

A Comparison of One- and Two-Compartment Neighbourhoods in Heuristic Search with Spatial Forest Management Goals

Tero Heinonen and Timo Pukkala

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.

This study presents a comparison of the performance of four heuristic techniques with one- and two-compartment neighbourhoods in harvest scheduling problems including a spatial objective variable. The tested heuristics were random ascent, Hero, simulated annealing and tabu search. All methods seek better solutions by inspecting the neighbourhood solutions, which are combinations that can be obtained by changing the treatment schedule in one (one-compartment neighbourhood) or two (two-compartment neighbourhood) compartments. The methods and neighbourhoods were examined in one artificial and four real landscapes ranging from 700 to 981 ha in size. The landscapes had 608 to 900 stand compartments, and the examined planning problems had 2986 to 4773 binary decision variables. The objective function was a multi-objective utility function. The spatial objective variable was the percentage of compartment boundary that joins two compartments, both of which are to be cut during the same 20-year period. The non-spatial objectives were net incomes of three consecutive 20-year management periods and the remaining growing stock volume at the end of the third 20-year period. In another problem formulation, the total harvest of the first 20-year period was used as an objective variable together with the spatial objective. The results showed that a two-compartment neighbourhood was systematically and often clearly better than a one-compartment neighbourhood. The improvements were greatest with the simplest heuristics, random ascent and Hero. Of the four heuristics, tabu search and simulated annealing proved to be the best methods, but with a two-compartment neighbourhood the differences between methods were negligible.

Keywords 2-optimal heuristic, Hero, simulated annealing, spatial optimisation, random ascent, tabu search

Authors' address University of Joensuu, Faculty of Forestry, P.O. Box 111, FI-80101 Joensuu, Finland **E-mail** timo.pukkala@joensuu.fi

Received 23 February 2004 **Revised** 2 July 2004 **Accepted** 6 July 2004

1 Introduction

Heuristic optimisation methods are increasingly being used in forest planning (Borges et al. 2002). The most common reason for using heuristics instead of mathematical programming is the need to accommodate spatial objectives in the planning model. The spatial objectives may aim at controlling the maximum or mean size of continuous cutting areas (Lockwood and Moore 1992, Dahlin and Sallnäs 1993, Murray and Church 1995, Boston and Bettinger 1999, 2001, Bettinger et al. 2002), at decreasing fragmentation (Öhman and Eriksson 1998, 2002), or at aggregating old forests (Öhman and Eriksson 1998, 2002, Öhman 2000) or habitat patches of threatened species (Boston and Bettinger 2001, Van Deusen 2001, Bettinger et al. 2002, Kurttila et al. 2002, Palahí et al. 2004). Sometimes, spatial optimisation is related to the amenity values of the forest (Pukkala et al. 1995).

The heuristics most commonly used in forest planning problems include simulated annealing, tabu search, and genetic algorithms. In Finland, a method called Hero has been used for more than a decade for both non-spatial (Pukkala and Kangas 1993, Kangas et al. 1996) and spatial (Pukkala et al. 1995) optimisation problems. A simple method called random ascent (or random descent in minimisation problems) may be added to the list of common heuristics (Reeves 1993). A number of studies have also presented various modifications and combinations of the basic techniques (Bettinger et al. 2002, Falcão and Borges 2002, Kurttila and Pukkala 2003, Pukkala 2004b).

Most of the common heuristics belong to the category of local improvement methods. Their idea is to improve the solution gradually by changing it locally, and usually only a little at a time. Such a small change is called a move. Typically, a move consists of changing the treatment schedule of one compartment and doing nothing to the other compartments. An exception to this principle is the genetic algorithm, which uses more drastic moves (e.g. Reeves 1993). Often, the term “solution” refers to a combination of treatment schedules of stands or compartments.

Solutions that can be obtained from the current solution with one move form its neighbourhood. If a move consists of a change in just one com-

partment, the neighbourhood may be referred to as a one-compartment neighbourhood. If changes are made simultaneously in two compartments, we may talk about a two-compartment neighbourhood. It is also possible to talk about one- or two-optimal (or three-optimal, etc.) heuristics depending on how many compartments will have a changed treatment schedule (Reeves 1993, p. 13, Bettinger et al. 1999, 2002).

One problem of heuristics is their inability to find the global optimum with certainty. If the optimisation problem has strict constraints or achievement targets for the non-spatial goals, the tendency of getting trapped in local optima may severely hinder the attainment of the highest possible value of the spatial objective variable. It may be the case that once the solution has reached the border of the feasible region or the target level of a non-spatial goal, most one-compartment moves may be either non-feasible or clearly poorer than the current solution. One possibility to overcome this problem is to use a two-compartment neighbourhood (Bettinger et al. 1999). If the treatment schedules are changed simultaneously in two compartments, the non-spatial objective variables may remain practically unchanged but the spatial objective may improve significantly. The optimisation procedure may have reached a solution where the harvesting goal is met, and any change from this level would deteriorate the objective function value or fall outside the feasible region. However, when two compartments participate in the move, the total harvest does not need to change notably but the locations of cuttings may improve considerably.

The use of two-compartment moves has received very little attention in forestry. Bettinger et al. (1999, 2002) studied it with the tabu search technique in problems involving strict constraints, and a combination of one- and two-compartment moves performed better than one-compartment moves. There seem to be no studies with utility theoretic problem formulations, in which the feasibility of solutions often plays a smaller role than in problems involving strict constraints.

