patches, with a consequence that the landscapes have more edge effect than what results from the use of MNM. It may be concluded that for a good

landscape in terms of old forest habitats it is not enough to maximize the area and aggregation of the patches, but also create a smooth landscape in terms of stand age.

The degree of superiority of the raster approach depends on the set management objectives, composition and structure of the forest, and the way in which the surveyor delineates the compartments. Therefore, overall conclusions about the magnitude of superiority of the raster approach cannot be drawn on the basis of Study III.

Studies III and V demonstrate that the current tools for spatial optimisation are sufficient for creating feasible treatment units from raster cells, implying that prior-planning compartment delineation is not necessary. When treatment units are formed in optimisation they may be regarded as temporary, i.e. they are no longer used after the treatment has been implemented. It should also be remembered that many of the traditional compartment boundaries are also invisible in the forest; they are subjectively drawn on aerial photographs in places in which the change in forest features is only gradual.

There was no variation in the initial forest data between raster cells that were located within the same initial compartment in Study III. Theoretically the raster forest is at least as efficient as the compartment forest. Abandonment of predefined boundaries leads to a less constrained use of resources: the treatments are not forced to follow compartment boundaries. The visualization of the plans showed that the landscape patches, especially cutting areas, frequently deviated from the compartment boundaries. Replacing the predefined and generally over one hectare forest compartments with small raster cells dramatically increases the number of calculation units that are used in the optimisation. This makes optimisation more difficult and leads to longer running times. One way to reduce computing time is the use of decentralized computing methods like cellular automata (CA) and the reduced costs method (RC). However, it should also be noted that the additional computing cost might be negligible compared to the potential benefits of using raster cells.

In Studies IV and V, solutions produced with decentralized methods were very competitive compared to solutions generated with traditional methods. In the non-spatial problems of Study IV 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 clearly better than SA solutions in terms of the spatial layout of desired features. The parameters of SA were optimised for the allowed computing time, which means that the performance of SA would not improve with parameter adjustments if the computing time is not continued.

The biggest advantage of decentralized methods 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 and RC to make calculations locally saves computing time. However, the main reason for the superiority of decentralized methods is greatly reduced solution space as compared to centralised heuristics (Hoganson and Rose 1984). 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 centralised methods. The CA developed in Study IV 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. First local optimisation is performed to reach a good spatial organisation, 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 the time with feasible solutions (e.g. Richards and Gunn 2003, Bettinger & Zhu 2006).

There were no significant differences in the performance of the parallel and sequential modes of the CA in Study IV. The performance of the CA was 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 Study IV 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.

The CA used in Studies IV and V resembles the method of Hoganson and Rose (1984) who used heuristically adjusted the 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 optimisations were repeated after every adjustment, and the optimisations were stopped when the constraining variables reached good enough values. The method Hoganson and Rose (1984) was developed for non-spatial problems. Decentralized computing like CA is the most suitable for solving problems in which the goals are additive, i.e. sums of stand level values. The CA also handle spatial goals in a very natural way. With most spatial goals the best treatment for a single cell or stand depends on the characteristics and treatments of surrounding stands, which corresponds to the evaluation process of CA. 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 maker's preferences, timber prices etc. Therefore, the requirement for optimality can be relaxed and minor violations of constrains accepted (Hoganson and Rose 1984).

In Study V, a spatial version of the reduced cost method (RC) for selecting the optimal treatments for stands was developed. It differs from the original method of Hoganson and Rose (1984) so that the stand-level objective function may depend on the conditions of neighbouring stands. For this reason, the method required an initial solution which must be produced before starting the RC heuristic. The initial solution affects all the subsequent solution. Study V used the schedule that was simulated first for the compartment as the initial solution. This schedule happened to be always the one in which the cutting was conducted in the first possible moment, which means that the initial solution had the maximum possible cut during the 1$^{st}$ time period and the total cut was also very high. This

means that the heuristic algorithm mainly had to replace cutting schedules by no-cutting schedules, and shrink the initially large cutting aggregations. This most probably is an easier task than creating good cutting area aggregations from an initial solution in which cuttings are not aggregated (Heinonen and Pukkala 2004). The initial solutions may explain why the RC method was especially good in aggregating cuttings. If the initial solution has consisted of the no-cutting schedules the results may have been better for old-forest aggregation and less good for cutting-area application.

The computing time of the RC method mainly depends on how many different sets of dual prices need to be used during the optimisation run, and what is the step size in the up-dating of dual prices. If the initial dual prices are near the final ones or the step size is large, the method is very fast. A large step size leads to the situation that the target levels of the constraints are only roughly met, which may not be a problem in most planning situations. To guess the initial dual prices so that they are close to the right ones is difficult for most managers, which means that in a practical application of the RC method the initial dual prices should most probably all be zero. The used of fixed initial dual prices with a fixed step makes the RC method extremely easy for a forest manager to use because in this case there are no parameters at all which should be asked from the manager.

