Back to the top, to the tippy tippy top.

Heuristic Algorithms for Global Optimisation

My research interests include using heuristic algorithms for *global optimisation*, especially in *combinatorial optimisation* problems. The prototypical example of combinatorial optimisation is the *Traveling Salesman Problem,* in which the best (i.e. the shortest) tour that visits each of *N* cities exactly once is to be found, starting and finishing the tour at the same city. Such problems are deceptively difficult and algorithms to solve combinatorial problems exactly, i.e. that find *the* best solution, often take an unacceptably long time. Thus non-exact or *heuristic* algorithms are used to produce 'good' quality solutions to a problem in a 'reasonable' amount of time.

When selecting an heuristic algorithm for a specific problem there is a general decision to be made, the algorithm can be either *problem specific* or *generally applicable.* A problem specific heuristic is very narrow in its application since it exploits information about the problem at hand to ease its search. A generaly applicable algorithm is much more general in its application and can be applied, with only very minor changes, to a wide spectrum of problem types. Since a generally applicable algorithm necessarily uses very little information about the structure of a problem, they tend to be slower than problem specific algorithms (potentially orders of magnitude slower), however the quality of solution is often comparable.

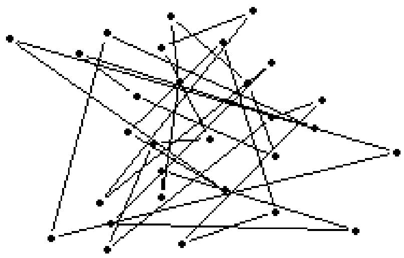
Why then would we use a generally applicable heuristic when they often produce solutions of inferior quality in a much longer time than a algorithm tailored to the problem? It depends on the reason for which the algorithm is being used and is also a matter of personal philosophy. If you are happy writing an algorithm from scratch every time a different problem presents itself, then tailored algorithms are the path to follow. However if you prefer to use the same technique time and again for a huge variety of problems, involving only minor changes in your algorithm, then generally applicable algorithms are the solution. Also, if the problem under consideration has no known solution method, then designing a tailored algorithm will involve constructing, from scratch, a method to solve (i.e. find 'good' solutions to) the problem. Most generally applicable algorithms, on the other hand, require only a method of determining the quality of a solution and sometimes also a method to move between solutions, both of which are relatively simple tasks for the majority of problem types.

So it is for the above reasons that I am investigating heuristic techniques used in global optimisation that can be applied to a wide variety of problem types. Specific algorithm paradigms I am studying are:

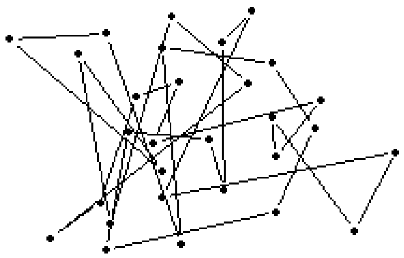
- Local Search
- Simulated Annealing
- Genetic Algorithms
- Threshold Accepting Methods (i.e. the Great Deluge Algorithm *et al.*)
- Tabu Search
- Artificial Neural Nets

Several of the above methods are discussed an 150 page reading paper Report.ps.gz (0.3 Mb).

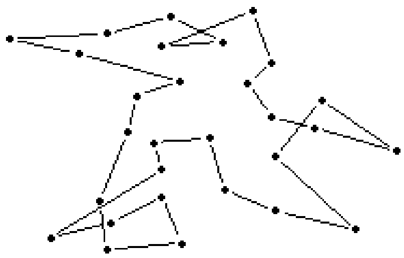
Below is an example of how the simulated annealing (SA) algorithm progresses towards a good (but probably not optimal) solution to a 29 city instance of the Traveling Salesman Problem. The SA algorithm is regulated by a control parameter called *temperature* due to an analogy with the physical process of annealing where a solid is heated to a high temperature and then cooled slowly and carefully to produce a solid with a minimum energy configuration. In the SA algorithm as the temperature decreases the order in the system increases. At high temperatures the system is chaotic and as the temperature decreases the order in the system becomes more apparent.



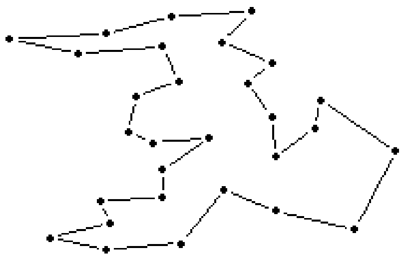
Temperature : 2.23503e+100



Temperature : 18672.1



Temperature : 166.634



Temperature : 14.2068

This page is under development. *Last updated 17 January 1995*

Grant Telfar
grant@isor.vuw.ac.nz