The objective of this study was to compare one- and two-compartment neighbourhoods in spatial utility maximisation problems, using heuristics that are based on local neighbourhood searches. The hypothesis was that a two-com-

Table 1. Information about test forests.

Variable	Area 1	Area 2	Area 3	Area 4	Grid
Area, ha	981	915	700	931	900
Number of compartments	763	677	608	803	900
Number of schedules	3568	2986	3228	4274	4773

partment neighbourhood should be superior to a one-compartment neighbourhood because the former is better in arranging the locations of cutting areas or habitat patches without requiring substantial changes in the other objectives. The analysed heuristics included random ascent, Hero, simulated annealing, and tabu search.

2 Material and Methods

2.1 Test Forests

The heuristic methods were tested in five test forests, four of which were real landscapes and one an artificial grid-like 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 (Table 1). Scots pine (*Pinus sylvestris* L.) is the most common tree species, followed by Norway spruce (*Picea abies* (L.) Karst.) and birch (*Betula pendula* Roth. and *Betula pubescens* Ehrh.). The age class distribution of the stands is rather uniform with a small peak in 20–40-year-old stands.

The Forest Centre of North Karelia has surveyed the forests using ocular compartment inventory. The inventory data were imported to the Monsu forest planning software (Pukkala 2004a). The simulation tool of Monsu was used to produce alternative treatment schedules for the compartments for a 60-year planning period consisting of three 20-year sub-periods. A regeneration cut with the necessary post-cutting treatments was simulated when the stand reached a minimum regeneration age or mean tree diameter. A thinning was simulated when stand basal area reached a so-called thinning-limit. All cuttings were simulated in the middle of the 20-year sub-period. The

simulations were based on the official silvicultural instructions but the timing of cuttings was varied to get more than one treatment schedule per compartment. The total number of schedules simulated for the compartments was about five times the number of compartments, i.e., on an average, one compartment had five different treatment schedules, differing mainly in the timing of cuttings.

As is typical in Finland, the four real landscapes included lakes and agricultural areas that partitioned the forest into rather small continuous forest areas and strips of forest. This kind of fragmentation due to water systems and other land uses restricts the possibilities of spatial optimisation to modify the structure of the landscape. Therefore, an artificial forest without any lakes and agriculture was created to see whether the optimisation results would be different in a continuous forest. The artificial landscape, referred to as Grid or Gridforest, 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 Gridforest were taken from two real landscapes (803 compartments from Area 4 and the remaining 97 compartments from Area 1).

2.2 Problem Formulations

The tested heuristic techniques were random ascent, Hero, simulated annealing and tabu search. They were used as implemented in the Monsu software (Pukkala 2004a), except that in searching for better solutions a two-compartment neighbourhood was used in addition to a one-compartment neighbourhood. The optimisation problem was formulated in a utility-theoretic way as follows:

Maximize

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

subject to

$$q_i = Q_i(\mathbf{x}) \quad i = 1, \dots, I \tag{2}$$

$$\sum_{k=1}^{N_n} x_{kn} = 1 \quad n = 1, \dots, N \tag{3}$$

$$x_{kn} = \{0, 1\} \tag{4}$$

where U is the total utility, I is the number of management objectives, a_i is the importance 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 stand n is treated according to schedule k , N_n is the number of alternative treatment schedules in stand n , and N is the number of stands.

The objective variables (q_i) can be any functions of decision variables (\mathbf{x}). The values of objective variables are calculated with “operators” or routines programmed in Monsu. This means that optimisation is not restricted to objectives that are linear combinations of decision variables. Optimisation proceeds so that once provided with an initial solution the heuristic optimisation algorithm makes changes in the combination of schedules that are currently in the solution, and calls a calculation routine that first calculates the values of objective variables using operators $Q_i(\mathbf{x})$ and then evaluates the total utility of the combination using Eq. 1. The new utility value is passed back to the optimisation algorithm, the functioning of which depends on how good the new combination (new solution) is compared to the previous one.

The sub-utility functions transform the absolute value of the variable measured in its own units into a relative sub-utility value. These functions are determined through the smallest possible, the target level, and the largest possible value of the objective variable, and the respective priorities.

The heuristic optimisation algorithms of Monsu are programmed so that only one management

