DIVERSIFIED SPECIES TECHNICAL NOTE

Number: DSTN-018
Date: July 2011

Comparison of Multi-objective Optimisation Techniques: Land Use Modelling

Summary

This note is technical in nature. It describes a number of problem-solving techniques that have been applied over the past 50 years to problems as wide ranging as aircraft design and land use decisions. It is fair to say that the use of these techniques is still very much in its infancy in New Zealand forestry science and land management, but their potential application remains huge.

Solving real-world land use problems is a complex task. Checking or enumerating all possible solutions may take thousands of years, even when using the fastest supercomputers. Having multiple objectives and spatial constraints exacerbates the problem by increasing model complexity and exponentially increasing computing requirements. The insight is to use intelligent algorithms (heuristics) that follow search paths that show potential for getting you nearer to your goal.

Determining the most appropriate heuristic to use for a land use problem is influenced by the number of objectives, available data, problem size (number of variables), spatial constraints, and the skill and experience of the model developer. Different heuristics are based on different search philosophies, with the consequence that they may well find different solutions. The combination of the differences in the appropriateness of a heuristic with the potential differences in solutions uncovered difficulties when attempting to compare heuristics for land use modelling.

This technical note briefly describes key heuristics for land use problems and touches on measures for comparing them. The research progress on the comparison of two heuristics, the Metropolis algorithm and the genetic algorithm. A specific land use problem using data from a large central North Island forestry, sheep, beef and dairy property was used for this comparison. For land use change driven by the need for environmental outputs without compromising return on investment, the Metropolis algorithm found a single solution that met these objectives. Further research, using genetic algorithms will enable us to find a set of efficient solutions from which the decision makers can choose based on their preferences and values.

Authors: Oliver Chikumbo

Introduction

Solving real-world land use problems is necessary for the informed management of our land resource. The complexity of real-world problems makes this a difficult task. This technical note describes some land use problems and the assessment of heuristic algorithms to address them, provides a short overview of key heuristic techniques, and reviews progress on the comparison of two of them.

Introducing Land Use Problems

Landscape modelling problems are extremely variable. They are dealt with at different planning levels, namely; strategic, tactical and operational. The real world is also characterised by uncertainty and conflicting goals. Scarcity or gaps in the availability of data or in relevant variables, especially critical data and at any planning level, influences the type of analysis and subsequent techniques that can be used.

Given the considerable variety for land use modelling, we confine our discussion to problems where outcomes are defined, and what we are searching for are the inputs needed

Future Forests Research Ltd, PO Box 1127, Rotorua. Ph: 07 921 1883 Email: info@ffr.co.nz Web: www.ffr.co.nz

DIVERSIFIED SPECIES TECHNICAL NOTE

Number: DSTN-018
Date: July 2011

to achieve those outcomes. More explicitly, the inputs need to be defined in terms of the timing of the application, the magnitude and frequency of application, and their spatial layout (for spatial problems). This then broadly defines them as optimisation problems, and more specifically as control problems. The spatial component (if relevant) turns the problem into a combinatorial one, one of the hardest problems to solve mathematically.

Adding more than one objective to an optimisation problem adds complexity. For example, you may want to optimise a structural design that is both light and rigid. Because these two objectives are in conflict with one another, a trade-off is required. There is an optimal lightest design, one that has the stiffest design, and an infinite number of designs that offer some compromise of weight and stiffness. This set of trade-off designs is known as a Pareto set. A design is judged to be Pareto optimal if it is not (completely) dominated by any other design. A Pareto optimal design must be better than another design in at least one aspect. If it is worse than another design in all respects, then it is dominated and is not Pareto optimal.

Suppose there is clarity on the planning level and the data available to address a problem. Then the size (the number of variables) of the problem presents challenges that can influence the optimisation technique used. The skill, bias and experience of the modeller will further influence the analysis approach adopted.

Assessing Algorithms

An accumulation of problematic issues such as data scarcity, the level of planning, size of problem, and the modeller's skill and bias, creates complexity that makes it difficult to make simple comparisons of different techniques and approaches. The best way to make comparisons is to compare the results of the same problem analysed by different heuristic search algorithms. The parameters in Table 1 may be useful for judging the quality of the solution obtained[1].

