Automated schematic map production using simulated annealing and gradient descent approaches

Suchith Anand[1], Silvania Avelar[2], J. Mark Ware[3], Mike Jackson[1]

[1]Centre for Geospatial Science, University of Nottingham, UK
[2]Technology and Society Lab, Institute for Materials Testing and Research (Empa), Switzerland
[3]Faculty of Advanced Technology, University of Glamorgan, Wales, UK
Tel: +44(0)115 846 8411
Email: suchith.anand@nottingham.ac.uk ; URL: http://www.nottingham.ac.uk/cgs/

1. Introduction

Generating schematic maps are an effective means of generalization of large scale network datasets. The aim is to enhance visualization at line networks and also make them user friendly for interpretation. The basic steps for generating schematic maps are to eliminate all features and networks (or portions of networks) that are not functionally relevant to the network system chosen for mapping. All geometric invariants of the network's structure are relaxed except topological accuracy. Routes and junctions are represented diagrammatically.

The schematization process was initially refined by Elroi (1988) as three main graphic manipulations. First, lines are simplified to their most elementary shapes. Next, lines are re-oriented to conform to a regular grid, such that they all run horizontally, vertically or at a forty-five degree diagonal. Third, congested areas are increased in scale at the expense of scale in areas of lesser node density. Topological errors can occur in the final network, but this was not treated. Implementation details as well as results were not given.

Steps one and two are the key components of the schematization process and their automation has been the focus of previous work by several researchers. The step of line simplification can be achieved using an algorithm such as that of Douglas and Peucker (1973). Care must be taken when performing simplification to avoid the introduction of topological errors. This can be achieved most easily by making use of topology preserving variants of the Douglas-Peucker algorithm, for example that presented by Saalfeld (1999), or other simplification approaches.

Avelar and Muller (2000) present an algorithm for the automatic generation of schematic maps from vector-based information of road networks. They make use of gradient-descent based optimization in an attempt to force the network to conform to specified constraints, such as orientation and minimum separating distance. Map modifications are achieved by the iterative displacement of network vertices. At each iteration vertex displacements are calculated and, provided topological consistency is maintained, are applied.

This paper introduces the concepts of schematic maps generation and looks into two optimisation approaches used for the automated generation of schematic maps: simulated annealing and gradient descent. The details of the experiments carried out using the two approaches on a test dataset and a comparison study of the performance on the dataset is presented.

2. Constraints for automated schematic map production

The schematic map production presented here considers five primary constraints (Anand et al 2006, Avelar 2002):

- Topological: The original network and derived schematic map must be topologically consistent;
- Orientation: If possible, network edges should lie in a horizontal, vertical or diagonal direction;
- Length: If possible, all network edges should have length greater than or equal to some minimum length (to ensure clarity);
- Clearance: If possible, the distance between disjoint features should be greater than or equal to some minimum distance (to ensure clarity);
- Angle: If possible, the angle between a pair of connected edges should be greater than or equal to some minimum angle (to ensure clarity).

Two secondary constraints are included in the simulated annealing approach (Anand 2006). Their purpose is to minimize unnecessary changes to the input network that are likely to occur due to the random nature of simulated annealing.

- Rotation: An edge's orientation should remain as close to its starting orientation as possible;
- Displacement: Vertices should remain as close to their starting positions as possible.

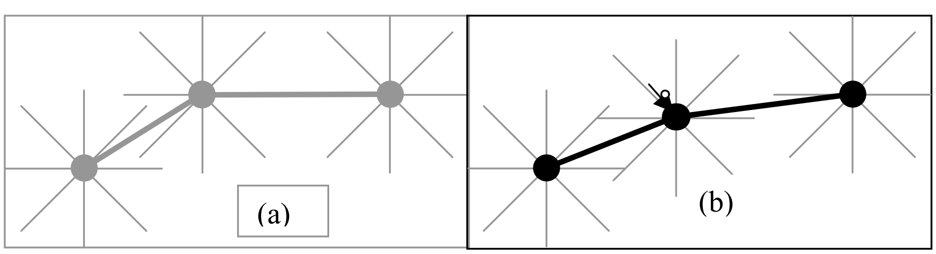
Each of these constraints can be evaluated using straightforward computational geometry functions, e.g. edge/edge intersection test and vertex to edge distance calculation. In order to work efficiently, certain of these functions require the use of a spatial index to avoid sequential scanning of the whole workspace. A simple regular two-dimensional indexing scheme was used in the implementation of the two optimization approaches.