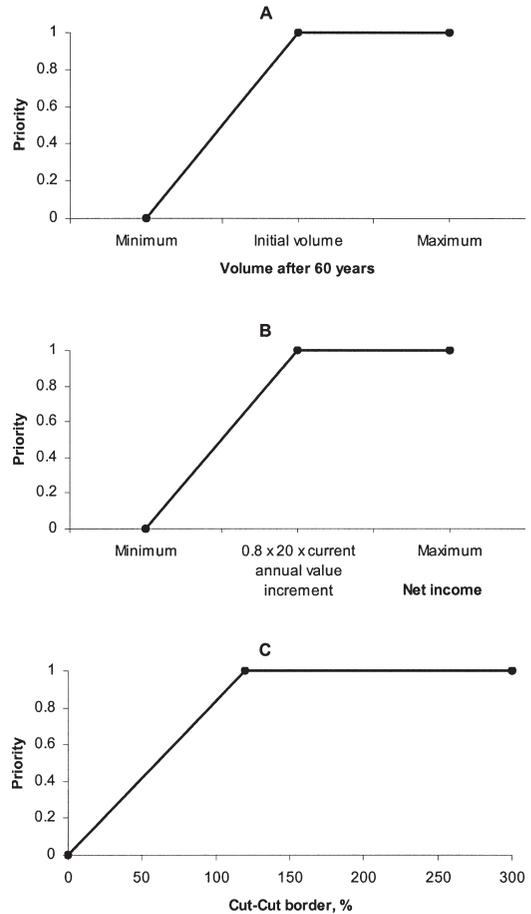


Fig. 1. Sub-utility functions for the remaining growing stock volume (A), 20-year net income (B) and cutting area aggregation (C) in the first optimisation problem (Eq. 5).

schedule of a compartment at a time can belong to the solution. Therefore, constraints of Eq. 3 need not be checked during optimisation, and there is no need to consider the feasibility of solutions; all solutions including exactly one treatment schedule per stand are feasible, and only such solutions are produced during the optimisation process.

The utility function that was maximised in this study included five objective variables

$$U = a_1 u_1(V) + a_2 u_2(I_1) + a_3 u_3(I_2) + a_4 u_4(I_3) + a_5 u_5(S) \tag{5}$$

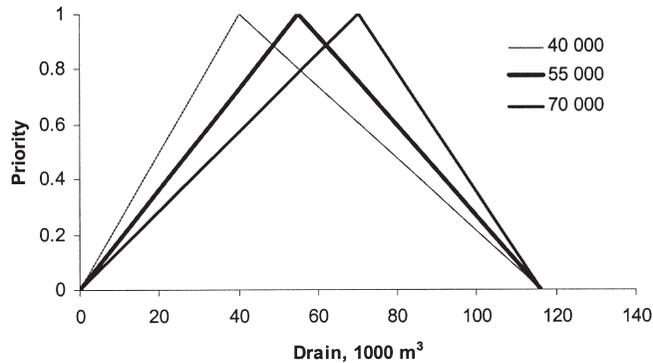


Fig. 2. Sub-utility functions for the total 20-year drain in the one-period optimisation problems (Eq. 6). The target drain is 40 000 m³, 55 000 m³, or 70 000 m³.

where a_i is the importance of objective i , u_i is a sub-utility function of objective i , V is growing stock volume at the end of the 60-year planning period, I_j is net income of the j th 20-year sub-period and S is a spatial objective variable. The importance of the volume objective (a_1) was 0.4 and all the other importances were 0.15. The spatial objective variable (S) used in the analyses was the proportion of such a compartment boundary (in percent), out of the total boundary length, that joins two compartments which are both to be cut during the same 20-year period. Maximisation of this objective variable tends to aggregate cuttings, resulting in both economic (reduced harvesting cost) and ecological (decreased fragmentation) benefits. Because the same compartment can be cut more than once during a long planning period, the theoretical maximum of the spatial objective variable was 300 percent of the total boundary length (every compartment cut during every 20-year sub-period).

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 variable. 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 the sub-utility (Fig. 1), with the result that the values of non-spatial objective variables were almost exactly equal to the target

level. This kind of problem formulation made it possible to concentrate the analysis of results on the spatial objective.

The same optimisation problem was solved 20 times with each of the four heuristics, and both with a one- and a two-compartment neighbourhood. This resulted in 800 optimisation runs. The purpose of 20 repeated optimisations with the same method and problem was to see how much the stochasticity of the method brings stochasticity in the solution and objective function value.

To examine the effect of problem formulation and to visualise differences between one- and two-compartment neighbourhoods, another objective function was formulated for the Gridforest, covering only the first 20-year sub-period:

$$U = a_1 u_1(H_1) + a_2 u_2(S_1) \quad (6)$$

where H_1 is the total harvested volume of the first 20-year sub-period and S_1 is the percentage of such a compartment boundary that joins two compartments which are both to be cut during the first 20-year period. Both a_1 and a_2 were 0.5. The sub-utility from the spatial objective variable was linearly proportional to S_1 . Three sub-utility functions were used for the harvested volume, each aiming at a different total harvest (Fig. 2). One utility function aimed at a harvest of 55 000 m³, which is practically the same as a random solution would produce. Therefore, the main task of optimisation was to aggregate cuttings, i.e. increase

boundaries that join cut stands from the value of a random solution (about 13% of the total boundary length). In the other sub-utility functions the target harvest was 15 000 m³ lower or higher than in a random solution, with the consequence that, in addition to relocating them, optimisation had to either decrease or increase cuttings.

2.3 Heuristics

The following explains briefly how the tested heuristics were used and which are the parameters through which the user can affect the algorithm. They are described in the way they are used with a one-compartment neighbourhood. A two-compartment neighbourhood is otherwise similar except that a move consists of changing the treatment schedule simultaneously in two compartments.

