Solving the Clear-Cut Scheduling Problem with Geographical Information Technology and Constraint Reasoning

Gunnar Misund
Bjørn Sigurd Johansen
Geir Hasle

SINTEF Informatics, P.O. Box 124, Blindern, N-0314 Oslo, Norway
{Gunnar.Misund, Bjorn.Johansen, Geir.Hasle}@si.sintef.no

Abstract:

In this paper we describe synergy effects of combining state-of-the-art Geographic Information Technology (GIT) with novel methods for planning and scheduling from the field of Constraint Reasoning (CR). We present a method for solving what we have called the Clear-Cut Scheduling Problem (CCSP), where the task is to assign clear-cutting times to regions in a given forest area over a long term horizon. The schedule must satisfy certain ecological, recreational and economical constraints, and, in addition, optimise on a number of partially conflicting criteria. Our approach is based on the combination of advanced spatial analysis and modern techniques for heuristic search. We have implemented a prototype Clear-Cut scheduling system called ECOPLAN. Empirical experiments have been carried out on a real-life test case consisting of a 500 stand forest property.
Keywords: *Forest Harvesting, Scheduling, GIS, Optimisation, Modern Heuristics, Tabu Search.*

1 Introduction

This article presents results from a research project at SINTEF Informatics carried out in 1994 (For a more detailed description of this work, see [MJHH95].) The main objective was to explore the assumed synergy in combining two separate R&D fields: Geographic Information Technology (GIT), and planning and scheduling based on Constraint Reasoning (CR). Forest management was chosen as an interesting and challenging application area. NORSKOG, a Norwegian association of forest owners, provided problem requirements and data needed to design, implement and test a software prototype for decision support in long term harvest scheduling.

During the past decade, the forest trade has faced a set of new challenges, both in Norway and in other parts of the world. Authorities and market segments demand accomplishment and documentation of sustainable forest harvesting. In addition to reaching economical objectives, the trade also has to take care of ecological concerns, such as wildlife preservation and biological diversity. Occasionally, recreational objectives have to be addressed.

Long term treatment schedules are considered one of the main vehicles in documenting adherence to external constraints, and, as control guidance for sustainable forest harvesting. The corresponding scheduling problem, which often is referred to as *spatially explicit* or *stand specific* treatment scheduling, has received considerable attention during the last ten years, both from the forestry research society, software vendors, and the operations research community. Still, there is no software available to support the

specific scheduling problems arising in Norwegian forestry. Existing methods do not handle wide planning horizons, or, they are not capable of handling spatial constraints and criteria.
The remainder of the paper is organised as follows: In section 2 we describe and discuss the Clear-Cut Scheduling Problem (CCSP) from a scheduling point of view. We outline some of the traditional approaches and describe the selected strategy. The design and implementation of an experimental prototype called *ECOPLAN* are described in section 3. A specific test case and selected results are presented in section 4. Some final remarks are given in section 5.

2 The Clear-Cut Scheduling Problem (CCSP)

We define the *Clear-Cut Scheduling Problem (CCSP)* as the assignment of clear-cut years to individual treatment compartments in a forest area. Given a forest area and its subdivision in stands, the task in CCSP is to generate a harvesting schedule for a given horizon which a) satisfies a number of constraints, and, b) strikes a careful balance between several criteria. In the following sections, we shall discuss the successful solution of an instance of the CCSP based on Norwegian legislation in general, and, in particular, restrictions on forest harvesting in the near-city areas of Oslo.

2.1 Description

The goal of the CCSP is to find a *complete*, *consistent*, and *optimised* clear-cut harvesting schedule. By *complete* we understand that each region must be assigned a time for future treatment. By *consistent* we mean that the schedule must not violate any hard constraints (A hard constraint is a relation which must be satisfied. A soft constraint may be relaxed). By *optimised*, we understand that the solution must optimise on certain criteria, e.g., by minimising or maximising the value of a defined objective function, and, it must satisfy all soft constraints to as large degree as possible. Time granularity is one year.
Topological description. The considered forest area is completely partioned into a number of non-overlapping regions. The underlying assumption is that each region is homogeneous with respect to the forest properties that are relevant to harvest scheduling (i.e., regions are stands).
Individual parameters/functions. For every region, several parameters/functions are given: The time of the most recent harvesting, minimum duration between harvests, maximum duration between harvests, optimal time between harvests (age/ripeness), the time it takes for trees to grow from 0 to a certain height, the area of each region, and the volume that may be harvested a number of years after the last harvesting.
Hard constraints. Before a region may be harvested, it is required that every neighbouring region has an average tree height of say at least 2 meters. We shall denote these hard constraints the *2-m constraints*.
Soft constraints. For economic and quality reasons, there are bounds on times between harvesting. These constraints may be relaxed in order to fulfill the 2-m constraint.
Criteria for an optimised schedule. Below we describe the four major optimisation criteria identified by forestry experts for the CCSP.
Optimal Harvesting Time. For every region, the harvesting time should be as close as possible to its optimal harvesting time.
Even Consumption. The estimated harvesting volumes for each year should be as close as possible to the average harvested volume.
Old Forest. The schedule should maintain a minimum area of forest above a given age

