

Application of Ant Colony Optimization for the Risk Management of Wind Damage in Forest Planning

Hongcheng Zeng, Timo Pukkala, Heli Peltola and Seppo Kellomäki

Zeng, H., Pukkala, T., Peltola, H. & Kellomäki, S. 2007. Application of ant colony optimization for the risk management of wind damage in forest planning. *Silva Fennica* 41(2): 315–332.

Ant colony optimization (ACO) is still quite a new technique and seldom used in the field of forest planning compared to other heuristics such as simulated annealing and genetic algorithms. This work was aimed at evaluating the suitability of ACO for optimizing the clear-cut patterns of a forest landscape when aiming at simultaneously minimizing the risk of wind damage and maintaining sustainable and even flow of periodical harvests. For this purpose, the ACO was first revised and the algorithm was coded using the Visual Basic Application of the ArcGIS software. Thereafter, the performance of the modified ACO was demonstrated in a forest located in central Finland using a 30-year planning period. Its performance was compared to simulated annealing and a genetic algorithm. The revised ACO performed logically since the objective function value was improving and the algorithm was converging during the optimization process. The solutions maintained a quite even periodical harvesting timber while minimizing the risk of wind damage. Implementing the solution would result in smooth landscape in terms of stand height after the 30-year planning period. The algorithm is quite sensitive to the parameters controlling pheromone updating and schedule selecting. It is comparable in solution quality to simulated annealing and genetic algorithms.

Keywords genetic algorithms, heuristics, simulated annealing, spatial optimization, harvest scheduling

Authors' address University of Joensuu, Faculty of Forest Sciences, P. O. Box 111, FI-80101 Joensuu, Finland **E-mail** hongcheng.zeng@joensuu.fi

Received 27 September 2006 **Revised** 9 January 2007 **Accepted** 14 February 2007

Available at <http://www.metla.fi/silvafennica/full/sf41/sf412315.pdf>

1 Introduction

Forest management objectives are nowadays very diverse and complex and they include also spatial goals, such as adjacency constraints in harvesting (Crowe and Nelson 2003) and spatial patterns of habitats for wildlife populations (Bettinger et al. 2002). Risk management of wind damage is also a kind of spatial problem. This is because the orographic terrain and surface roughness affect the wind profile and consequently the risk of wind damage. The surface roughness is significantly affected by silvicultural measures. For example, forest gaps created by clear-fellings can speed up the high-speed winds and increase their frequencies (Venäläinen et al. 2004). Trees located at newly created forest edges have the highest risk of wind damage. Old stands are the most vulnerable to be damaged in these circumstances (Zeng et al. 2004). Thus, the risk of wind damage of a forest area depends on the spatial patterns of stand characteristics such as stand height and age. Fragmented forests may induce more risks because of longer stand edges at risk (Zeng et al. 2007). On the other hand, the landscape configuration is changing along with forest growth and dynamics as well as management regimes. For example, aggregated clear-cuts can smooth the landscape regarding stand height and reduce spatial fragmentation (Zeng et al. 2007).

Forest planning problems with spatial objectives are often formulated as non-linear integer programming problems, which cannot always be solved by traditional mathematical programming, e.g. linear programming (Boston and Bettinger 2002). Heuristic optimization techniques, which can solve combinatorial problems, are therefore increasingly being used in forest planning (Borges et al. 2002). Examples of these heuristics include simulated annealing, SA (e.g. Dahlin and Sallnäs 1993, Lockwood and Moore 1993, Öhman 2000), tabu search, TS (e.g. Bettinger et al. 1997, Boston and Bettinger 1999) and genetic algorithms, GA (e.g. Lu and Eriksson 2000, Falcao and Borges 2001). Recently, different variations and combinations of these techniques have also been applied in forestry (Bettinger et al. 1999, Boston and Bettinger 2002, Falcao and Borges 2002, Heinonen and Pukkala 2004).

In addition to those heuristics mentioned above,

ant colony optimization (ACO) proposed by Colorni et al. (1991) has also been widely used in decision making recently (Bonabeau et al. 2000). It mimics the behaviour of ants in their search for the shortest path from their colony to food source: ants lay pheromone (chemical) trails when they move. The pheromone evaporates at certain rate with time. The more there is pheromone in a path, the more likely it will be followed by other ants. The ACO was first and most commonly used for Traveling Salesman Problem (TSP), which consists of finding the shortest tour between a number of cities when visiting each only once and ending at the starting point (e.g. Dorigo et al. 1996, Merkle et al. 2000, Sun et al. 2004). The ACO has also previously been applied in some other combinatorial optimization problems, such as machine scheduling (e.g. Bauer et al. 1999) and quadratic assignment problems (Maniezzo and Colorni 1999).

In regard to the risk management of wind damage in forest planning, Zeng et al. (2007) were among the first ones who integrated heuristics (SA, TS and GA) with a forest growth and yield model, mechanistic wind damage model, and geographical information system (GIS). They optimized the clear-cut patterns of a forest under multiple objectives concerning timber harvest and the risk management of wind damage.

Ant colony optimization is still quite a new technique and seldom if ever used in the field of forest planning. This work was aimed at evaluating the suitability of ant colony optimization for optimizing forest clear-cut patterns to minimize the expected wind damage under the constraint of sustainable and even flow of periodical (10-year) timber harvest. For this purpose, the ACO algorithm was revised and coded using the Visual Basic Application (VBA) of the GIS software ArcGIS (ESRI 2006). The performance of the ant colony optimization (ACO) was demonstrated in a forest located in central Finland using a 30-year planning period. It was compared to the performance of other heuristics, namely simulated annealing and genetic algorithms.

2 Material and Methods

2.1 Study Area and Simulation of Treatment Alternatives

The forest area employed in this study represented a typical boreal forest in central Finland (63°01'N, 27°48'E). It was mostly dominated by Scots pine (*Pinus sylvestris*) and Norway spruce (*Picea abies*) stands, but some silver birch (*Betula pendula*) stands were also present. The study area was surveyed in 2001 and included 225 ha of forests (excluding clear-cut areas) and 4 ha of open terrain (including fields and clear-cut areas). The 225 ha of forest consisted of 63 ha of Scots pine (36 stands), 158 ha of Norway spruce (75 stands) and 4.5 ha of silver birch (4 stands). The GIS software ArcGIS was used to store forest stand data and their boundaries. A forest growth and yield model SIMA (Kellomäki et al. 1992), mechanistic wind damage model HWIND (Peltola et al. 1999) and ACO were all embedded in the ArcGIS. The SIMA model (Kellomäki et al. 1992) is a non-spatial gap model which simulates forest growth and yield in individual stands based on mean stand characteristics as regulated by environmental conditions and management. The HWIND model (Peltola et al. 1999) describes the mechanistic behavior of Scots pine, Norway spruce and birch trees under wind loading and predicts the minimum wind speeds lasting 10 min at 10 m above ground level at which trees will be uprooted or broken. This, so-called critical wind speed, is the highest speed that a tree can resist. Trees with smaller critical wind speeds have higher risks of wind damage.