Random Ascent

In random ascent, a set of initial solutions is produced by selecting a random treatment schedule for each compartment from all schedules simulated for it. The best random solution is the initial solution of random ascent, which first selects a random compartment and then a random treatment schedule for the selected compartment. If the selected schedule improves the objective function value, it is included in the solution, otherwise it is not. The search procedure is stopped when the maximum number of trials, as specified by the user, is reached. To decrease problems arising from getting stuck to local optima, the whole process of generating an initial solution and applying random ascent can be repeated for a user-specified number of times. The parameters of the random ascent heuristic are: number of random searches to produce an initial solution, number of iterations (attempted moves) in random ascent, and number of repeated optimisations.

Hero

In Hero the initial solution is also the best of a user-defined number of random solutions.

Starting from the initial solution, the stands and their treatment schedules are explored sequentially to see whether another treatment schedule would improve the objective function value. If an increase is detected, the treatment schedule that improves the solution replaces the previous one. When all treatment schedules of all stands are examined in this sequential way, the process is repeated until no schedules that would further improve the solution can be found. The parameters of Hero are the number of random searches to produce an initial solution and the number of repeated optimisations. With a two-compartment neighbourhood the first compartment in which a change is made is selected in the same way as described above, i.e. sequentially, but the other compartment is selected randomly. Otherwise the method and the stopping criteria are exactly the same.

Simulated Annealing

Simulated annealing, like the previous methods, uses 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 consists of selecting first a random compartment and then a random schedule that would replace the current schedule of the selected compartment. Moves that improve the objective function value are always accepted. Non-improving moves are accepted with a probability of $p = \exp(-(U_{\text{New}} - U_{\text{Old}}) / T_i^{-1})$, where T_i is the current "temperature", and U is the objective function value. During the optimisation process, the temperature cools according to a given cooling schedule. The search stops when a user-specified stopping temperature is reached or a certain number of consecutive temperatures (5 in this study) go without any change in the solution. The user can affect simulated annealing through six parameters: number of random searches to produce an initial solution, starting temperature, freezing (stopping) temperature, cooling multiplier, number of iterations (attempted moves) in the initial temperature, and an iteration multiplier to get the number of iterations in the next

temperature. The cooling multiplier is a number smaller than one (usually 0.8–0.999) with which the current temperature is multiplied to get the next temperature. An iteration multiplier greater than one makes it possible to intensify search as a function of cooling.

Tabu Search

Tabu search is based on searching the neighbouring solution space before accepting one change in the solution. Typical of tabu search are also tabu lists (e.g. Glover 1989, 1990). They memorise recent moves, and can be used to prohibit them for some time. Typically, a move is kept in the tabu list for a certain number of iterations. This number is the initial tabu tenure of the move. Every iteration reduces the tabu tenures of all moves in the list by one. A move is again allowed once its tabu tenure has decreased to zero. In this study, iteration refers to the production of a set of random candidate moves (Table 2) and acceptance of one of them. Full neighbourhood was not used, since it is very large with two-compartment neighbourhood. The best non-tabu move is accepted. If all candidates are in the tabu list the one with the shortest tabu tenure is accepted. If a candidate move would yield a solution better than the best obtained so far, it is accepted even if the move is tabu (aspiration criterion; see Glover and Laguna 1993). Tabu search can also have lists other than the one described above, but they were not used in this study. In the tabu search method of Monsu, the initial tabu tenure is different for the entering schedule (schedule that enters the solution) and the leaving schedule. The initial tabu tenure of the entering schedule is one fifth of the initial tenure of the leaving schedule. This means that the whole compartment in which a change was made is forbidden for one fifth of the “length of tabu list” (i.e. initial tabu tenure of a leaving schedule). The user can control Monsu’s tabu search through the number of iterations, number of candidate moves per iteration, and the initial tabu tenure of a leaving schedule.

2.4 Optimisation Parameters

The values of the parameters that the optimisation methods need were “optimised” using the real forests and a one-compartment neighbourhood (Table 2). The optimisation software (Monsu) proposes reasonable parameter values. One parameter at a time was changed from this value to see if the change improves the objective function value. An improving change was maintained. However, changes that improved the objective function value only marginally at a high computational cost were not accepted.

Since the purpose of the study was not to compare different heuristics, but to analyse the use of different neighbourhoods, the main target of parameter setting was to find values that do not favour one of the two neighbourhoods, at least not the two-compartment neighbourhood. Because of this, the parameters were “optimised” for a one-compartment neighbourhood, and the same values were used with a two-compartment neighbourhood. An exception was tabu search, in which the parameters were different for one- and two-compartment neighbourhoods (Table 2). The reason was that with a two-compartment neighbourhood the number of tabu moves is twice as high as with a one-compartment neighbourhood, forcing the method to accept poorer moves. To counteract this, a shorter tabu list and a higher number of candidate moves were used with a two-compartment neighbourhood. It was checked that the parameters used with the two-compartment neighbourhood were not better with a one-compartment neighbourhood than the ones that were actually used with the one-compartment neighbourhood.

3 Results

The mean objective function values calculated from 20 repeated optimisations with the first problem formulation (Eq. 5) were systematically higher with a two-compartment neighbourhood (Table 3). The standard deviations were very small compared to the differences between means, indicating that the differences between neighbourhoods are significant and not caused by the

Table 2. Optimisation parameters (N is number of compartments; try is attempted move).