3. A simulated annealing approach to generate schematic maps

The simulated annealing (SA) based schematization algorithm used in this work is similar to that used by Agrawala and Stolte (2001) to render easy-to-read non-schematic route maps. At the start of the optimization process SA is presented with an initial approximate solution (or state). The simulated annealing based algorithm is given below.

In the case of the schematic map production, the input is the initial network: line features made up of edges, which in turn are made up of vertices. The initial state is evaluated using a cost function C; this function assigns to the input state a score that reflects how well it measures up against a set of given constraints. If the initial cost is greater than some user defined threshold (i.e. the constraints are not met adequately) then the algorithm steps into its optimisation phase. This part of the process is iterative. At each iteration the current state (i.e. the current network) is modified to make a new, alternative approximate solution. The current and new states are said to be neighbours. In simulated annealing algorithms the neighbours of any given state are generated usually in an application-specific way. In the algorithm presented here, a new state is generated by the function RandomSuccessor, which works by selecting a vertex at random in the current state and subjecting it to a small random displacement, subject to some maximum displacement distance (Figure 1). This compares to the random displacement methods favoured by Agrawala and Stolte (2001) and is in keeping with the random approach inherent to most simulated annealing based solutions. The new state is also evaluated using C**.** A decision is then taken as to whether to switch to the new state or to stick with the current. Essentially, an improved new state is always chosen, whereas a poorer new state is rejected with some probability p, with p increasing over time. The iterative process continues until stopping criteria are met (i.e. a suitably good solution is found or a certain amount of time has passed or a certain number of iterations have taken place without improvement).

```
Procedure SA_SchematicMap(Initial, Annealing_Schedule, Stop_Conditions)
input: initial state, annealing schedule, stop conditions
output: Cost_current
begin
Current←Initial
t←GetInitialTemperature(Annealing_Schedule)
Cost_current=C(Current)
while NotMet(Stop_Conditions) do
    New←RandomSuccessor(Current)
    Cost_new=C(New)
    ΔE←Cost_current-Cost_new
    if ΔE >0 then
        Current←New
        Cost_current=Cost_new
    else
        p = e^(-ΔE/t)
        r = Random(0,1)
        if (r<p) then
            Current←New
            Cost_current=Cost_new
        end if
    end if
    t←UpdateTemperature(t, Annealing_Schedule)
end while
end
```



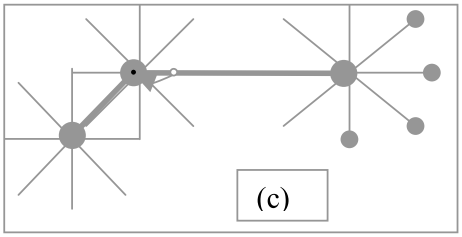


Figure 1: Two new states generated using random displacement of vertex. (a) Current state; (b) Displacement of randomly chosen vertex resulting in poorer solution due to orientation constraint being violated; (c) Displacement of randomly chosen vertex resulting in better solution.

As with other simulated annealing solutions, at each iteration the probability p is dependant on two variables: ΔE (the difference in cost between the current and new states) and t (the current temperature). p is defined as:

$$p = e^{-\Delta E / T}.$$

The variable t is assigned a relatively high initial value; its value is decreased in stages throughout the running of the algorithm. At high values of t higher cost new states (large negative ΔE) will have a relatively high chance of being retained, whereas at low values of t higher cost new states will tend to be rejected. The acceptance of some higher cost new states is permitted so as to allow escape from locally optimal solutions. In practice, the probability p is tested against a random number r ($0 \leq r \leq 1$). If $r < p$ then the new state is accepted. For example, if $p = 1/3$, then it would be expected that, on average, every third higher cost new state is accepted. The initial value of t and the rate by which it decreases is governed by what is called the annealing schedule. Generally, the higher the initial value of t and the slower the rate of change, the better the result (in cost reduction terms); however, the processing overheads associated with the algorithm will increase as the rate of change in t becomes more gradual.