The SIMA model was used to simulate the growth and dynamics of the forest stands over a 30-year period with different treatment schedules. During the 30-year planning period, there may be no clear-cut, or a clear-cut at the 5th, 15th or 25th year of simulation. Therefore, each stand may have at maximum 4 alternative treatment schedules. Stands that did not meet the clear-cut criteria at the 5th, 15th or 25th year had less than 4 alternative schedules. The criteria, which specified whether a clear-cut was allowed or not, were the mean tree diameter at breast height (DBH) and stand age. The DBH criterion was set at 27 cm for Scots pine and 24 cm for Norway spruce and

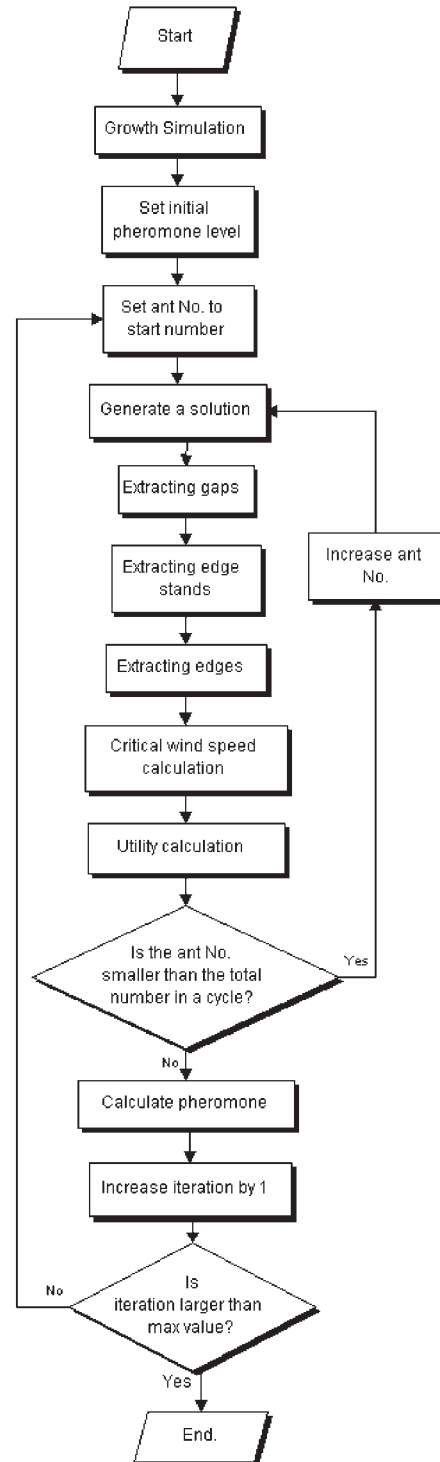


Fig. 1. The flow of alternative solution generation.

silver birch. The age criterion was 70 for Scots pine, 80 for Norway spruce, and 60 years for silver birch. After clear-cut, artificial regeneration was applied so that the stand was planted with seedlings of the same tree species which were previously growing there. No other management practices were applied during the simulation.

A candidate solution was a combination of treatments schedules for all the stands. It represented a possible forest plan. The 10-year timber harvests and the risk of wind damage were evaluated for each candidate solution during the optimization process. Since usually only the trees located at forest edges are uprooted or broken in Finnish conditions (Talkkari et al. 2000, Pellikka and Järvenpää 2003), only stands next to gaps (edge stands) were considered to be at risk in this study. Forest gaps include open fields, clear-cut stands and seedling stands with average tree height less than a certain threshold (2 m used in this study). The edge stands and the boundaries between edge stands and gaps (edges) were extracted as polygon and polyline maps, respectively (Fig. 1). Because of forest treatment and the dynamics of stands, the edge stands and edges changed with time and were therefore updated to correspond to different time points. In Finnish conditions, stands with critical wind speeds $< 20 \text{ m s}^{-1}$ can be considered to be at risk (Zeng et al. 2006). For each edge stand, the risk of wind damage (critical wind speed) was calculated by the HWIND model at the 5th, 15th and 25th simulation year.

2.2 Problem Formulation

Each solution was evaluated for the timber harvests and the risk of wind damage using an additive utility function (Eq. 1):

$$U(s) = W_h U_h(s) + W_r U_r(s) \tag{1}$$

where $U(s)$ is the total utility of solution s , $U_h(s)$ and $U_r(s)$ are the sub-utilities from timber harvesting and the risk of wind damage for solution s , respectively; W_h and W_r are the weights of timber harvesting objective and the risk of wind damage objective, respectively. Both weights were set a value of 0.5 in this study.

The sub-utility of timber harvesting was the

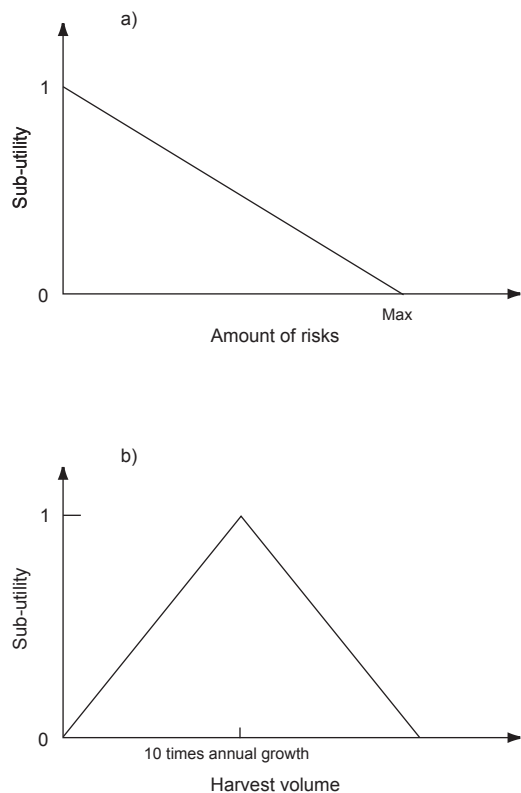


Fig. 2. The sub-utility functions for the risk variables and harvest volume.

sum of the harvest utilities of the three 10-year sub-periods. The utility value of each 10-year period increased with harvested volume until it was 10 times of the annual growth (aspired level), after which the utility started to decrease (Fig. 2b). The sub-utility for the risk of wind damage was calculated by the relative values of the number of stands at risk, their total area and the length of edges at risk (i.e. stands with critical wind speeds $< 20 \text{ m s}^{-1}$) (Eq. 2):

$$U_r(s) = W_d \sum_{T=1}^3 U_{d,T}(s) + W_a \sum_{T=1}^3 U_{a,T}(s) + W_e \sum_{T=1}^3 U_{e,T}(s) \tag{2}$$

where W_d , W_a and W_e are the weights of the three risk variables: the number of stands at risk, their

area and the length of edges at risk. Index T represents a 10-year sub-period. The sub-utility functions of the number of stands at risk, their area and the length of edges at risk had a descending shape, i.e. sub-utilities decreased with the increment of the three variables (Fig. 2a). The weights W_d , W_a and W_e were set to 0.111, 0.222 and 0.667, respectively, because the length of edges at risk was considered the most important variable to assess the risk of wind damage in Finnish conditions (Zeng et al. 2006). The purpose of the optimization was to maximize the total utility, which means minimizing the risk of wind damage and trying to keep an even flow of periodical timber harvest (10-year harvests).

$$P_{xy}^k(t) = \begin{cases} \frac{[\tau_{xy}(t)]^\alpha \cdot [\eta_{xy}]^\beta}{\sum_{l \in allowed_k} [\tau_{xl}(t)]^\alpha \cdot [\eta_{xl}]^\beta} & \text{If } y \in allowed_k \\ 0 & \text{Otherwise} \end{cases} \quad (3)$$

where $P_{xy}^k(t)$ is the probability for ant k to move from city x to city y , l is the index of the cities which allowed for ant k to visit ($allowed_k$), t represents the moving time of the ant from one city to another and also the iterations, $\tau_{xy}(t)$ is the intensity of pheromone trail on path (x, y) at time t , i.e. the pheromone laid by previous ants who had chosen path (x, y) , η_{xy} is called visibility and is the inverse value of the distance between cities x and y , α and β are two positive parameters that control the relative importance of pheromone versus visibility.