The key point when developing the RC method further is to improve the heuristics which changes the dual prices during the optimisation run. The initial dual prices should be zero or other values that are deduced by the algorithm, and the heuristic should use big steps in the beginning of the process. Then, the fine-tuning of the dual prices near the end of optimisation should be made in a more sophisticated way than in Study V. The decentralized algorithms could be further developed and diversified 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


Aarts, E. & Lenstra, J. K. 1997. Introduction. In: Aarts, E., ja Lenstra, J. K. (eds.) Local search in combinatorial optimization. John Wiley & Sons, New York, pp. 1-16.

Baskent, E. Z. & Jordan, G. A. 1995. Characterising spatial structure of forest landscape. Canadian Journal of Forest Research 25: 1830-1849.

- & Jordan, G. A. 2002. Forest landscape management modelling using simulated annealing. Forest Ecology and Management 165: 29-45.

Bettinger, P., Sessions, J. & Boston, K. 1997. Using tabu search to schedule timber harvests subject to spatial wildlife goals for big game. Ecological Modelling 42: 111-123.

-, Boston, K. & Sessions, J. 1999. Intensifying a heuristic forest harvest scheduling procedure with 2-opt decision choices. Canadian Journal of Forest Research 29: 1784-1792.

-, 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.

- & Zhu, J. 2006. A new heuristic method solving spatially constrained problems based on mitigation of infeasibilities radiating outward from forced choice. Silva Fennica 40(2): 315-333.

Borges, J.G., Hoganson, H.M. & Falcão, A.O. 2002. Heuristics in multi-objective forest planning. In: Pukkala, T. (ed), Multi-objective Forest Planning. Kluwer Academic Publishers, Dordrecht, pp. 119-151.

Boston, K. & Bettinger, P. 2001. The economic impact of green-up constraints in the southeastern United States. Forest Ecology and Management 145: 191-202.

- & Bettinger, P. 2002. Combining tabu search and genetic algorithm heuristic techniques to solve spatial harvest scheduling problems. Forest Science 48: 35-46

Brumelle, S., Granot, D., Halme, M. & Vertinsky, I. 1997. A tabu search algorithm for finding good forest harvest schedules satisfying green-up constraints. European Journal of Operation Research 106: 408-424.

Dowsland, K.A. 1993. Simulated annealing. In: Reeves, R. S. (ed.), Modern heuristic techniques for combinatorial problems. Blackwell Scientific Publications, pp. 20-69.

Dueck, G. & Scheuer, T. 1990. Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. Journal of Computer Physics 90: 161–175.

Falcão, A. O. & Borges, J. 2002. Combining random and systematic search heuristic procedures for solving spatially constrained forest management scheduling models. Forest Science 48(3): 608-621.

Glover, F. 1989. Tabu search – Part I. ORSA Journal of Computers 1: 190-206.

Haara, A. 2003. Comparing simulation methods for modelling the errors of compartment inventory data. Silva Fennica 37(4): 477-491.

Hoganson, H.M. & Borges, J.G. 1998. Using dynamic programming and overlapping subproblems to address adjacency in large harvest scheduling problems. Forest Science 44(2): 526-538.

- & Rose, D.W. 1984. A simulation approach for optimal timber management scheduling. Forest Science 30: 220–238.

Holmgren, P. & Thuresson, T. 1997. Applying objectively estimated and spatially continuous forest parameters in tactical planning to obtain dynamic treatment units. Forest Science 43 (3): 317-326.

Hooke, R. & Jeeves, T. 1961. "Direct search" Solution of Numerical and Statistical Problems. J. Ass. Comp. Mach. 8: 212 – 229.

Hurme, E., Kurttila, M., Mönkkönen, M., Heinonen, T. & Pukkala, T. 2006. Optimizing species persistence and timber harvest: a site-specific spatial habitat model in forest planning calculations. Landscape Ecology. In press.

Kirkpatrick, S., Gelatt Jr., C. D. & Vecchi, M. P. 1983. Optimization by Simulated Annealing. Science 220(4598): 671-680.

Kurttila, M. 2001. The spatial structure of forests in the optimization calculations of forest planning — a landscape ecological perspective. Forest Ecology and Management 142(1-3): 129-142.

-, Pukkala, T. & Loikkanen, J. 2002. The performance of alternative spatial objective types in forest planning calculations: a case for flying squirrel and moose. Forest Ecology and Management 166: 245-260.

Lind, T. 2000. Strategic forestry planning – evaluation of variable spatial aggregations and forest landscapes. Ph.D. Thesis, Acta Universitatis Acriculturae Sueciae Silvestria 149. 28 pp.

Luonnonläheinen metsänhoito, 1994. Metsänhoitosuositukset. Metsäkeskus Tapion julkaisuja 1994. 74s. In Finnish.