One classic comparison example is the comparison between a heuristic (i.e. a method used to rapidly come to a solution that is hoped to be close to the best possible answer, or optimal solution) and conventional optimisation techniques (such as Linear Programming). The comparison was frequently used in top scientific journals in the mid-nineties which focused on determining the "optimal" solution. It is, however, not possible to verify an optimum when dealing with *NP*-hard problems, where *NP*-hard refers to non-deterministic problems whose computation time can be quantified by a simple polynomial function of the number of variables. Non-deterministic problems permit more than one choice of next move at some step in the computation.

By definition, *NP*-hard problems are those that experience an exponential increase in computation time with every unit increase in the number of variables, becoming rapidly prohibitive in cost as the number of variables increases. *NP* problems are considered "hard" in the sense that they are not currently solvable in deterministic (i.e. permitting at most one next move at any step in a computation) polynomial time[2].

If you can solve a *P* (i.e. deterministic polynomial time) problem with a conventional technique, it means that the solution is optimal and no other method can improve it. For heuristics, however, there is no guarantee that they will perform equally for that *P* problem. The key point is that for *NP*-hard problems, conventional techniques will simply not yield any results, as it would take thousands of years even on a supercomputer to find a solution. This is where heuristics come in. By their very nature of not guaranteeing optimality they therefore warrant a more thoughtful comparison.

With heuristics, issues such as the differentiability of functions and standardization of data no longer become necessary. However, this flexibility, which makes heuristics application-neutral, comes at a cost, as heuristics tend to be problem-specific and have to be tailored to the data and problem definition.

Future Forests Research Ltd, PO Box 1127, Rotorua. Ph: 07 921 1883 Email: info@ffr.co.nz Web: www.ffr.co.nz

DIVERSIFIED SPECIES TECHNICAL NOTE

Number: DSTN-018
Date: July 2011

Unfortunately the concept taken from conventional techniques – that data sets can be standardized and therefore it is possible to have a "one size fits all" technique or tool – does not generally apply to heuristics. While it may apply to tightly specified subsets of problems, such an approach would place large restrictions on land use problem types, severely limiting problem solving for *NP*-hard problems.

Table 1. These measures help in understanding why one algorithm is successful or more successful than another algorithm. Without them, there is no real understanding of how the algorithms are actually searching the search space.

Name	Purpose
Best found cost	Useful for comparison of algorithms where one algorithm is clearly superior to another, but gives no indication of why it is superior.
Number of repairs to solution of cost TARGET	Implementation independent measure of how fast an algorithm finds a certain quality of solution. This is a finer measure than the best-found cost.
CPU seconds for run	Practical measure if the algorithm is to be used to solve real problems.
Local minima cost over a run: Mean and standard deviation	Measure of the quality of search space visited by the local search algorithm.
Number of moves between local minima	This gives an indication of how diverse or intense the search is.
Number of repetitions during search	This gives an indication of how much cycling is going on during the search.
Entropy: Mean, Standard deviations, Min, Max	This measures how well the search space is covered by the local search algorithm.

The Theory Behind the Techniques

Sitting at the heart of all optimisation techniques is "hill-climbing", which is essentially an iterative movement that relies on finding the "better" local neighbouring potential solution. Other techniques use different approaches that improve on the hill-climbing iterative movement. For example, simulated annealing and genetic algorithms rely on probabilistic, memoryless decisions, whereas tabu search is based on the use of a memory of previously visited solutions. Genetic algorithms use probabilistic transition rules to determine the next point from a database of movements simultaneously "climbing" a search surface in parallel. Therefore, hill-climbing, tabu search, simulated annealing and genetic algorithms correspond to different search philosophies, that is, simply, memory-based and probabilistically-driven. In this section, more detail will be provided for each search technique.

Hill-climbing

Hill-climbing is initiated by randomly selecting a potential solution and then iteratively improving on this solution by exploiting other potential solutions in the neighbourhood, i.e. the heuristic climbs in the steepest permissible direction. Therefore, each step of the algorithm makes locally the best choice and hence it is called a greedy algorithm. It avoids an exhaustive search and makes maximum use of local information. Hill-climbing requires only a limited amount computer memory (only the current state is

Future Forests Research Ltd, PO Box 1127, Rotorua. Ph: 07 921 1883 Email: info@ffr.co.nz Web: www.ffr.co.nz

DIVERSIFIED SPECIES TECHNICAL NOTE

Number: DSTN-018
Date: July 2011

stored), is easy to implement efficiently and works well for most unimodal type functions where the solution space has only one extremum.