The pheromone intensity $\tau_{xy}(t)$ is updated when the ant cycle is ended, i.e. all the ants complete their tours. This is because the pheromone evaporates at a certain rate with time, and new pheromone is added if new ants select the path (x, y) . The optimization process is iterated for a number of ant cycles. It stops when all the ants in the same cycle select the same tour of all the cities. The terminating conditions can also be set in some other way, e.g. on the basis of the number of iterations, or the aspired values of objective variables.

In forest planning problems there are no real paths for ants to select. The treatment schedules of the same stand, however, can be treated as different "path" options for ants to select. This kind

2.3 Application of the Ant Colony Optimization (ACO)

The original algorithm of ACO was specially designed for the traveling salesman problem (TSP) (Dorigo et al. 1996, Dorigo and Gambardella 1997). In this optimization technique, each alternative solution is named as an ant. A number of ants compose an ant cycle. The ants are usually located at different cities (e.g. one ant for each city). All the ants move at the same time. An ant located at city x goes to city y among the unvisited cities according to a probability (Eq. 3) (Dorigo et al. 1996):

of path has no distance, i.e. the visibility values are not used in our revised ant colony optimization algorithm. Thus, an ant selects a schedule of a stand according to the probability (Eq. 4):

$$P_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha}{\sum_{j=1}^{J_i} [\tau_{ij}(t)]^\alpha} \quad (4)$$

where $P_{ij}(t)$ is the probability to select schedule j of stand i at time t (i.e. ant cycle and iteration), $\tau_{ij}(t)$ is the pheromone of schedule j of stand i at ant cycle t , α is a parameter controlling the intensity of the pheromone on the probability of selecting the schedule, J_i is the total number of schedules of stand i . The pheromone $\tau_{ij}(t)$ is updated at each cycle t (Eq. 5):

$$\tau_{ij}(t) = \rho \cdot \tau_{ij}(t-1) + \Delta\tau_{ij}(t) \quad (5)$$

where ρ is a coefficient such that $1-\rho$ is the evaporation rate of the pheromone, $\tau_{ij}(t-1)$ is the previous pheromone intensity of the schedule j . At the beginning of optimization the initial pheromone $\tau_{ij}(0)$ is usually set to a constant value (number of ants per cycle/3 in this study). $\Delta\tau_{ij}(t)$

is the new pheromone trail at time t added by the ants who select schedule j of stand i . It is calculated by Eqs. 6 and 7:

$$\Delta\tau_{ij}(t) = \frac{J_i}{n} \sum_{k=1}^n \Delta\tau_{ijk}(t) \tag{6}$$

$$\Delta\tau_{ijk}(t) = \begin{cases} Q \cdot [U_k(t)]^\gamma & \text{If ant } k \text{ selects schedule } j \text{ of stand } i \\ 0 & \text{Otherwise} \end{cases} \tag{7}$$

where $\Delta\tau_{ijk}(t)$ is the pheromone added by ant k to schedule j of stand i during ant cycle t , n is the number of ants per cycle, $U_k(t)$ is the utility value of ant k of cycle t calculated by Eq. 1 ($U_k(t)$ is corresponding to $U(s)$ in Eq. 1), γ is a positive parameter that controls the effect of utility on the amount of added pheromone (i.e. a power function of $U_k(t)$), Q is a constant to prevent the cumulative pheromone level from being too high or too small. Different values of Q were tested in this study and the final pheromone values of all the schedules in each stand were output and studied. Q with a value of 32 gave reasonable pheromone intensities after a number of ant cycles, and was therefore used in the optimizations of this study.

The original algorithm of ACO will most probably get trapped in local optima (Sun et al. 2004) because the pheromone contents of some paths (schedules in this study) may be too low to be selected by any ants within a small number of iterations. In order to guarantee that each schedule in a stand can get enough pheromone at cycle t , a lower bound of the pheromone in each stand was set in our revised algorithm (Eq. 8):

$$\tau_{iMin}(t) = \frac{\tau_{iMax}(t)}{J_i \left(\frac{t}{g} + 1 \right)} \tag{8}$$

where $\tau_{iMax}(t)$ is the maximum pheromone among the schedules of stand i at cycle t , g is a parameter that controls intensity of decreasing the lower bound values. The lower bound will decrease with the number of cycles so that the algorithm can converge at the final stage.

An ant completes its tour by selecting one schedule per stand (i.e. to compose a forest plan). The pheromone of each schedule in each stand

is updated after all the ants in a cycle complete their tours. Before a new ant cycle is created, the utilities of the forest plans are calculated and the pheromone levels of the schedules are updated according to the utilities of the plans. Therefore, an iteration of the algorithm is equal to developing a set of forest plans (ants), evaluating the plans, and updating the pheromone levels. The ant with the best utility value is always maintained in the memory. The final solution of the algorithm (Appendix 1) is the plan found during the whole run.

2.4 Performance of the ACO for the Risk Management of Wind Damage in Forest Planning

The performance of the ACO was tested using different values of the parameters of the algorithm (i.e. α , γ , g and n , see Table 1). However, coefficient ρ controlling pheromone evaporating was fixed at 0.5 in the sensitivity analysis according to suggestions in earlier studies (Gorigo et al. 1996, Bonabeau et al. 2000). Moreover, since the calculation within the ArcGIS is very time consuming, the algorithm was moved outside ArcGIS for the sensitivity analysis. All the other analyses were done in ArcGIS and they used the “optimal” parameter values found in the sensitivity analyses. In the sensitivity analyses conducted outside ArcGIS, the input dataset and the problem formulation were the same as in ArcGIS, but the spatial calculation was simplified because GIS was not used. The gap size was fixed at 10 times the mean tree height of the stand. In addition, only 3000 solutions (forest plans) were evaluated in one optimization run. This number of solutions was also used in ArcGIS as a stopping criterion. This is because the optimal parameter values depend on the allowed converging time, and converging time mostly depends on the number of candidate solutions. The optimization was repeated 6 times with each parameter combination in sensitivity analyses. The parameter value, which gave the largest average value of the best ants in the 6 repetitions, was chosen as the “optimal” parameter value.

The output results were evaluated to confirm if the ACO would achieve the defined objectives,

Table 1. Sensitivity analysis of ACO parameters.

Parameter	Analyzed values
u	6, 10, 14, 16, 18, 22
Ant numbers of a cycle	20, 30, 40, 50
α	0.8, 1.0, 1.1, 1.2
g	0.25, 0.5, 0.8, 1.0, 1.2, 1.5, 4

i.e. minimizing the risk of wind damage and maintaining even-flow of periodical harvests. The forest landscape at the end of the planning period was described by a fragmentation metric called Contrast-Weighted Edge Density (CWED; see FragStats 2006). The CWED metric measured the fragmentation based on the contrast of the neighbor stands in terms of the tree height.

The performance of ACO was compared with two other heuristic optimization techniques, namely simulated annealing (SA) and genetic algorithms (GA). SA and GA were used for the same planning problem with the same formulation (utility function). Since the extraction of forest gaps and edges, and calculation of gap sizes, dominated the computing time of the whole optimization process, the heuristic algorithm had little influence on the total calculating time. The same number of alternative solutions (forest plans) as in ACO (i.e. 3000) was tested in SA and GA so that they were running for almost the same time.