Method and parameter	Parameter value
Random ascent; 1- and 2-compartment neighbourhood	
Number of optimisations	5
Number of random searches to produce initial solution	$0.03 \times N$
Number of tries per optimisations	$20 \times N$
Hero; 1- and 2-compartment neighbourhood	
Number of repeated optimisations	5
Number of random searches to produce initial solution	$0.05 \times N$
Simulated annealing; 1- and 2-compartment neighbourhood	
Number of random searches to produce initial solution	$0.1 \times N$
Starting temperature	$0.1 \times 1/N$
Freezing temperature	$(0.1 \times 1/N)/20$
Cooling multiplier	0.9
Tries in initial temperature (I_0)	N
Tries in other temperatures	$1.1 \times I_{t-1}$
Tabu search, 1-compartment neighbourhood	
Number of iterations	$3 \times N$
Tabu tenure (tabu duration), number of iterations	$0.05 \times N$
Number of candidate moves per iteration	50
Tabu search, 2-compartment neighbourhood	
Number of iterations	$3 \times N$
Tabu tenure (tabu duration), number of iterations	$0.03 \times N$
Number of candidate moves per iteration	$0.2 \times N$

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. All objective function values were rather close to one, and the absolute differences were small. This was because the objective function was formulated in such a way that all goals except the spatial one could be fully attained in all optimisations, resulting in a total utility of 0.85 from the non-spatial goals alone. The differences in the objective function value therefore reflect differences in the attainment of the spatial objective.

The benefit of using a two-compartment neighbourhood was greatest with the Hero technique and smallest with the two best methods, i.e. simulated annealing and tabu search (Fig. 3). Therefore, the use of a two-compartment neighbourhood decreased differences in the performance of optimisation methods. The use of a two-compartment neighbourhood increased the share of the compartment boundary joining two cutting areas by

about 10 percentage points with the Hero method, and 5, 3 and 2 percentage points, respectively, with random ascent, tabu search and simulated annealing. Since the absolute values of the spatial objective (percentage of border joining two compartments both of which are to be cut during the same 20-year period) were less than 100%, in fact mostly between 50 and 80%, the relative changes were somewhat higher than the above-mentioned percentage points.

The standard deviations of the objective function values, calculated from 20 repeated optimisations, were usually smaller with a two-compartment neighbourhood than with a one-compartment neighbourhood (Fig. 4). An exception is simulated annealing, with which a one-compartment neighbourhood gave a smaller standard deviation. The standard deviations indicate that the use of a two-compartment neighbourhood improves the robustness of optimisation with all methods except simulated annealing.

The optimisation times were longest in tabu search with a two-compartment neighbourhood,

Table 3. Mean and standard deviation of objective function value (Eq. 5) in 20 repeated optimisations for different heuristics with one- and two-compartment neighbourhood, and the average time required for convergence. Boldface indicates the best mean or maximum objective function value found for the test forest.

Variable	Random ascent		Hero		SA		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
Maximum	0.9441	0.9495	0.9466	0.9579	0.9584	0.9613	0.9555	0.9600
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
Maximum	0.9265	0.9317	0.9265	0.9364	0.9395	0.9403	0.9388	0.9404
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
Maximum	0.9504	0.9564	0.9599	0.9700	0.9685	0.9718	0.9658	0.9718
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
Maximum	0.9506	0.9592	0.9527	0.9699	0.9705	0.9736	0.9667	0.9709
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
Gridforest								
Mean	0.9397	0.9443	0.9404	0.9556	0.9588	0.9612	0.9567	0.9590
Maximum	0.9421	0.9458	0.9428	0.9583	0.9601	0.9620	0.9580	0.9598
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

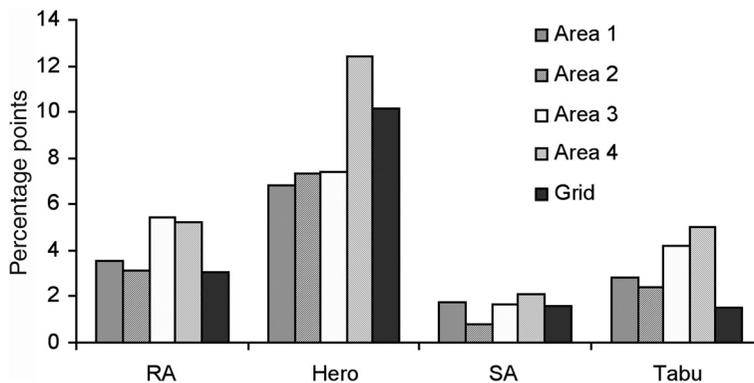


Fig. 3. Change in the spatial objective when using a two-compartment instead of a one-compartment neighbourhood (RA = random ascent, SA = simulated annealing, Tabu = tabu search).

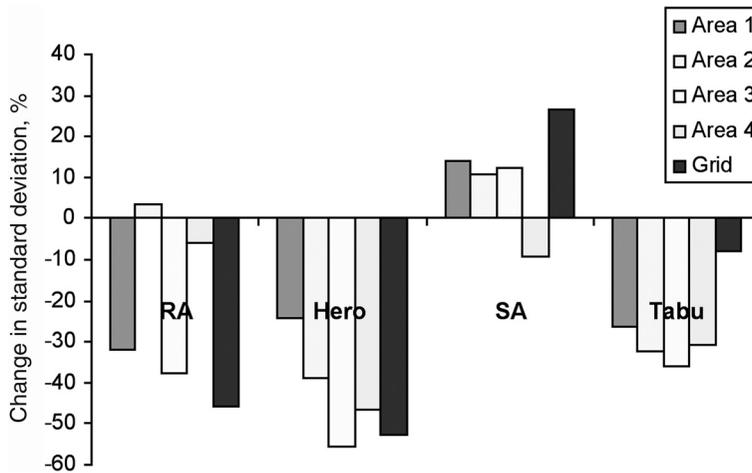


Fig. 4. Change in the standard deviation of objective function value, calculated from 20 repeated optimisations, when using a two-compartment instead of a one-compartment neighbourhood (RA= random ascent, SA= simulated annealing, Tabu= tabu search).