threshold, over the schedule horizon. We may, for instance, want to minimise the sum of violations of the old forest constraint.
Visual Impact. The schedule should minimise visual damage relative to a given set of viewpoints. By projecting the landscape (requiring a terrain model) to the respective viewing frames, it is possible to calculate the total area in these images which correspond to clear-cuts. We then may want to minimise the maximum clear-cut contribution (the ``worst'' visual impact) over all years.

2.2 Problem Solving Techniques

The CCSP is an example on a complex combinatorial optimisation problem. Focusing on the hard 2-m constraint, the CCSP may be regarded as a *Constraint Satisfaction Problem (CSP)* [Tsa93]. In particular, there are strong similarities between the CCSP and the *Graph Colouring Problem (GCP)*. Informally, the task in the GCP is to assign colours (from a given set of colours) to the nodes in a graph in such a way that no neighbours is given the same colour. The GCP belongs to the class of **NP**-complete problems [GJ79], for which there probably does not exist any efficient (polynomial) algorithm. Although there are additional constraints and objectives, our conjecture is that the CCSP is **NP**-hard (A proof is beyond the scope of this paper). We must therefore lower our expectations and concentrate on finding high-quality solutions in limited time. Adding the complexity, this problem indeed calls for efficient, robust and flexible optimisation techniques.
Mathematical Programming. *Linear Programming*, in particular, *Mixed Integer Programming (MIP)* and *Goal Programming (GP)* have been applied to the CCSP and similar problems [WMMK94]. Our initial attempts to formulate the CCSP as a MIP has lead us to conclude that this approach is not well suited, for the following reasons: 1) lack of flexibility in expressing constraints and objective criteria, 2) lack of support for mixed-initiative problem solving and 3) lack of repositories for heuristics to guide combinatorial search.
Systematic Tree Search (STS). Taking a Constraint Satisfaction Problem perspective on the CCSP, several backtracking tree search and consistency techniques are viable. *Standard Backtracking (SB)* may be seen as the basis for these techniques. SB will iteratively construct a solution by successively assigning values to the problem variables (i.e., assign a harvesting year to one region, then to another, and so on) while checking whether constraints are satisfied. If no value is possible for the current variable due to constraint violations, the algorithms will backtrack and try a new value for the previously instantiated variable. We have evaluated STS with several variable and value ordering heuristics in the context of finding a 2-m feasible solution. In initial empirical investigations it was not possible to obtain a solution within acceptable response limits, even for small CCSP problem instances (due to the exponential time complexity of STS).
Iterative Improvement Techniques (IIT). Over the past few years, these methods have shown remarkable performance in providing high quality solutions to scheduling problems in limited time [Dor95]. The basic idea is *neighbourhood search*, i.e., given any complete solution, generate a neighbourhood by applying a set of modification operators. The search for a better solution then proceeds iteratively by selecting the best neighbour as the new current solution. In this basic form, IIT is a hill-climbing algorithm which might get stuck in local optima. To remedy this, so-called *meta-heuristics* may be employed, e.g., *Simulated Annealing (SA)* [Kir83] or *Tabu Search (TS)* [Glo90]. A particularly nice feature of IIT in the context of decision-support is their *anytime* characteristic. The iterative problem solving process may be interrupted at any time, and the best solution so far is available for presentation. IIT has earlier been applied to forest management problems, e.g., to solve the afforestation problem[MJTVV92].

2.3 IIT - The Selected Search Strategy

Our selected search strategy for the CCSP is IIT with the Tabu-Search (TS) meta-heuristic [Glo90]. TS is composed of a neighbourhood operator, an evaluation function for neighbours, a tabu criterion, and an aspiration criterion. A method for generating an initial candidate solution is needed to initiate the iterative improvement process.