SA is a local improvement method. A change of the solution is called a move. Solutions that can be obtained with one move from the neighborhood of the current solution. In this work, a move was equal to making two simultaneous changes in the current solution, i.e. the treatment schedule was changed simultaneously in two stands, because the two-stand neighborhood has been found to be a better approach in spatial problems than the more commonly used one-stand neighborhood (Heinonen and Pukkala 2004). Moves that improve the objective function value are maintained. Non-improving moves are maintained 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 is gradually decreased so that at the end of the search the likelihood of accepting inferior moves is close to zero. In this study, the search was stopped when a certain number solutions had been evaluated. Different values of the parameters of simulated annealing were studied using trial and error, and the following values were used with a total number of 3000 solution evaluations: The initial and freezing (stop) temperature were 0.72135 and 0.00334, respectively. The cooling multiplier was calculated from the temperatures so that a fixed number of iterations (3000) were performed.

Unlike SA, the search process of GA is not based on neighborhood search. Instead, GA are based on an initial population of solution alternatives, their evaluation and their breeding. Each iteration (named as a generation) has a set of alternative solutions called parent chromosomes. Two of these parent chromosomes were selected to generate a new chromosome for the next generation. One chromosome was selected with a probability proportion to its ranking; the other was chosen randomly with an equal probability for all the remaining chromosomes. The two selected parent chromosomes were processed by crossing over (combining parts of two chromosomes) and mutation (random change in one or several genes, or stands) giving birth to a new chromosome (offspring). The incremental GA technique was used in this study, i.e. the new chromosome replaced one chromosome of the current population. The removed chromosome was selected based on its objective function value, the probability of removal being highest for chromosomes that had a low utility function value. The parameters of the GA were optimized as follows: population size of each generation was set as 40 solutions. The probabilities of mutation at first iteration (0.5) and at last iteration (2.0) were set according to similar study of Zeng et al. (2007), and other studies on forest planning by Heinonen and Pukkala (2004) and Pukkala and Kurttila (2005). The probability of a mutation for intermediate iterations was obtained with linear interpolation. A probability of 0.5 means that one random stand (gene) is mutated with a probability of 0.5. A probability of 2.0 means two random stands are mutated with a probability of 1.0.

3 Results

3.1 Effects of ACO Parameters

When studying the sensitivity of the ant colony optimization (ACO) to its parameters, it was found that small changes in γ had a very small effect on the performance of the algorithm. For example, the performance of $\gamma=10$ was difficult to distinguish from the performance of $\gamma=6$ and 14. The same occurred when comparing cases $\gamma=16$ or $\gamma=18$ to $\gamma=14$ or $\gamma=22$. This is because the utilities of most solutions (ants) were within a small range (0.7–1). Therefore, parameter γ was changed on a larger scale in the sensitivity analysis so that there

was more difference in the pheromone between schedules within the same stand. The higher was the value of γ , the more important were the ants with high utilities in reinforcing new pheromone. The algorithm converged faster with high γ values than with low γ values (Fig. 3), i.e. the mean utility of an ant cycle increased fastest and its standard deviation decreased fastest when γ was set to 22. Although the algorithm with parameter of $\gamma=22$ had the highest mean utility at the final stage, it did not output the best utility value. The algorithm gave the best utility (0.9771) with $\gamma=18$ when the other parameters were fixed as: $\alpha=1$, $g=1$, and 30 ants per cycle.

The influence of parameter γ on the pheromone

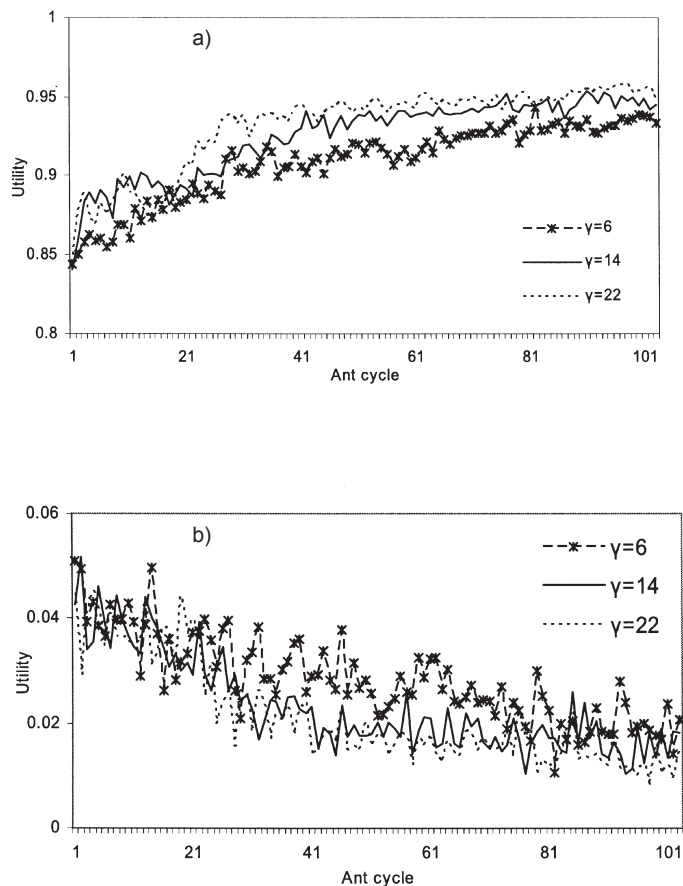


Fig. 3. Sensitivity analysis for parameter γ : mean utility (a) and its standard deviation (b). Other parameters were fixed: $\alpha=1$, $g=1$. Each cycle had 30 ants.

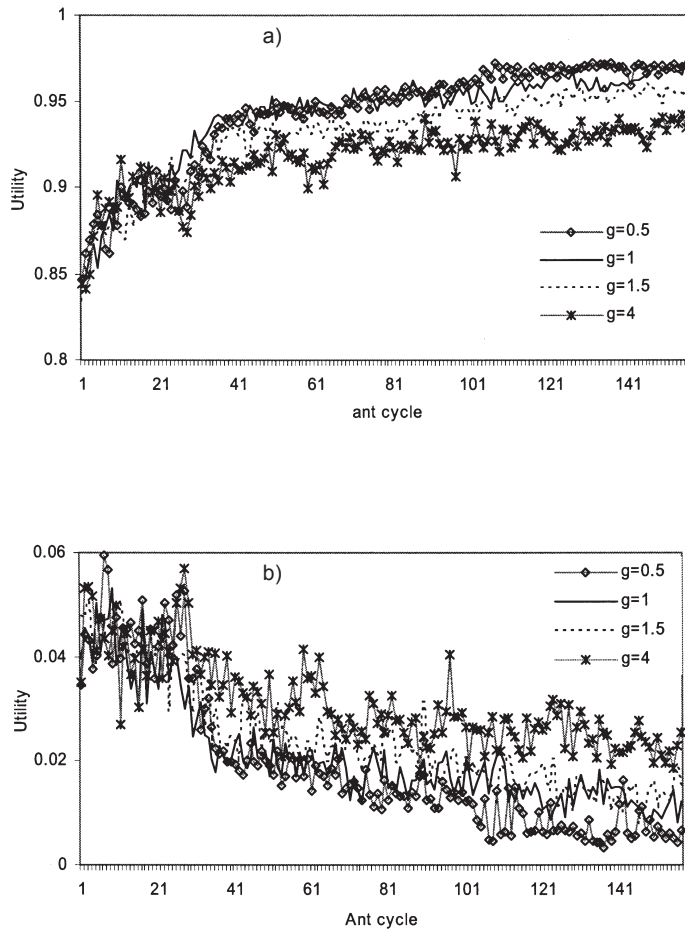


Fig. 4. Sensitivity analysis for the lower bound of pheromone (g): mean utility (a) and its standard deviation (b). Other parameters were fixed: $\gamma = 14$, $\alpha = 1$, 20 ants per cycle.

depends on the lower bound g . The algorithm with a larger value of g (e.g. 4) had a lower mean utility and higher standard deviation in most ant cycles (Fig. 4). This is because a large value of g (e.g. 4) limits the difference between the maximum and minimum pheromone (Eq. 8). As a result, the algorithm converged more slowly. When parameter g was decreased to a very small value, for example 0.25, the result differed little from the case of $g=0.5$. It seems that 0.25 was a too small value to constrain the lower bound pheromone, and therefore it had little influence on the algorithm performance.