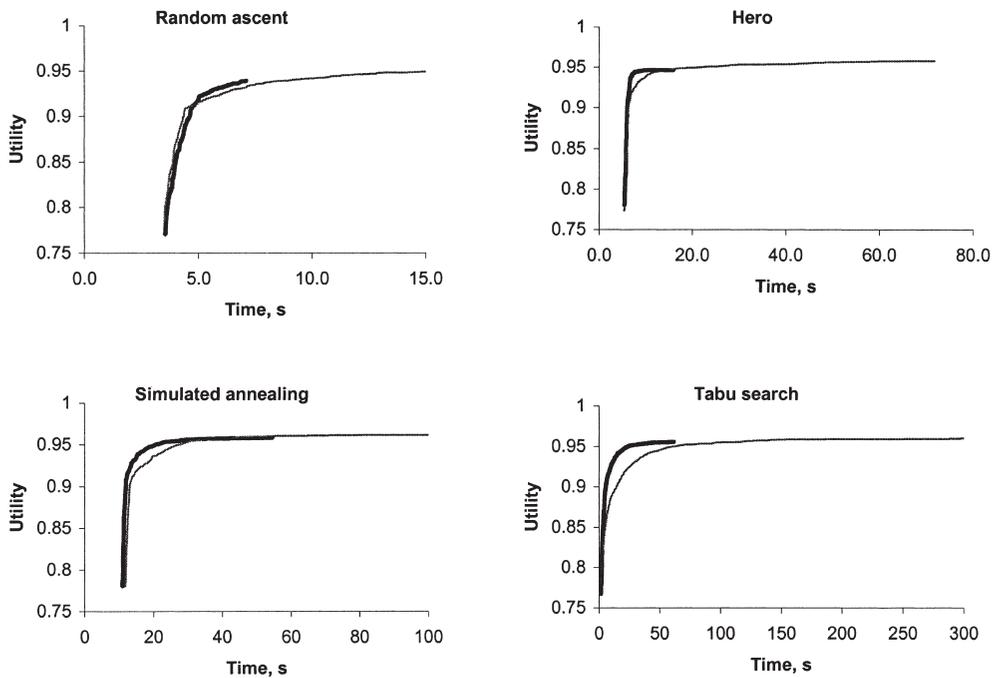


Fig. 5. Development of objective function value in the best (highest utility value) of 20 repeated optimisations in Area 1 when using a one-compartment (thick line) or a two-compartment neighbourhood (thin line).

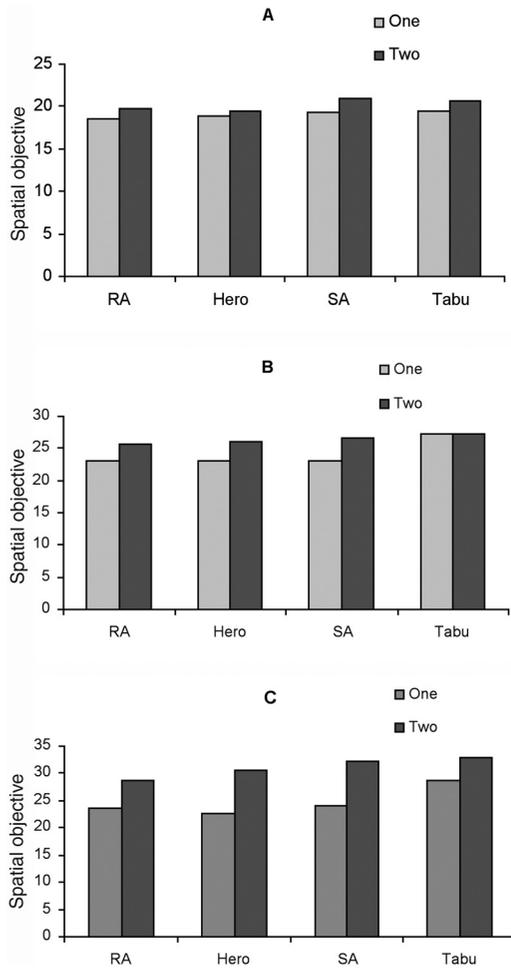


Fig. 6. Change in the value of spatial objective variable when using a one- or two-compartment neighbourhood in optimisation. The target 20-year drain is 40 000 m³ (A), 55 000 m³ (B) or 70 000 m³ (C). (RA=random ascent, SA=simulated annealing, Tabu=tabu search).

followed by Hero with a two-compartment neighbourhood (Table 3, Fig. 5). The use of a two-compartment neighbourhood always lengthened optimisation time. This is understandable because the evaluation of the effect of changes in two compartments takes more time than evaluating only one change. Another reason is related to the stopping criterion: in Hero, optimisation stops when no improvements can be found during one scan of

all treatment schedules. Because a two-compartment neighbourhood is better at finding improving moves, optimisation also continues for much longer. In tabu search, with a two-compartment neighbourhood 140 to 196 candidate moves per iteration were produced and evaluated but with a one-compartment neighbourhood only 50, which explains most of the computing time difference. 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 (Fig. 5).