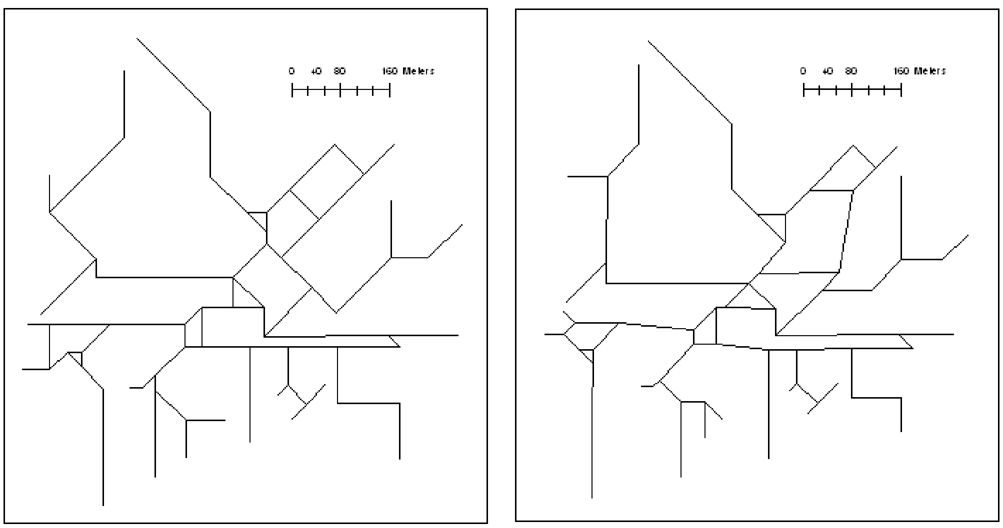
The viability of any SA algorithm depends heavily on it having an efficient cost function, the purpose of which is to determine for any given element of the search space a value that represents the relative quality of that element. The cost function used here, C, is called repeatedly and works by assessing the extent to which a given state meets the set of constraints of the map.

When invoked initially, C evaluates a cost for each vertex in the network. This cost represents the extent to which each vertex meets the set of constraints. The overall cost is found by summing the individual vertex costs. A record of the individual vertex costs is maintained for future reference, meaning that, in any further call, C has to consider only vertices with costs affected by the most recent vertex displacement (Ware et al, 2006).

4. Comparison with gradient descent approach

A gradient descent version of the schematic software was implemented, in order to gain understanding of how the simulated annealing application compares to a gradient descent (GD) based optimization. The gradient descent algorithm differs mainly in the way a new solution is selected for comparison with the current candidate solution. It attempts to proceed toward an optimal solution by finding a sequence of solutions, each of which is better than the previous one. This involved modifying the code in the simulated annealing version to ensure that all negative moves are rejected. If a neighbourhood search does not find any new state better than the current one, GD becomes stuck in the local minima.

The two implementations were compared by generating schematic maps from a road network for 100 seconds. The input parameters were the same for both techniques. The experiment was repeated 10 times for the same dataset and results averaged to take account of randomness. In each of the 10 experiments for the example dataset simulated annealing produced better results than gradient descent.