When studying effect of the number of ants per cycle on the convergence of the algorithm, the same total number of candidate solutions (3000) was still used. Thus the number of ant cycles differed when using different number of ants per cycle. There were over 150 cycles when each cycle had only 20 ants, and about 60 cycles with 50 ants per cycle. The algorithm gave a much higher mean utility when a cycle had 20 ants compared with 30 or 50 ants in each cycle (Fig. 5). Even within the same number of ant cycles, the mean utility was larger when there were only 20 ants in a cycle, while it was smaller when

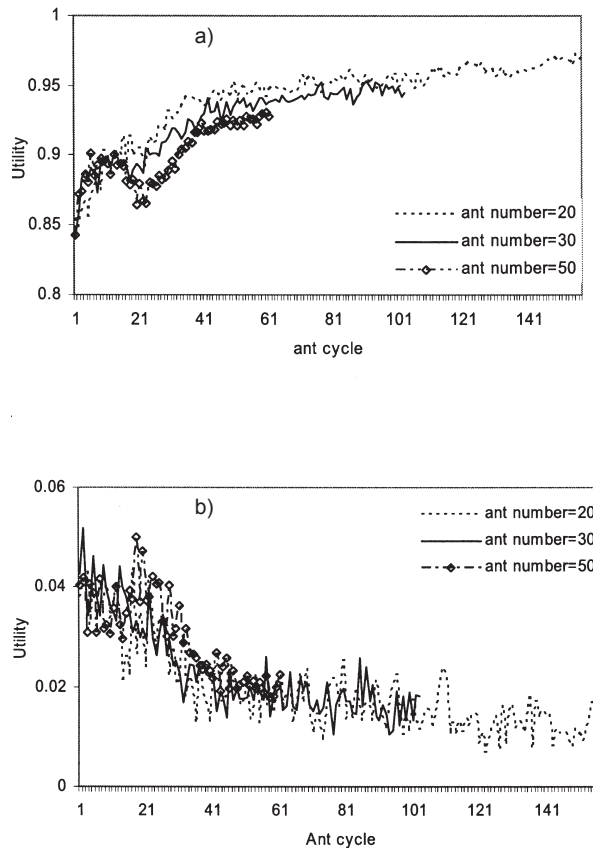


Fig. 5. Sensitivity analysis for the number of ants in a cycle: mean utility (a) and its standard deviation (b). Other parameters were fixed: $\gamma=14$, $\alpha=1$, $g=1$.

the cycle had 50 ants (Fig. 5). Therefore, many more iterations were needed for the algorithm to converge when each cycle had 30 or 50 ants. As 20 ants per cycle is a small number already, smaller number of ants per cycle was not tested in this study.

Since both parameter γ and the number of ants per cycle can affect the convergence of the algorithm, the two parameters were combined and their joint influence on the algorithm performance was studied (Table 2). It was found that different combined parameter values result in a rather similar utility. For example, when the cycle had 20 ants with parameter $\gamma=14$, the algorithm gave a similar utility as in the case when a cycle had 30 ants with $\gamma=18$. Therefore if parameter γ

Table 2. The best utility outputted by the ant colony optimization using different combined values of parameter γ and the number of ants per cycle.

	Number of ants per cycle			
	20	30	40	50
$\gamma=6$	0.975	0.973	0.971	0.969
$\gamma=10$	0.975	0.974	0.971	0.971
$\gamma=14$	0.9773	0.974	0.972	0.972
$\gamma=16$	0.9767	0.975	0.975	0.972
$\gamma=18$	0.975	0.9771	0.975	0.974
$\gamma=22$	0.976	0.974	0.976	0.974

Note: The values in bold were outputted maximum utilities.

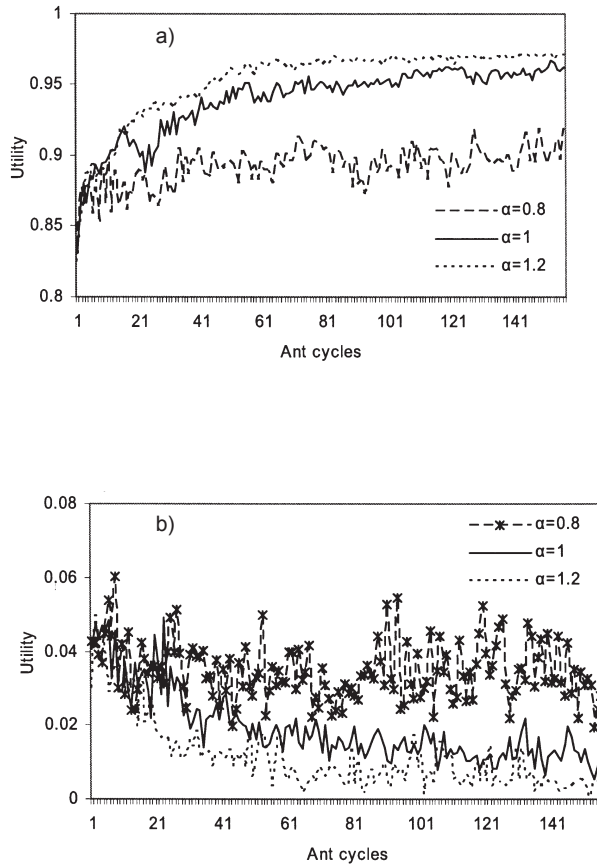


Fig. 6. Sensitivity analysis for parameter α : mean utility (a) and its standard deviation (b). Other parameters were fixed: $\gamma=14$, $g=1$. Each cycle had 20 ants.

was fixed to 14, the optimal number of ants per cycle was 20; while it was 30 when parameter γ was fixed to 18. The combination $\gamma=14$ and 20 ants per cycle gave the largest utility (Table 2). Thus, the joint influence of different parameters needs to be studied in order to search the optimal parameter values of the algorithm.

Compared to parameter γ and the number of ants per cycle, a smaller change of parameter α , defining the selecting probability, could significantly affect the performance of ant colony optimization. The mean utility increased significantly and its standard deviation decreased significantly when α was increased by 20% (i.e. 1.2), and vice versa (Fig. 6). The algorithm seemed to converge much earlier during the optimization

process when α was 1.2, i.e. the standard deviation of utility within a cycle was around 0.01 after 60 cycles (Fig. 6b). Due to smaller variation, it gave a smaller best utility than $\alpha=1$ even though $\alpha=1.2$ had a larger mean utility. The effect of parameter α is so significant because it changes the probability of selecting schedules directly and no lower bound affects the selecting probability between schedules within the same stand.