The second problem formulation (Eq. 6), covering only one 20-year period, supports the results obtained with the first objective function (Eq. 5): the use of a two-compartment neighbourhood improves the attainment of the spatial objective (Fig. 6). However, the benefit clearly depends on the amount of cuttings and on the difference in cuttings between the initial solution and the target level. With the smallest cutting target (Fig. 6A), random initial solutions produce more cuttings than required, and the optimisation algorithm must drop cuttings schedules from the solution, preferably from compartments that are not surrounded by other cuttings. With a somewhat higher cutting target (Fig. 6B) the main task of optimisation is to place the cuttings in a more aggregated way. With the highest cutting target of 70 000 m³ (Fig. 6C) good solutions require more cuttings than random initial solutions would produce, located in an aggregated way. It seems that especially the last sub-problem with a 70 000-m³ cutting target is difficult to solve satisfactorily with a one-compartment neighbourhood. Fig. 7 shows that the improvements with two-compartment neighbourhoods are rather evident and clearly discernible with the naked eye. The figure also shows that tabu search is by far the best method to be used for solving this problem if a one-compartment neighbourhood must be used.

4 Discussion

The results suggest that the use of a two- instead of a one-compartment neighbourhood in the neighbourhood search clearly improves the results of

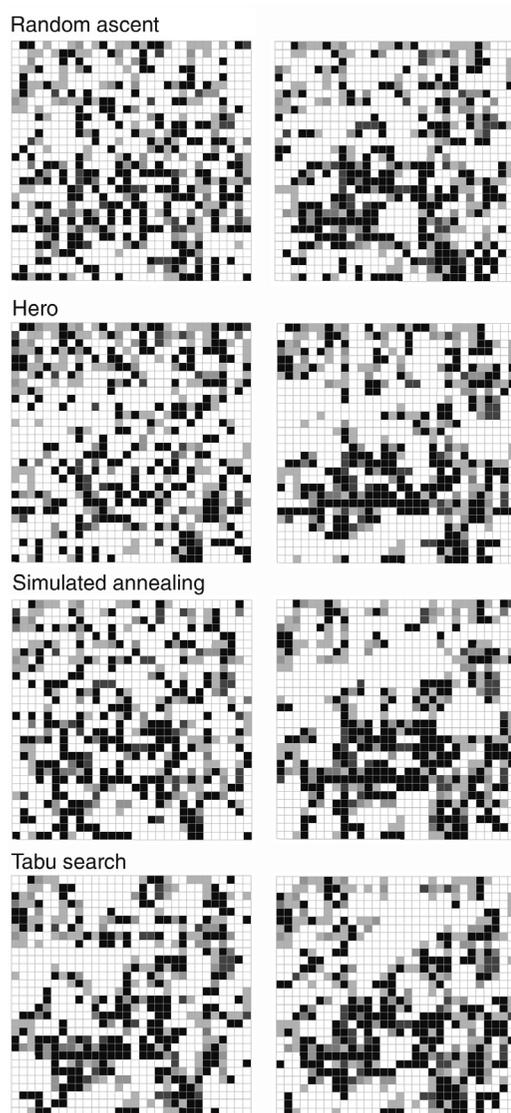


Fig. 7. Spatial distribution of 20-year cuttings in the optimal solutions found by different heuristics when using a one-compartment (left) or a two-compartment neighbourhood (right). In all solutions the total 20-year harvest is 70 000 m³. Dark shadings indicate regenerative cuts and light shadings thinning treatments.

local improvement heuristics in spatial problems. The problems of this study were formulated in a utility theoretic way so that the target levels of objective variables were given through 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 even 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.

Because two-compartment moves seem 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 this study work with a one-compartment neighbourhood (Palahí et al. 2004, Pukkala and Kurttila 2005). In these studies, a genetic algorithm was the worst method in non-spatial problems but the best method in spatial problems. However, in Bettinger et al. (2002) genetic algorithms were not particularly good in spatial problems.

Of the four methods tested, simulated annealing and tabu search were the best ones, simulated annealing being slightly better with the first problem formulation (Eq. 5, Fig. 1) and tabu search with the second (Eq. 6, Fig. 2). This was quite an expected result (see e.g. Bettinger et al., 1999, 2002, Pukkala and Kurttila 2005) because simulated annealing and tabu search have means to get away from local optima: simulated anneal-

ing by accepting inferior solutions with a small probability, and tabu search by accepting the best candidate move even when it is poorer than the current solution. However, it seems that the use of two-compartment moves greatly decreases the differences between methods, indicating that it significantly mitigates the harmful influence of local optima on the search process of the simplest heuristics.