Hill-climbing techniques make moves that improve only the cost function or leave its value unchanged (the so-called sideway moves). Four flavours of hill-climbing can be identified[3]:

- Steepest ascent with guaranteed progress: all points in the neighbourhood are evaluated, and the point that gives the greatest improvement is selected. If there is no point that is strictly better, then the process is stopped;
- First ascent with guaranteed progress: the first point found in the neighbourhood that is better than the current point becomes the new current point. If there is no point that is strictly better, then the process is stopped;
- Steepest ascent with neutrality: all points in the neighbourhood are evaluated, and the point that gives the greatest improvement is selected. If the best point has equal evaluation to the current point, it becomes the new current point; and
- First ascent with neutrality: the first point found in the neighbourhood, which is better than the current point becomes the new current point. If the best point has equal evaluation to the current point, it becomes the new current point.

Any of these versions of hill-climbing can be combined with random search. When the hill-climbing gets stuck at a local optimum, then it chooses another random starting point and starts all over again. This is called hill-climbing with random restarts. Hill climbing has the advantage that it is very easy to implement. It has the disadvantage that it gets stuck in local optimum. Sometimes the random restart method is effective at overcoming this problem, and sometimes it is not.

Tabu Search

The main component of tabu search is the use of memory. Discoveries of more refined ways to exploit this memory and of more effective ways to apply it to special problem settings provide one of the key research thrusts and account for its growing success in dealing with *NP*-hard problems[4]. The central idea is to allow non-improving moves and deterministic choice of a solution from the neighbourhood of the current solution.

Solutions already visited are declared tabu and the search will move to the best non-tabu neighbour. The algorithm is as follows:

Choose an initial solution s
REPEAT
Move to the best s´ from the neighbourhood of s that is not tabu;
Update the tabu list of solutions;
UNTIL *stopping criteria*

The computational time for the algorithm to consistently check the tabu list is kept at a minimum by ensuring that only the last visited solutions are stored in the list. This is still a memory-hungry strategy where only the properties of the solutions visited are stored in the list, hence all solutions with such properties are declared tabu. In a “regular” tabu search, one must also evaluate the objective for every element of the neighborhood *N(S)* of the current solution. This can prove extremely expensive from a computational standpoint. An alternative is to instead consider only a random sample *N'(S)* of *N(S)*, thus reducing significantly the computational burden.

However, it can happen that properties in the tabu list may prevent solutions that have not been visited. To counter this effect, aspiration criteria are introduced that may overrule the tabu state of a solution. For example, a simple aspiration criterion may be a solution with a better objective value than the best found so far.

Future Forests Research Ltd, PO Box 1127, Rotorua. Ph: 07 921 1883 Email: info@ffr.co.nz Web: www.ffr.co.nz

DIVERSIFIED SPECIES TECHNICAL NOTE

Number: DSTN-018
Date: July 2011

Simulated Annealing

Simulated annealing is a generalisation of a Monte Carlo method for examining the equations of state and frozen states of *n*-body systems[5]. The concept is based on the manner in which liquids freeze or metals re-crystallise in the process of annealing.

In an annealing process, a melted substance, initially at high temperature and disordered, is slowly cooled so that the system at any time is approximately in thermodynamic equilibrium. As cooling proceeds, the system becomes more ordered and approaches a "frozen" ground state at *T=0*. Hence the process can be thought of as an adiabatic approach to the lowest energy state. The essence of the process is slow cooling, allowing ample time for re-distribution of the particles of an *n*-body system, as mobility is lost.

The Boltzmann probability distribution is used to express the idea that a system in thermal equilibrium at temperature *T* has its energy probabilistically distributed among all different energy states *E*, i.e.

Probability(E) = exponential(-E/kT)

The quantity *k* (Boltzmann's constant) is a constant of nature that relates temperature to energy. The system sometimes goes uphill as well as downhill, with less likelihood of an uphill excursion at lower temperatures.

These principles are coded in numerical calculations, and thus a simulated thermodynamic system is assumed to have an initial state of energy *E* and temperature *T*. Holding *T* constant the initial state is perturbed and the change in energy *dE* is computed. If the change in energy is negative the new state is accepted. If the change in energy is positive it is accepted with a probability given by the Boltzmann factor:

p = (E1 –E2)/kT

as the analogy to statistical mechanics suggests[5]. This general scheme of always taking a downhill step while sometimes taking an uphill excursion is known as the Metropolis algorithm.