3.2 Performance of ACO in the Forest Planning Problem

According to the results of all the sensitivity analyses the values of the parameters were set

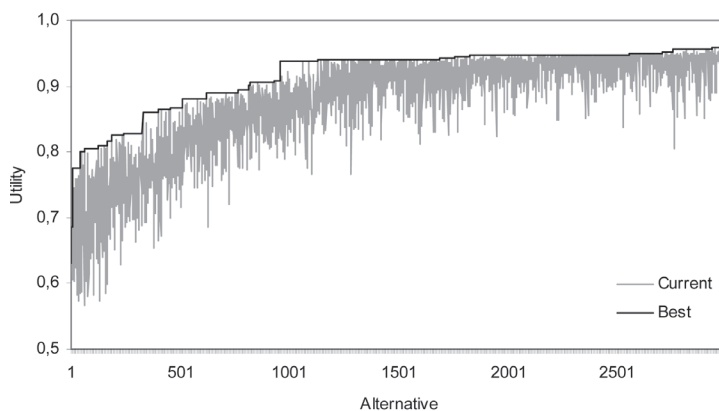


Fig. 7. The current and best utilities output by ACO.

as follows: $\gamma=14$, $g=0.5$, $\alpha=1$, and 20 ants per cycle. The clear-cut regime of the study area was optimized using the ACO inside ArcGIS with the optimal parameter values.

The ACO algorithm performed logically in the forest spatial planning problem under multiple objectives (Fig. 7). Both the mean and best utility increased during the optimization process. The mean utility of each ant cycle increased from 0.68 at the beginning to 0.94 at the end of optimization while the standard deviation of a cycle decreased from 0.06 to 0.02. Thus, the solution improved during the optimization process and the algorithm gradually converged, but it did not reach the stagnation stage after over 150 ant cycles (i.e. 3000 solutions, see Fig. 7).

The ACO optimization also gave rational results for the forest planning problem. The harvested volumes were quite even for the three 10-year sub-periods (Table 3). In addition, the risk of wind damage decreased significantly (Table 3). Most of the clear-cut stands were clustered at each period (Fig. 8). Clustered clear-cuts reduced the total length of forest edges. A few scattered clear-cuts may be required to fulfill the even-flow target of timber harvesting. In addition, the landscape configuration regarding stand height was also smoothed during the 30-year period (Fig. 9). Fewer old stands, which are the most vulnerable to be damaged, were located at the edges as compared to the initial forest. Furthermore, the fragmentation statistic CWED decreased from

Table 3. The harvested timber and the risk of wind damage output by the ACO.

Planning period	Harvest (m ³)	The risk of wind damage		
		Stands	Area (ha)	Edges (m)
1st period	9174	5	7.3	1501
2nd period	9379	4	9.3	439
3rd period	8919	1	2.3	864

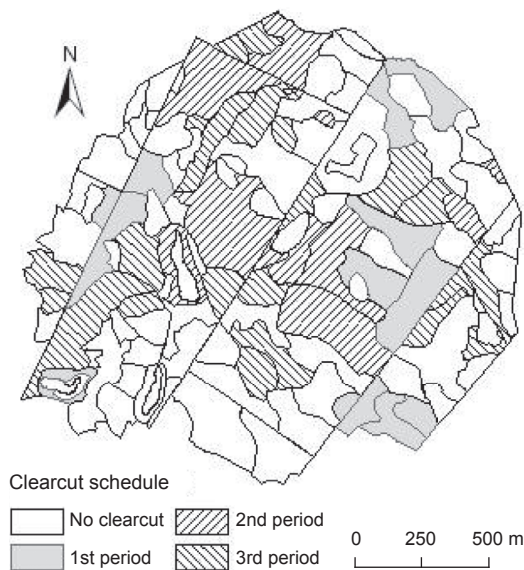


Fig. 8. The optimal clear-cut schedules output by ACO.

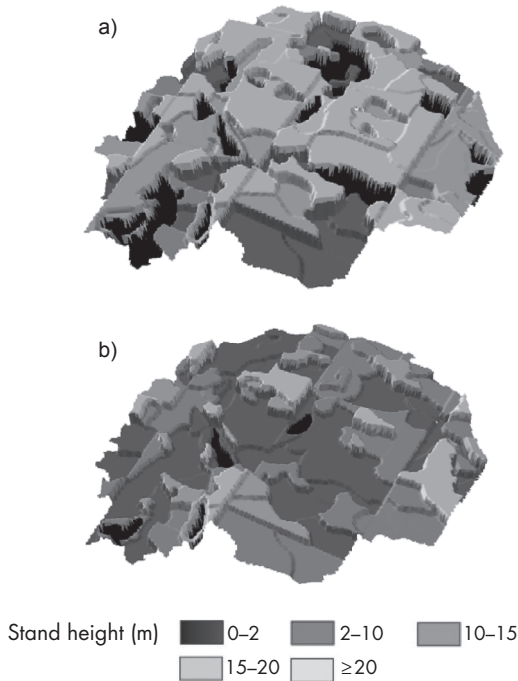


Fig. 9. The landscape configuration of stand height: (a) the initial forest (b) the forest at 30th year.

78 m/ha at the beginning to 65 m/ha at the end of simulation. Thus, the ACO algorithm found reasonable temporal schedules and spatial patterns of clear-cuts to decrease the risk of wind damage while maintaining even periodical harvests.

When comparing the performance of ACO with other heuristics simulated annealing (SA) and genetic algorithm (GA), it was found that in general, the three heuristic techniques resulted in very similar final output (Fig. 10). The final utility values outputted by ACO, SA and GA were 0.960, 0.966 and 0.958, respectively.

The optimization process within ArcGIS was very time-consuming. This program was coded and run in a PC with CPU of 1.5G Hz and RAM of 1G bytes using MS Windows 2000 operating system. In the pretest of the ACO coded in ArcGIS, the algorithm spent on average 21 h 54 min to complete the evaluation of altogether 1500 alternative solutions. The main reason for the slowness is that the spatial calculation (extraction of gaps, edge stands and edges, gap size calculation) in ArcGIS is complicated and therefore time-consuming. This kind of spatial calculation must also be repeated for each alternative solution. Therefore, the complexity of the ACO algorithm itself had little impact on the total computing time compared to spatial calculation in GIS. The same situations occurred with SA and GA implemented in ArcGIS. Furthermore, the computing time varied significantly. In the pre-tests with 1500 solution evaluations the minimum computing time was 16 h 6 min while the maximum was 27 h 11 min. The large variation in computing time is mainly due to differences in forest gap patterns, which induce variation in the time required for the spatial calculation.

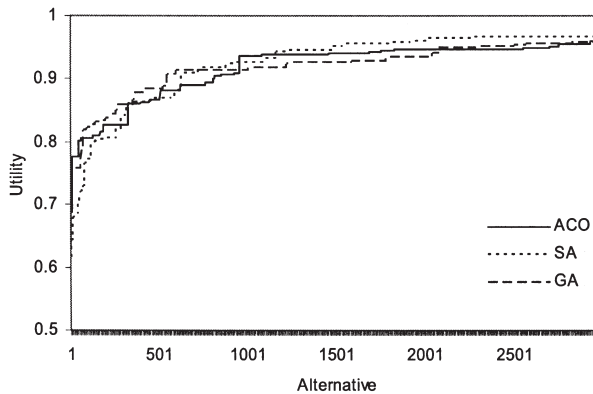


Fig. 10. Development of objective function value (utility) in ant colony optimization (ACO), simulated annealing (SA) and genetic algorithms (GA).

4 Discussion and Conclusions

This study revised the original ant colony optimization (ACO) to be specific for the combinatorial problems in forest planning. The revised ACO performed logically since the objective function value (utility) was improved and the algorithm gradually converged during the optimization process. The ACO gave reasonable solutions to the forest planning problem set in this work. The harvests of the three sub-periods were quite even. The clear-cuts were clustered so as to reduce the length of vulnerable forest edges. The landscape was smoothed to prevent old stands from being located at edges. The results were similar to an earlier study by Zeng et al. (2007) who used simulated annealing (SA), tabu search (TS) and genetic algorithms (GA) to solve a similar problem. Furthermore, the performance of the ACO was also found in our work comparable to two other heuristic optimization techniques, named as SA and GA.