Lockwood, C. & Moore, T. 1993. Harvest scheduling with spatial constraints: a simulated annealing approach. Canadian Journal of Forest Research 23: 468-475.

Lu, F. & Eriksson, L. O. 2000. Formation of harvest units with genetic algorithms. Forest Ecology and Management 130: 57-67.

Mathey, A.-H., Krcmar, E. & Vertinsky, I. 2005. Re-evaluating our approach to forest management planning: A complex journey. The Forestry Chronicle 81(3): 359-364.

McGarigal, K. & Marks, B.J. 1995. FRAGSTATS: spatial pattern analysis program for quantifying landscape structure. USDA Forest Service, Pacific Northwest Research Station, Portland, OR. General Technical Report. PNW-351.

Murray A. 1999. Spatial restrictions in harvest scheduling, Forest Science 45(1) 1-8.

- & Church, R. 1995. Measuring the efficacy of adjacency constraint structure in forest Planning, Canadian Journal of Forest Research 25: 1416-1424.

Nalle, D. J., Arthur, J. L. & Sessions, J. 2002. Designing compact and contiguous reserve networks with a hybrid heuristic algorithm. Forest Science 48(1): 59-68.

Næsset, E. 2002. Predicting forest stand characteristics with airborne scanning laser using a practical two-stage procedure and field data. Remote Sensing of Environment 80: 88-99.

- , Gobakken, T., Holmgren, J., Hyyppä, H., Hyyppä, J., Maltamo, M., Nilsson, M., Olsson, H., Persson, Å. & Söderman, U. 2004. Laser Scanning of Forest Resources: The Nordic Experience. Scandinavian Journal of Forest Research 19: 482- 499.

Palahí, M., Pukkala, T., Pascual, L. & Trasobares, A. 2004. Examining alternative landscape metrics in ecological forest landscape planning: a case for capercaillie in Catalonia. Investigaciones Agrarias: Sist. Recur. For. 13 (3), 527-538.

Park, M.-W. & Kim, Y.-D. 1998. A systematic procedure for setting parameters in simulated annealing algorithms. Computers Ops. Res. **25**: 207-217.

Pukkala, T. 2002. Introduction to multi-objective forest planning. In: Pukkala, T (ed.), Multi-Objective Forest Planning. Kluwer Academic Publishers, Netherlands, pp.1-19.

-, 2004. Monikäytön suunnitteluohjelma Monsu. Ohjelmiston toiminta ja käyttö. University of Joensuu Faculty of Forestry. 72 pp. In Finnish.

- & Kangas, J. 1993. A heuristic optimization method for forest planning and decision making. Scandinavian Journal of Forest Research 8: 560–570.

- & Kurttila, M. 2005. Examining the performance of six heuristic search techniques in different forest planning problems. Silva Fennica 39(1): 67-80.

Reeves, C. R. & Beasley, J. E. 1993. Introduction. In: Reeves, C. R. (ed.). Modern heuristic techniques for combinatorial problems. Blackwell Scientific Publications, New York, pp. 1-19.

Richards, E.W. & Gunn, E.A. 2003. Tabu search design for difficult forest management optimization problems. Canadian Journal of Forest Research 33(6): 1126-1133.

Strange, N., Meilby, H. & Thorsen, J. T. 2002. Optimizing land use in afforestation areas using evolutionary self-organization. Forest Science 48(3): 543-555.

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

Tokola, T., Pitkänen, J., Partinen, S., & Muinonen, E. 1996. Point accuracy of a non-parametric method in estimation of forest characteristics with different satellite materials. International Journal of Remote Sensing 17(12): 2333–2351.

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

Tuominen, S. & Haakana, M. 2005. Landsat TM imagery and high altitude aerial photographs in estimation of forest characteristics. Silva Fennica 39(4): 573–584.

Von Neumann, J. 1966. Theory of self-reproduction automata. Burks, A. ed. University of Illinois Press. 77 pp.

Ward, D. P., Murray, A. T. & Phinn, S. R. 2003. Integrating spatial optimization and cellular automata for evaluating urban change. Annals of Regional Sciences 37: 131-148.

Weintraub, A., Church, R. L., Murray, A. T. & Guignard, M. 2000. Forest management models and combinatorial algorithms: analysis of state of the art. Annals of Operations Research 96: 271-285.

Öhman, K. 2000. Creating continuous areas of old forest in long-term forest planning. Canadian Journal of Forest Research 30: 1817-1823.

-, 2001. Forest planning with consideration to spatial relationships. Doctoral diss. Department.of Forest Resource Management and Geomatics, SLU. Acta Universitatis agriculturae Sueciae, Silvestria vol. 198. 32 pp.

- & Eriksson, L. O. 2002. Allowing for spatial consideration in long-term forest planning by linking linear programming with simulated annealing. Forest Ecology and Management 161: 221-230.

- & Lämås, T. 2003. Clustering of harvest activities in multi-objective long-term forest planning. Forest Ecology and Management 176: 161-171.