The Evaluation Function assigns goodness values to candidate schedules. We have selected a straightforward approach where the evaluation function is a weighted sum of the four optimisation criteria components described above. In addition, a penalty function for violations of the 2-m constraints is introduced as a component. The selection of appropriate weights is non-trivial.

Neighbourhood Operator. We have selected a neighbourhood operator which simply generates the neighbourhood by modifying exactly one harvesting year. The operator generates only local-feasible harvesting years (i.e. within the legal harvesting year interval) relative to the selected modified region. This operator is simple, but generates a large neighbourhood.

Initial Schedule. We have selected a greedy algorithm for generating the initial schedule. It assigns the local-optimal harvesting time to every region, where possible.

The Tabu Criterion specifies moves that are tabu and thus will not be executed. In TS, the iterative improvement basically consists of movement to the neighbour with the best value of the evaluation function. To escape from local optima, neighbours with certain defined properties are defined as tabu. Currently, we use a simple criterion, stating that we are not allowed to move to a neighbour which harvesting year has changed within a certain number of iterations.

The Aspiration Criterion. In TS, a move which is defined as tabu may be performed if allowed by an aspiration criterion. Our current choice of aspiration criterion checks for global improvement. If a move is deemed tabu, but will result in the best schedule encountered so far, the move will be performed anyway.

3 Design and Implementation of ECOPLAN

The ECOPLAN prototype is designed as a synthesis of four modules. The core of the system is a close integration of a scheduling engine called the *IIT Kernel* and a set of *GIT Services*. These two modules are embedded in an interface environment to facilitate interaction with the operator and communication with data sources. The functionality of the four components is briefly outlined below.

User Interface: Information on, and access to, input data. Control of the optimisation process (parameter settings, manual interruptions). Modification of existing constraints/ criteria and addition of new ones. Selections of output presentations, including animations. Currently implemented exclusively with development and evaluation of the IIT Kernel and the GIT Services in mind, with a parameter file as the main control.

Data Interface: Communication with external data sources. Conversion of input data. Output of results. Currently taken care of by means of plain ASCII files.

GIT Services: Generation of customised terrain models from scattered data, such as elevation contours, 3D data on road and stream networks, and geodetic points. Generation of consistent topological models from spaghetti data of site polygons. Spatial calculations of for example area, perimeter and distance. Visual viewpoint analysis. Preparation of colour coded digital maps. Preparation of data for 3D visualisation.

The GIT services are composed of a diversity of public domain tools and a suite of spatial analysis methods designed and developed particularly for the ECOPLAN module. The terrain generation and analysis, e.g., the Visual Impact analysis, is implemented by extensive use of the SISCAT library [sis95], [ADH95], a comprehensive, C++ based toolkit for construction of surfaces from various kinds of scattered data.

IIT Kernel: Modeling and management of all information relevant to the scheduling optimisation. Methods and algorithms for iterative schedule improvement. Implemented from scratch in the object-oriented language C++, in a way which makes it easy to plug in new iterative improvements techniques (see section 2.2), and to implement new constraints and criteria.

4 A Case and Some Results

To enable the investigation of behaviour and performance of the ECOPLAN prototype, we were provided with data on a forest area in the South-Eastern part of Norway.
Our test case is a 16 sq.km. forest, see figure 1, of which roughly speaking 85% is considered productive. The area is subdivided into approximately 500 stands. The average size of a stand is 28.000 sq.m., varying from 200 sq.m. to 148.000 sq.m. The forest is relatively young, about 60% of the total area consists of forest which is less than 30 years old.