The revised ACO is quite different from the original ACO algorithm because forest planning problems are different from the traveling salesman problem for which ACO was originally designed. Firstly, the treatment schedules are not real paths and therefore they have no distance values. Thus, the probability of selecting a schedule within a stand only depended on the pheromone trail laid by ants (Eq. 4). Secondly, in the traveling salesman problem a city may have several paths connecting to different cities so that all the cities are connected as a network. The number of ants in each cycle is usually set the same as the number of cities and each city is located an ant (Dorigo et al. 1996). However, the treatment schedules of forest stands do not form a network. Therefore, all the ants start their tour from the first stand and follow the same sequence to the last stand, although they could equally well start from different stands and follow different sequences. In addition, the best ant is maintained in the next ant cycle so that the importance of the best ant is reinforced in the optimization. Dorigo et al. (1996) also used a similar elitist strategy to reinforce the importance of the best ant since it proved to be efficient to search for the optimal solution.

The algorithm was programmed in the Visual Basic Application (VBA) of the GIS software

ArcGIS that accommodates the topology of polygons and arcs. The spatial complexity, such as extraction of gaps, edge stands and edges, calculation of gap sizes, and the links between edges and edge stands, could be done within the program automatically. On the other hand, the spatial calculation must be repeated for each alternative solution because different clear-cut patterns have different spatial patterns of gaps and consequently different edge stands and edges. This kind of spatial calculation is time-consuming, and the complexity of the heuristic algorithm itself has little impact on the total computing time for this specific problem. Thus, the parameter values, which affect the converging time and the possibility of getting trapped into local optimal, seemed to be important when solving the problems with spatial calculations. This is also why only 120 stands and 3000 alternatives were demonstrated in this study. The same routine can, however, be applied to a larger area and with more iterations without any technical problems provided that advanced computers are available or parallel system implemented. The computing may also be a reason why it is not common to call GIS from a heuristic algorithm during an optimization run. Instead, simplified algorithms for dealing with the topology are utilized to fasten the computation of spatial variables.

The initial pheromone and the lower bound of pheromone affected so that most schedules were pre-tested during a rather small total number of iterations. The lower bound strategy was also proved efficient by Sun et al. (2004). Higher values of lower bound (affected by parameter g) may, however, delay the convergence of the algorithm as demonstrated in this study. In addition, higher values of γ (Eq. 7) can reinforce the weights of the ants with higher utilities when calculating the new added pheromone trail. The difference between the best and worst schedules will also be enlarged and the optimization process will converge faster. The effects of higher values of parameter γ may, however, be limited by the lower bound if parameter g is set to a high value. Thus, the value of γ must be consistent with the value of g .

Moreover, with the same total number of evaluated alternatives, smaller number of ants per ant cycle increases the number of ant cycles

and therefore the algorithm converges faster. On the other hand, it has a higher risk of getting trapped into local optimal. Unlike parameters γ and the number of ants per cycle, a small change of parameter α will significantly affect the performance of the algorithm. This is because α directly affects the schedule selecting probabilities without limitation of the lower bound of the pheromones.

When searching for the optimal values of the parameters by sensitivity analysis it may not be enough to change only one parameter value and fix all the others. The optimal value of a parameter may depend on the values of other parameters. Therefore, it would be better to change several parameters simultaneously and study their joint impact on the algorithm. Pukkala and Heinonen (2006) proposed a method for finding the optimal parameter combination of heuristics used in forest planning. The use of this method would mean, however, that much more time is needed to find the optimal parameter values. Furthermore, the optimal values of parameters most probably depend on the number of iterations (convergence time). Usually, more iterations produce better solutions and need such parameter values for the algorithm that lead to slower convergence.

In conclusion, the revised ant colony optimization can be applied in forest planning to solve combinatorial problems since it performs logically, outputs reasonable results, and is comparable to other heuristics. In addition, the GIS software, which serves as a platform of the ACO program, expanded the program to solve the problems with spatial calculations (e.g. the risk management of wind damage as well as sustainable even flow timber harvesting). As the spatial calculation is complicated and time-consuming, it is important to carefully balance between the converging time and the likelihood of getting trapped in local optima when setting the parameter values and number of iterations.

Acknowledgements

This work is related to the research carried out under the Finnish Centre of Excellence Program (2000–2005) at the Centre of Excellence for

Forest Ecology and Management (Project No. 64308), University of Joensuu, Faculty of Forestry. It is also related to the work carried out through the SUNARE Research Program promoted by the Academy of Finland (2001–2004), under the project "Silvicultural strategies for managing wind and snow-induced risks in forestry" (SilviRisks, Project No. 52724). The support from the Academy of Finland, the National Technology Agency (Tekes), and the University of Joensuu is gratefully acknowledged. The authors would also like to thank Mr. Juha Hiltunen, Forest Centre Pohjois-Savo, for providing the forest stand data (X-forest-data).

References

- Bauer, A., Bullnheimer, B., Harti, R.F. & Strauss, C. 1999. An ant colony optimization approach for the single machine total tardiness problem. In: Proc. Congr. Evolutionary Computation (CEC'99). IEEE Press, Piscataway, NJ. p. 1445–1450.
- Bettinger, P., Sessions, J. & Boston, K. 1997. Using Tabu search to schedule timber harvest subject to spatial wildlife goals for big game. *Ecological Modeling* 94: 111–123.
- , Boston, K. & Sessions, J. 1999. Combinatorial optimization of elk habitat effectiveness and timber harvest volume. *Environmental Modeling and Assessment* 4: 143–153.
- , Graetz, D., Boston, K., Sessions, J. & Chung, W. 2002. Eight heuristic planning techniques applied to three increasingly difficult wildlife planning problems. *Silva Fennica* 36(2): 561–584.
- Bonabeau, E., Dorigo, M. & Theraulaz, G. 2000. Inspiration for optimization from social insect behaviour. *Nature* 406: 39–42.
- Boston, K. & Bettinger, P. 1999. An analysis of Monte Carlo integer programming, simulated annealing and tabu search for solving spatial harvest scheduling problems. *Forest Science* 45: 292–301.
- & Bettinger, P. 2002. Combining Tabu search and genetic algorithms heuristic techniques to solve spatial harvest schedule problems. *Forest Science* 48(1): 35–46.
- Borges, J.G., Hoganson, H.M. & Falcao, A.O. 2002. Heuristics in multi-objective forest planning. In: Pukkala, T. (ed.). *Multi-objective forest plan-*