To generalize the application of the Metropolis algorithm to apply it beyond thermodynamic systems, the following characteristics are featured[6]:

- A description of possible system states;
- A generator of random changes in the state – these changes are "options" presented to the system;
- An objective function E (analog of energy) whose minimisation is the goal of the procedure; and
- A control parameter T (analog of temperature) and an "annealing schedule" that determines the trend of lowering T. For example, after how many random changes in the state is each downward step in T taken, and how large is the step. The assignment of initial T and the annealing schedule may require insight into the problem, trial-and-error, and experience.

How do these characteristics fit into a combinatorial optimisation problem? The current state of the thermodynamic system is analogous to the current solution to the combinatorial problem, the energy equation for the thermodynamic system is analogous to the objective function, and ground state is analogous to the global minimum. The major difficulty (the art) in implementing the algorithm is that there is no obvious analogy for the temperature T with respect to a parameter in the combinatorial problem.

Genetic Algorithms

This approach emulates biological adaptation in a population where each organism has a different set of genes. These genes correspond to the parameters of the model to be optimised, and so each organism represents a potential solution to the optimisation problem. For each organism we can determine how well its parameters solve the problem in hand: this

Future Forests Research Ltd, PO Box 1127, Rotorua. Ph: 07 921 1883 Email: info@ffr.co.nz Web: www.ffr.co.nz

DIVERSIFIED SPECIES TECHNICAL NOTE

Number: DSTN-018
Date: July 2011

determines the "fitness" of each organism, with better (lower error) solutions corresponding to higher fitness. The idea behind a genetic algorithm is then to produce new members of the population from members of the current population using various *genetic operators*.

The process of *recombination* (or *crossover*) is the equivalent of two parents interchanging genes to create new offspring. The operation of *mutation* is a means by which the offspring receives randomly changed genes from one parent only. Therefore, new generations are created iteratively, and at each stage the fitness of each member is a measure of how well each solves the optimisation problem. The goal of this approach is to produce diversity within the population and so explore various gene combinations or solutions to the optimisation problem[7].

However, this is not a blind search: the probability of reproduction is related to the fitness of the parents, i.e. fitter parents are generally made to produce more children. Nevertheless, not only the fittest individuals produce children: although that may seem the best strategy, it is similar to permitting only downhill moves in the search for the global minimum. Such a strategy is prone to getting stuck in local minima. Therefore, while less fit members may be produced in the interim, the goal is that, over many generations, the fitness landscape is comprehensively explored, and that the population adapts towards a fitter state.

Here is the general search process for a genetic algorithm[8]:

- Create a random initial state: An initial population is created from a random selection of solutions (which are analogous to chromosomes);
- Evaluate fitness: A value for fitness is assigned to each solution (chromosome) depending on how close it is to solving the problem (satisfying objective and constraint criteria). These "solutions" are not to be confused with "answers" to the problem; think of them as possible characteristics that the system would employ in order to reach the answer;
- Reproduce (and children mutate): Those chromosomes with a higher fitness value are more likely to reproduce offspring (which can mutate after reproduction). The offspring is a product of the father and mother, whose composition consists of a combination of genes from them, a process is known as "crossing-over"; and
- Next generation: If the new generation contains a solution that produces an output that is close enough or equal to the desired answer then the problem has been solved. If this is not the case, then the new generation will go through the same process as their parents did. This will continue until a solution is reached.

Experience and Update in Application

The Metropolis algorithm was used to solve a multi-objective land use and management problem for 1500-ha farm with 315 paddocks in the Rotorua District. The Wharenui farm had plantation forestry, dairy, sheep and beef farming. A multi-objective optimisation was used to analyse a 50-year period (from 2006-2056) to determine land use change driven by the need to reduce nitrogen leaching, phosphorus loss and sedimentation, without compromising return on investment. Each paddock was theoretically eligible for land use change in any year, though there was a cost to such a change, and there were multiple versions of each of the land uses, e.g. dairying using feed-pads.

Clearly the optimisation problem was large because of the number of continuous and discrete variables. *A priori* knowledge from the client helped to focus the search for optimal solutions by ranking preferred management alternatives and land uses for each paddock that were most likely to generate maximum return.