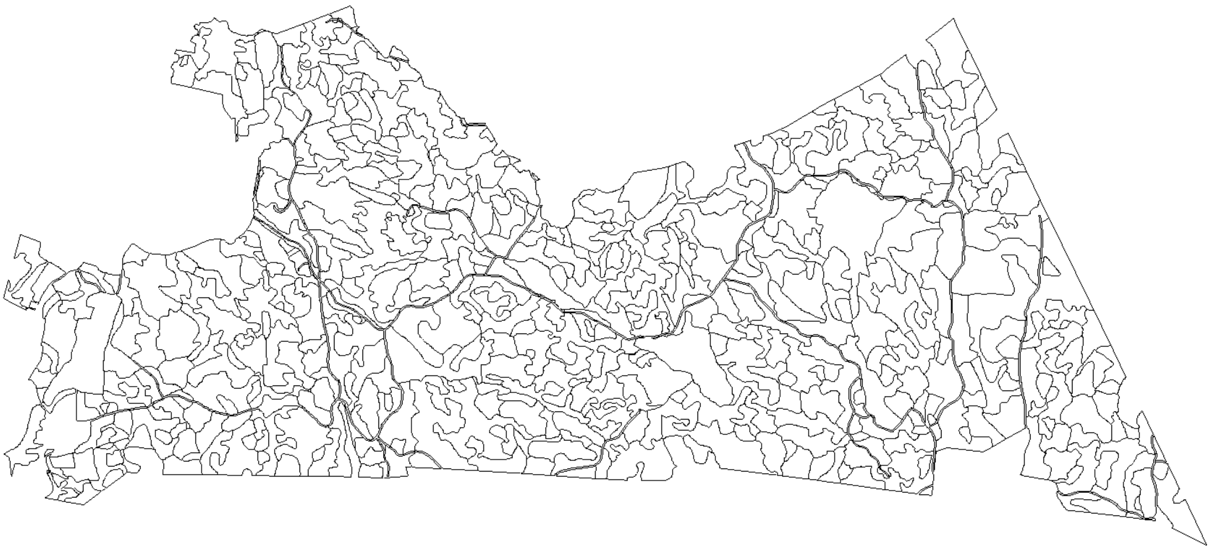

Figure 1: Geometry of the stands.

The ECOPLAN prototype handles site-specific information. Some simplifications have been made:

- All stands consist of only one single wood species.
- All stands are considered equal with respect to site quality.
- The volume growth function is designed as simply as possible, i.e., linear growth up to a given threshold age, and stagnation thereafter.
- The only forest treatment considered is clear-cutting.
- ``Old'' forest is defined to be more than 60 years for all stands.

The scheduling horizon is 100 years. For each year, we need to determine the set of stands to be treated. In the next sections we present results based on a series of experimental runs of ECOPLAN with different parameter settings. The presentation focuses on one single constraint or criterion at a time. However, the results are generated with all constraints and criteria simultaneously active.

4.1 2-m constraint.

The 2-m constraint implies that all neighbouring stands to a clear-cut region must be higher than 2 meters. Experiments showed that all 2-m constraints are satisfied after a relatively low number of iterations. Typically, the number of 2-m constraint violations decreases dramatically during the first 100 iterations. It takes equally many iterations to resolve the last few conflicts in a typical case.

Even Consumption. The Even Consumption criterion implies that the optimisation process seeks to distribute harvesting volume evenly over the scheduling period. In figure 2, we illustrate how this criterion gradually improves. Harvested volume as a function of time over the scheduling horizon is shown for three different schedules, the initial greedy solution, and the 600 and 1200 iteration schedules, respectively.

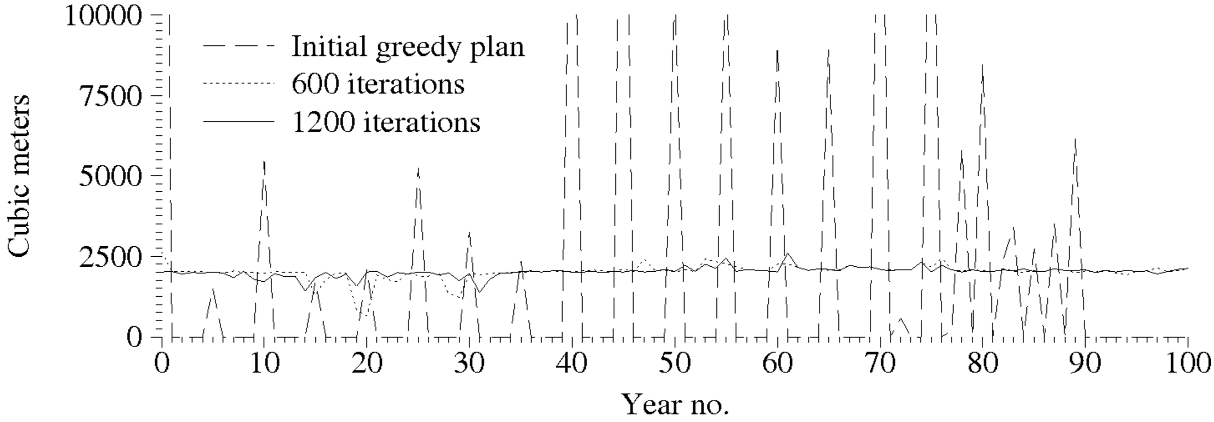

Figure 2: Development of the Even Consumption criterion.