- ning. Kluwer Academic Publishers, Dordrecht. p. 119–151.
- Colorni, A., Dorigo, M. & Maniezzo, V. 1991. Distribution optimization by ant colonies. In: Varela, F. & Bourguine, P. (eds.). Proc. Conf. Artificial Life. Paris, France. Elsevier. p. 134–142.
- Crowe, K. & Nelson, J. 2003. An indirect search algorithm for harvest-scheduling under adjacency constraints. *Forest Science* 49(1): 1–11.
- Dahlin, B. & Sallnäs, O. 1993. Harvest scheduling under adjacency constraints – a planning problem study from the Swedish sub-alpine region. *Scandinavian Journal of Forest Research* 8: 281–290.
- Dorigo, M. & Gambardella, L.M. 1997. Ant colonies for the traveling salesman problem. *BioSystems* 43: 73–81.
- , Maiezzo, V. & Colorni, A. 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics* 26(1): 29–41.
- ESRI. 2006. ArcGIS desktop – general concepts forum. [Internet site]. Available at: <http://forums.esri.com/Thread.asp?c=93&f=982&t=58595&mc=23#msgid228147>. [Cited 20 Feb 2006].
- Falcao, 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.
- FragStats. 2006. FragStats documentation. [Online document]. Available at: http://www.umass.edu/landeco/research/fragstats/documents/fragstats_documents.html. [Cited 14 Nov 2006].
- 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): 319–332.
- Kellomäki, S., Väisänen, H., Hänninen, H., Kolström, T., Lauhanen, R., Mattila, U. & Pajari, B. 1992. Sima: a model for forest succession based on the carbon and nitrogen cycles with application to silvicultural management of the forest ecosystem. *Silva Carelica* 22: 1–91.
- Lockwood, C. & Moore, T. 1993. Harvest scheduling with spatial constraints, a simulated annealing approach. *Canadian Journal of Forest Research* 23: 468–478.
- Lu, F. & Eriksson, L.O. 2000. Formation of harvest units with genetic algorithms. *Forest Ecology and Management* 130: 57–67.
- Maniezzo, V. & Colorni, A. 1999. The ant system applied to the quadratic assignment problem. *IEEE Trans. Knowledge Data Engin.* 11: 769–778.
- Merkle, D., Middendorf, M. & Schmeck, H. 2000. Pheromone evaluation in ant colony optimization. *Industrial electronics society. 26th annual conference of the IEEE.* 4. p. 2726–2731.
- Öhman, K. 2000. Creating continuous areas of old forest in long-term forest planning. *Canadian Journal of Forest Research* 19: 715–724.
- Pellikka, P. & Järvenpää, E. 2003. Forest stand characteristics and wind and snow induced forest damage in boreal forest. *Proceedings of the International Conference on Wind Effects on Trees*, held in September 16–18, 2003, University of Karlsruhe, Germany. p. 8.
- Peltola, H., Kellomäki, S., Väisänen, H. & Ikonen, V.P. 1999. A mechanistic model for assessing the risk of wind and snow damage to single trees and stands of Scots pine, Norway spruce, and birch. *Canadian Journal of Forest Research* 29: 647–661.
- Pukkala T. & Heinonen, T. 2006. Optimizing heuristic search in forest planning. *Nonlinear Analysis: Real World Applications* 7: 1284–1297.
- & Kurttila, M. 2005. Examining the performance of six heuristic optimisation techniques in different forest planning problems. *Silva Fennica* 39(1): 67–80.
- Sun, J., Xiong, S.W. & Guo, F.M. 2004. A new pheromone updating strategy in ant colony optimization. *Proceedings of the third international conference on machine learning and cybernetics*. Shanghai, 26–29 Aug. p. 620–625.
- Talkkari, A., Peltola, H., Kellomäki, S. & Strandman, H., 2000. Integration of component models from the tree, stand and regional levels to assess the risk of wind damage at forest margins. *Forest Ecology and Management* 135: 303–313.
- Venäläinen, A., Zeng, H., Peltola, A., Talkkari, A., Strandman, H., Wang, K. & Kellomäki, S. 2004. Simulation of the influence of forest management on wind climate on a regional scale. *Agricultural and Forest Meteorology* 123: 149–158.
- Zeng, H., Peltola, H., Talkkari, A., Venäläinen, A., Strandman, H., Kellomäki, S. & Wang, K. 2004. Influence of clear-cutting on the risk of wind damage at forest edges. *Forest Ecology and Management* 203: 77–88.
- , Peltola, H., Talkkari, A., Strandman, H., Venäläinen, A., Wang, K. & Kellomäki, S. 2006. Simulations of the influence of clear-cuttings on the risk

of wind damage on a regional scale over a 20 year period. *Canadian Journal of Forest Research* 36: 2247–2258.

- , Pukkala, T. & Peltola, H. 2007. The use of heuristic optimization in risk management of wind damage in forest planning. *Forest Ecology and Management* 241: 189–199.

Total of 33 references

Appendix 1. The algorithm of the revised ACO.

The algorithm of the revised ACO is listed below:

1. Initialization
 - Set initial pheromone $\tau_{ij}(0)=c$, c is a constant value.
 - Create the initial ant cycle (i.e. create ants by randomly selecting a schedule per stand) and calculate the timber harvesting, the risk of wind damage and the utility of each ant.
 - Search the BestAntId according to the utilities.
 - Set UpdateAntId in the cycle.
 2. Pheromone calculation
 - FOR $i=1$ TO standcount //standcount is the number of stands
 - IF stand i has more than 1 schedule THEN
 - FOR $j=1$ TO schedulecount (number of schedules) of stand i
 - Calculate $\tau_{ij}(t)$ by Eqs. (5)–(7).
 - NEXT j (schedule)
 - Check the lower bound pheromone value of schedules in stand i by Eq. (8).
 - NEXT i (stand)
 3. Create new ant cycle by updating ants in previous ant cycle
 - FOR $k=1$ to antCount //antcount is the number of ants per cycle
 - FOR $i=1$ TO standcount
 - IF stand i has more than 1 schedules THEN
 - Select a schedule for stand i by Eq. (4).
 - NEXT i (stand)
 - Calculate utility for the new ant.
 - IF ant k is not BestAntId THEN
 - Replace ant k .
 - Renew updateAntId.
 - NEXT k (ant)
 - Update the BestAntId.
 4. IF the stop criteria not satisfied THEN
 - GOTO step 2, repeat cycle.
 - ELSE
 - Stop.
-

Appendix 2. Notations.

$allowed_k$:	The cities allowed for ant k to select for its trip;
g :	A parameter that controls the minimum pheromone value;
i :	Index of forest stand;
j :	Index of schedule;
J_i :	Total number of schedules in stand i ;
k :	Index of ant number in an ant cycle;
n :	Number of ants per cycle;
$P_{xy}^k(t)$:	Probability of selecting path (x,y) for ant k at time t ;
Q :	A constant to prevent the cumulative pheromone level from being too high or too small to be used for calculating the schedule selecting probability;
T :	Index of sub-planning period (10 year);
t :	Time of ant moving;
$U(s)$:	Total utility of solution s (forest plan);
$U_h(s)$:	Sub-utility of timber harvesting;
$U_r(s)$:	Sub-utility of the risk of wind damage;
$U_d(s)$:	Sub-utility of the number of stands at risk;
$U_a(s)$:	Sub-utility of the areas at risk;
$U_e(s)$:	Sub-utility of the length of edges at risk;
$U_k(t)$:	Utility of ant k in cycle t ; it corresponds to $U(s)$;
W_h :	Weight of timber harvesting objective;
W_r :	Weight of the risk of wind damage objective;
W_d :	Weight of the number of stands at risk;
W_a :	Weight of the areas at risk;
W_e :	Weight of the length of edges at risk;
x, y :	Index of cities;
α :	Positive parameter to control the importance of pheromone
β :	Positive parameter to control the importance of visibility;
γ :	A positive parameter to control the effect of utility on the amount of added pheromone;
η_{xy} :	Visibility, i.e. the inverse value of the distance between cities x and y ;
ρ :	A coefficient such that $1 - \rho$ is the evaporation rate of the pheromone;
$\tau_{xy}(t)$:	Intensity of pheromone trail on path (x, y) at time t ;
$\tau_{ij}(t)$:	Pheromone of schedule j in stand i at time t ;
$\Delta\tau_{ij}(t)$:	New pheromone trail added by the ants that select schedule j in stand i ;
$\Delta\tau_{ijk}(t)$:	New pheromone added by ant k that selects schedule j in stand i ;
$\tau_{iMax}(t)$:	The maximum pheromone among the schedules of stand i at time t ;
$\tau_{iMin}(t)$:	The minimum pheromone among the schedules of stand i at time t .