Inspection of the temporal development of the objective function value during the optimisation process (Fig. 5) suggests that it might be beneficial to use a one-compartment neighbourhood in the beginning of the optimisation and a two-compartment neighbourhood near its end to refine the exploration of the solution space (Boston and Bettinger 2001, Bettinger et al. 2002). In the same way, many other combinations of techniques may work better than a single technique alone (Boston and Bettinger 2002, Bettinger et al. 2002, Falcão and Borges 2002, Kurttila and Pukkala 2003, Pukkala, 2004b). In this study the use of a two-compartment neighbourhood with the Hero technique can be interpreted as being a hybrid of random ascent heuristic and the original Hero method (Pukkala and Kangas 1993). It is possible to switch to another algorithm during optimisation, or to change the parameter settings. As the number of possibilities with various hybrid techniques is very high, useful research on heuristic optimisation techniques will certainly continue in the near future.

The performance of a heuristic algorithm depends on the parameters controlling it, in addition to the type of the optimisation problem. Therefore, no single study can give an overall ranking of methods. In this study, the optimisation parameters were “optimised” for a one-compartment neighbourhood, and the same values were used with a two-compartment neighbourhood. The exception was tabu search, for which good optimisation parameters were found separately for one- and two-compartment neighbourhoods. The use of one-compartment neighbourhood parameters with two-compartment moves in tabu search had resulted in a slightly better performance of the two-compartment neighbourhood in all four real forests, but clearly poorer in the artificial Grid-forest. In addition to clear differences in objective function values, the choice of optimisation

parameters in such a way that they do not favour a two-compartment neighbourhood is a justification for the conclusion that two-compartment moves are better than one-compartment moves in spatial optimisation problems.

References

- Bettinger, P., Boston, K. & Sessions, J. 1999. Intensifying a heuristic forest harvest scheduling search 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(2): 561–584.
- 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. p. 119–151.
- 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. 2001. Development of spatially feasible forest plans: a comparison of two modelling approaches. *Silva Fennica* 35(4): 425–435.
- & Bettinger, P. 2002. Combining tabu search and genetic algorithm heuristic techniques to solve spatial harvest scheduling problems. *Forest Science* 48: 35–46.
- Dahlin, B. & Sallnäs, O. 1993. Harvest scheduling under adjacency constraints – a case study from the Swedish sub-alpine region. *Scandinavia Journal of Forest Research* 8: 281–290.
- Dowland, K.A. 1993. Simulated annealing. In: Reeves, C.R. (ed.) 1993. *Modern heuristic techniques for combinatorial problems*. John Wiley & Sons, Inc. p. 20–69.
- Falcão, A.O. & Borges, J.G. 2002. Combining random and systematic search heuristic procedures for solving spatially constrained forest management scheduling problems. *Forest Science* 48: 608–621.
- Glover, F. 1989. Tabu search – Part I. *ORSA Journal of Computing* 1: 190–206.
- 1990. Tabu search – Part II. *ORSA Journal of Computing* 2: 4–32.

- & Laguna, M. 1993. Tabu search. In: Reeves, C.R. (ed.) 1993. Modern heuristic techniques for combinatorial problems. John Wiley & Sons, Inc. p. 70–150.
- Kangas, J., Loikkanen, T., Pukkala, T. & Pykäläinen, J. 1996. A participatory approach to tactical forest planning. *Acta Forestalia Fennica* 251: 24 p.
- Kurttila, M. & Pukkala, T. 2003. Combining holding-level economic goals with spatial landscape-level goals in the planning of multiple ownership forestry. *Landscape Ecology* 18: 529–541.
- , 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.
- Lockwood, C. & Moore, T. 1992. Harvest scheduling with spatial constraints: a simulated annealing approach. *Canadian Journal of Forest Research* 23: 468–478.
- Murray, A.T. & Church, R.L. 1995. Heuristic solution approaches to operational forest planning problems. *Operations Research Spektrum* 17: 193–203.
- Öhman, K. 2000. Creating contiguous areas of old forest in long term forest planning. *Canadian Journal of Forest Research* 30: 1817–1823.
- & Eriksson, L.O. 1998. The core area concept in forming contiguous areas for long term forest planning. *Canadian Journal of Forest Research* 28: 1032–1039.
- & Eriksson, L.O. 2002. Allowing for spatial consideration in long term forest planning by linking linear programming and simulated annealing. *Forest Ecology and Management* 161: 221–230.
- Palahí, M., Pukkala, T., Pascual, L. & Trasobares, A. 2004. Examining alternative landscape metrics in ecological forest landscape planning: a case for capercaillie in Catalonia. *Investigación Agraria, Sistemas y Recursos Forestales*. (In press).
- Pukkala, T. 2004a. Monikäytön suunnitteluohjelma Monsu. Ohjelmiston toiminta ja käyttö. University of Joensuu, Faculty of Forestry. 72 p.
- 2004b. Dealing with ecological objectives in the Monsu planning system. *Silva Lusitania*. (In press).
- & 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*. (Accepted).
- , Nuutinen, T. & Kangas, J. 1995. Integrating the amenity of forest area into numerical forest planning. *Landscape and Urban Planning* 132: 185–195.
- Reeves, C.R. (ed.). 1993. Modern heuristic techniques for combinatorial problems. John Wiley & Sons, Inc. 320 p.
- Van Deusen, P.C. 2001. Scheduling spatial arrangement and harvest simultaneously. *Silva Fennica* 35(1): 85–92.

Total of 28 references