The initial solution is, as described in section 2.3 and 4.1, generated by cutting the forest at the locally optimal harvest time. Due to the structure of the input data (the stands are classified according to five year intervals), we get large consumed volumes every 5th year and no activity in the intermediate years. The schedule is considerably improved after 600 iterations, and after 1200 rounds the yearly consumption seems to stabilise at an approximately even level.

Old Forest. Due to ecological considerations, the area percentage of old forest should be kept above a certain threshold level. In our case, we define old forest to be 60 years or older, and we want 40% or more of the total area to be occupied by stands of this age class.

The initial stage of the case forest is characterised by a large amount of young stands. Thus, the forest needs time to meet the old forest criterion. In the greedy solution, stands are cut as soon as they reach optimal ripeness which is above the old forest age. Hence, we get an uneven, oscillating pattern.

This pattern is considerably improved during the optimisation process. In an optimised schedule, we reach a stable situation after approximately 45 years of harvesting. The desired level of old forest is reached, and the oscillating behaviour is avoided.

Visual Impact. In figure 3, we have simulated the visual impact from a given viewpoint in a certain year of a good (right) and bad (left) schedule. In the 3D views, clearings and young stands are rendered light grey, while older forest is darker. The left view presents a landscape which may be characterised as totally demolished, while the right visual impact is satisfactory.

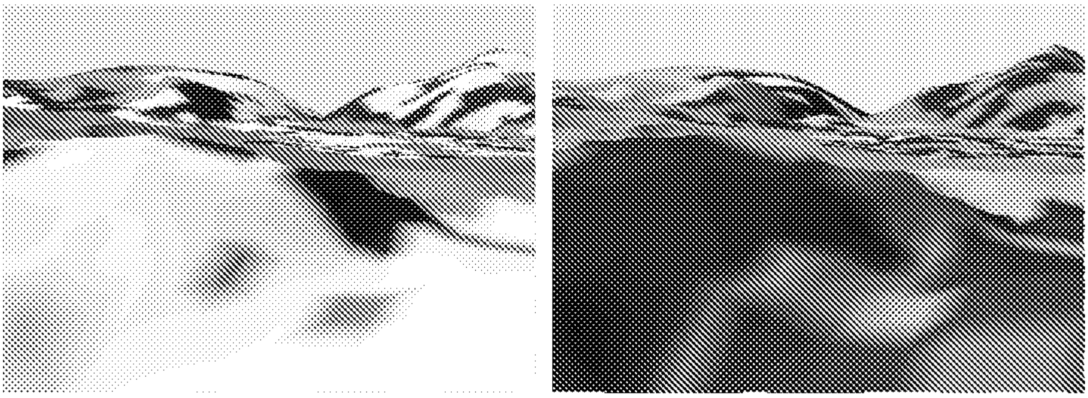

Figure 3: Visual Impact criterion, bad and good vistas.

The optimisation procedure will favour a schedule were many of the vistas are visually pleasant. Solutions that contain visually unpleasant harvesting patterns are heavily punished.

Optimal Harvest Time. The initial solution is defined as a schedule where the Optimal Harvest Time criterion is maximised, i.e. that every stand is clear-cut exactly in the year when the forest is considered to be as ``ripe'' as possible. In contrast to the other criteria, the Harvest Time goodness will inevitably decrease during the optimisation process when all constraints/criteria are active. After 1200 iterations, we reach an acceptable result. Approximately 25% of the regions are still optimally harvested. Many stands are clear-cut at near-optimal time. Relatively few stands are found in the sub-optimal parts of their feasible interval.

4.2 Performance

The ECOPLAN prototype is implemented in a UNIX environment, and the tests have been conducted on standard mid-range workstations.

To establish the values of the various parameters, an optimisation session typically started with test-runs with a low number of iterations, and perhaps with only one or two active constraints. Finally, an optimisation process involving a high number of iterations was executed to gain a satisfactory solution.