(a) (b)

Figure 2: Comparison between the highest costs simulated annealing result and the lowest cost gradient descent result to show the variation. (a) SA final cost = 115. (b) GD final cost = 132. (Oscar dataset, Ordnance Survey, UK)

Starting with an initial schematic map cost of 764, the summary of simulated annealing final costs were: 112 (average), 109 (lowest), and 115 (highest). Gradient descent costs were: 138 (average), 132 (lowest), and 147 (highest). A sample output of both techniques is shown in Figure 2. The superior performance of simulated annealing can be explained by its occasional acceptance of negative moves and hence ability to escape local minima. Notice that some schematic edges in the gradient descent network for the stopping time are not yet oriented correctly (Figure 2.b); this is most likely due to the fact that neighboring edges all have low cost and so constituent vertices have very little incentive to move. It could also happen that oscillation between a current state and a new one occurs, such that the network is not any more improved or takes a long time to improve, looking like local minima.

5. Conclusions

This paper has introduced two possible ways of producing schematic maps, the simulated annealing and gradient descent approaches respectively, and shows the comparison of results generated by both approaches. Both simulated annealing and gradient descent based methods have shown good results. In both approaches a stopping criterion enables the system to decide when to stop the iterations, even though it has not reached the perfect answer and can in fact never know when and if it actually will.

Gradient descent can only provide locally optimal solutions and these solutions depend on the starting state. This can be considered restrictive in some situations; however in the production of schematic maps it is advantageous, because a given initial state will always produce the same result. Simulated annealing can avoid suboptimal results, because it will not get stuck every time a locally optimal solution is encountered, but it can produce very different results each time it is run.

Further work is still necessary to find data characteristics and costs to decide for the specific optimization strategy to be applied. We expect to repeat the experiment for various datasets and observe outputs and errors for small and large datasets. We should also investigate if the input network can be preprocessed or additional modifications made to the gradient descent so that it can keep moving towards the required criteria.

6. Acknowledgements

The authors express thanks for the Ordnance Survey, UK for the data used in this work.

7. References

AGRAWALA M. and STOLTE C. 2001 Rendering effective route maps: improving usability through generalization. In Proceedings of *SIGGRAPH 2001*. Los Angeles: 241-250.

ANAND S. 2006 Automatic derivation of schematic maps from large scale digital geographic datasets for MobileGIS, PhD Thesis, University of Glamorgan.

ANAND, S. WARE, J M. and TAYLOR, G.E., 2006 Generalization and schematization of large scale digital geographic datasets for MobileGIS applications, Drummond J., Billen R., Forrest D. and João E. (eds), *Dynamic & Mobile GIS: Investigating Change in Space and Time*, Taylor & Francis publications: pp 161-174.

AVELAR S. and MULLER M. 2000 Generating Topologically Correct Schematic Maps. In Proceedings of 9th *International Symposium on Spatial Data Handling*. Beijing, pp. 1472-1480.

AVELAR S. 2002 *Schematic Maps on Demand: Design, Modeling and Visualization*, PhD Thesis no.14700, Swiss Federal Institute of Technology: Zurich.

DOUGLAS D. H. and PEUCKER, T.K. 1973. Algorithms for the reduction of the number of points required to represent a line or its caricature. *The Canadian Cartographer*, 10(2):112-122.

ELROI D. 1988 Designing a Network Linemap Schematization Software Enhancement Package. In Proceedings of the Eighth Annual *ESRI User Conference*, Redlands, CA.

SAALFELD A. 1999 Topologically consistent line simplification with the Douglas-Peucker algorithm. *Cartography and GIS*, 26(1): 7-18.

WARE J. M., ANAND S., TAYLOR G.E. and THOMAS N. 2006 Automatic Generation of Schematic Maps for Mobile GIS Applications, Special issue on Location Based Services and Mobile GIS, *Transactions in GIS*, Blackwell Publishing.,10(1): pp 25-42.

Biography

Suchith Anand is Research Fellow in Mobile location based services at the Centre for Geospatial Science, University of Nottingham

Silvania Avelar is doing Post Doctoral research at Technology and Society Lab, Swiss Federal Institute for Materials Testing and Research (Empa), Switzerland

J Mark Ware is Reader in the Faculty of Advanced Technology, University of Glamorgan

Mike Jackson is the Director of the Centre for Geospatial Science, University of Nottingham