The result of this optimisation showed an increase in area of 59% to plantation forestry, 134% to beef cattle, 1.4% to sheep and elimination of dairy farming, with positive environmental outcomes. The returns were still

Future Forests Research Ltd, PO Box 1127, Rotorua. Ph: 07 921 1883 Email: info@ffr.co.nz Web: www.ffr.co.nz

DIVERSIFIED SPECIES TECHNICAL NOTE

Number: DSTN-018
Date: July 2011

maximized under such a scenario. The environmental outcomes included a 30% reduction in nitrogen leaching, 16% decrease in phosphorus loss and 9% in sedimentation.

Given that the formulated problem had 14 objectives, it was not expected to have only one solution. Instead, based on the author's previous experience, a suite of efficient solutions (i.e. the Pareto optimal solutions) were expected that a decision-maker could choose from. The Metropolis algorithm, however, is not suited to that kind of analysis as it would require many runs of the model with subtle changes in the constraints in order to determine the Pareto set. Genetic algorithms being a population search algorithm would be more suited to determining the Pareto set.

There is work going on (2010) to formulate the same problem using genetic algorithms. Currently the model is running without spatial constraints, which are still being formulated. Parallelisation of the code is also being investigated in order to shorten computation time, which has increased substantially.

The possibility of finding a Pareto set will no doubt enhance an understanding of the solution space, far better than the single solution derived from the original Metropolis formulation. Determining a single solution from the Pareto set will then depend on the preferences of the decision maker.

Conclusion

The number of complexities that need to be considered when modelling land use problems has a fundamental effect on the approach to solving them. The problems are computationally difficult or *NP*-hard, and heuristics may be the only options when other methods would be mathematically intractable.

An overview of the heuristic search approach was given. Four variants of hill-climbing were described: tabu search, Metropolis algorithm, simulated annealing and genetic algorithms. One of them, the Metropolis algorithm, was applied to a 315 paddock farm. While this provided useful results, a second approach based on genetic algorithms is being investigated in order to increase the knowledge of the solution space, i.e. to determine the Pareto set of optimal solutions. This will provide even more options for the decision maker.

Comparing these two heuristics to determine, for example, which is the 'best' approach for land use problems remains problematic. Different heuristics are appropriate for different numbers of objectives, data and variables, to the presence of spatial constraints, and the skill and experience of the model developer. This, combined with different search strategies potentially finding different solutions, means that there is no one heuristic that will work best for all land use problems.

References

1. Mills P., Monitors for GLS and other meta-heuristics. 2000. Constraint satisfaction Group, Department of Computer Science, University of Essex.
2. De Jong, K. A. and W. M. Spears, Using genetic algorithms to solve NP-complete problems. In: Third International Conference on Genetic Algorithms, George Mason University, Fairfax, VA., 1989. p. 124-132.
3. Wright, A. W., Traditional and hill-climbing methods. 2002. (ed. L. notes). http://www.cs.umt.edu/CS/COURSES/CS555/lect9_18.pdf.
4. Glover, F., A user's guide to tabu search. Annals of Operations Research, 1993. 41: 3-28.
5. Metropolis N., Rosenbluth A., Teller A. and Teller E., Equation of state calculations by fast computing machines. Journal of Chemical Physics, 1953, 21: 1087-1021.
6. Press W. H., Teukolsky S. A., Vetterling W. T. and Flannery B. P. , Numerical Recipes in C: The Art of Scientific Computing. 1992. Cambridge University Press, New York, NY.

Future Forests Research Ltd, PO Box 1127, Rotorua. Ph: 07 921 1883 Email: info@ffr.co.nz Web: www.ffr.co.nz

7. Bailer-Jones D. M. and Bailer-Jones C. A., Modelling data: Analogies in neural networks, simulated annealing and genetic algorithms. In: Model-Based Reasoning: Science, Technology, Values (eds. L. Magnani & N. Nersessian). 2002. Kluwer Academic/Plenum Publishers, New York.
8. Whitley D., A Genetic Algorithm Tutorial. Publisher, Colorado State University Technical Rpt CS-93-103, 40, 1993.

Future Forests Research Ltd, PO Box 1127, Rotorua. Ph: 07 921 1883 Email: info@ffr.co.nz Web: www.ffr.co.nz