One single pass involves a preprocessing phase (see section 3) and a number of improvement steps. The preprocessing typically lasts from 20 to 30 seconds up to a few of minutes, depending on the input data. The ECOPLAN prototype generates from ten to hundred new schedules per minute. Not surprisingly, the performance has shown to be relative to the number of active constraints. However, there is a significant potential in optimising the implementation.

5 Final Remarks

The ECOPLAN Project. As presented in section 4, the scheduling strategy for the CCSP reported in this paper has shown excellent performance on real data. Based on the ECOPLAN prototype, an R&D project was launched in September 95. The main goal is to develop a module for long term treatment scheduling under economical and ecological constraints and criteria, customized for Norwegian forestry.

The project will undertake comparative experiments with alternative IIT meta-heuristics as well as other search techniques. The goal is to develop search strategies that are able

to handle CCSPs with a number of regions which is an order of magnitude larger than in the case data described in this paper.
Major efforts will be devoted to the refinement of the underlying model from a forestry point of view. Integration with stand simulation software, and the accommodation of several types of treatment (thinning, sparse cutting) are important issues in this context. Special attention will be paid to develop a simple, intuitive, and efficient user interface. Flexible and efficient methods for integration with external information repositories will be provided by implementing an advanced software integration platform.
GIT and Constraint Reasoning. The extra efforts needed to establish and maintain the multi-disciplinary profile of the project proved to be highly rewarding, and a necessary condition to achieve our goals. The supply of well-known methods and techniques for management and analysis of spatial information is large and varying. This is also true for planning and scheduling. However, by combining the technologies, we achieved results beyond expectations at a relatively modest cost.
We suggest that generalised versions of the CCSP problem, such as area planning in the municipalities, agricultural management, campaign planning in marketing and advertising etc., may be solved by similar approaches.

6 Acknowledgements

We would like to thank NORSKOG for their supply of the Clear-Cut Scheduling Problem and access to real-life case data. We also are in debt to our fellow researchers Trond Vidar Stensby and Per Kristian Nilsen for their competent contributions in the implementation process of the ECOPLAN prototype. Jørgen Haukland at Gjøvik College provided valuable input regarding the the forestry issues of the problem. Finally we want to thank Torgrim Johan Castberg for permitting the use of the forest case data in the SINTEF research project, and Arne Løkketangen at Molde College for introducing us to Tabu Search and providing us with valuable comments and suggestions for improvement.
This work has been partially supported by the Norwegian Research Council, project MOI.31386, and SINTEF Informatics, project 3391 1000.

References

ADH95
Erlend Arge, Morten Dæhlen, and Øyvind Hjelle. Mathematical Software for Terrain Modelling. In Jan Terje Bjørke, editor, *SCANGIS '95 - Proceedings of the 5th Research Conference on GIS, 12th 14th June 1995, Trondheim, Norway*, May 1995. ISBN 82-993522-07.

Dor95
Jürgen Dorn. Iterative Improvement Methods for Knowledge-Based Scheduling. *AI Communications*, 8(1):20 -- 34, 1995.

GJ79
M R Garey and D S Johnson. *Computers and Intractability*. Freeman, 79.

Glo90
F Glover. Artificial Intelligence, Heuristic Frameworks and Tabu Search. *Managerial and Decision Economics*, 11:365--375, 90.

Kir83
S Kirkpatrick. Optimization by Simulated Annealing. *Science 220*, 83.

MJHH95
G. Misund, B. S. Johansen, G. Hasle, and J. Haukland. Integration of Geographical Information Technology and Constraint Reasoning - A Promising Approach to Forest Management. In Jan Terje Bjørke, editor, *SCANGIS '95 - Proceedings of the 5th Research Conference on GIS, 12th 14th June 1995, Trondheim, Norway*, May 1995. ISBN 82-993522-07.

MJTVV92
R Mørk Jørgensen, H R Thomsen, and R V Valqui Vidal. The Afforestation Problem: A Heuristic Method Based on Simulated Annealing. *European Journal of Operational Research 56*, pages 184--191, 92.

sis95
SISCAT - The SINTEF Scattered Data Library (version 2.1). Technical report, SINTEF Informatics, Oslo, 1995. Reference manual.

Tsa93
E Tsang. *Foundations of Constraint Satisfaction*. Harcourt Brace & Co, 93.

WMMK94
G Weintraub, A Jones, A Magendzo, M Meacham, and M Kirkby. A Heuristic System to Solve Mixed Integer Forest Planning Models. *Operations Research*, 42(6):1010--1023, 11 94.