Ant Colony Optimization Applied to Network Schematization

J. M. Ware[1], N. Richards[2]

[1]University of Glamorgan, Pontypridd CF37 1DL, Wales, UK
Email: jmware@glam.ac.uk

[2] University of Glamorgan, Pontypridd CF37 1DL, Wales, UK
Email: nrichards@bridgend.ac.uk

1. Introduction

This paper details an on-going PhD project that is studying the application of Ant Colony Optimization (ACO) to automated map generalization problems. One objective of the project is to compare ACO with alternative approaches. The paper begins by introducing the general ideas of ACO. It then presents a specific map generalization problem - network schematization - and an ACO solution is described. Initial results are presented and compared with a previous Simulated Annealing solution.

2. Ant Colony Optimization (ACO)

An ant when searching for a food source initially wanders randomly. Upon finding food it returns to its colony while laying down a volatile chemical trail called a pheromone (Goss et al, 1989). Other foraging ants that smell the pheromone trail are more likely to be influenced to follow the path to the food source than continuing to wander randomly. Probabilistically, the stronger the pheromone trail, the more likely it is that an ant will follow the path. Pheromone trails are strengthened by each ant following the path. Initially, there may be many different pheromone trails leading to a single food source. However, over time the strength of the pheromone chemical evaporates reducing its attractiveness to other foraging ants. Thus, as an ant walks, the pheromone it deposits evaporates behind it after a period of time. Longer trails take more time for each ant to walk along and it follows that there is less pheromone density over the trail as a result. Shorter routes to a food source will not take as long to traverse, which helps to maintain a high degree of pheromone density because any evaporation is compensated for by additional pheromone deposits. The more ants that are attracted to the trail the more pheromone density increases. Pheromone evaporation is a vital component to the ant colony to prevent all routes having equal attractiveness. Over a period of time the ants will converge on the shortest path to the food source. This collective ant behaviour of pheromone laying, sensing and following paths to food source was the original inspiration for ACO algorithms. In a simulated system, evaporation prevents an algorithm from converging to a local optimum.

The ACO meta-heuristic (Dorigo et al, 1996) employs a colony of artificial ants that collaborate to find a good solution to a discrete combinatorial optimisation problem. The colony of artificial ants communicates with each other indirectly through the use of artificial pheromone trails. Artificial ants possess some of the characteristics of their real counterparts as well as additional traits that generally suit the optimisation problem at hand.

3. Map Schematization

Perhaps the most well known example of a schematic map is the London Tube map designed by Harry Beck (see http://www.tfl.gov.uk/tfl/maps-home.shtml for this and many hundreds of other examples). The types of schematic maps dealt with in this paper have the following properties:

(i) They are derived from network data sets consisting of polylines, edges and vertices;
(ii) Polylines are simplified to their most elementary shapes;
(iii) They are topologically equivalent to the input network;
(iv) If possible, edges should lie in horizontal, vertical or diagonal direction;
(v) If possible, edges should have length greater than some minimum length (effectively increasing map scale in congested areas).

This paper addresses points (iii)-(v), considering them an optimization problem. Given an input network (pre-simplified using a suitable line generalization algorithm), an alternative state can be obtained by displacing one or more of the network vertices, resulting in re-orientation, shortening and lengthening of edges. The search space being examined is the set of all possible states of the input network. Each state can be evaluated in terms of how closely it resembles a schematic map (i.e. meets a set of constraints based on (iii)-(v)). However, finding the best state by exhaustively generating and evaluating all possible states in not possible, as for any realistic data set the search space will be excessively large. An ACO algorithm for producing schematic maps for network data has therefore been developed.

4. ACO Applied to Map Schematization

4.1 The Algorithm

Each vertex in the network is assigned two matrices– a displacement matrix and a pheromone matrix. The displacement matrix is centred over the original location of the vertex and its cells represent all possible locations into which the vertex can move. Cell size governs the minimum distance a vertex can be displaced. Cell size together with matrix size (the number of cells) determines the maximum distance a vertex can move. Each displacement matrix cell has a corresponding cell in the pheromone matrix. The value of a pheromone cell represents pheromone strength at its corresponding location at any given time; to begin, all pheromone matrix values are initialized to a pre-determined value.

ACO is an iterative process involving a colony of artificial ants working in parallel. Ant colony size is an input parameter to the algorithm (there is no set value). For each iteration each ant starts with the original network (no vertex displacement) and builds its own solution by performing a fixed number (e.g. 1000) of vertex displacements. After each displacement the network is evaluated (against the constraints) and assigned a cost. For each displacement, a vertex is randomly selected and allowed to move from its current matrix location to an adjacent matrix cell. The direction of vertex movement is chosen by a so-called state transition rule, which is influenced by the vertex's associated pheromone matrix (higher values encourage movement) and additional heuristic information (e.g. immediate cost benefit). Within each iteration, a displacement triggers a corresponding reduction of pheromone value (this so-called local update encourages a more complete exploration of the search space). Note that for each vertex all ants are accessing and updating the same pheromone matrix. At the end of each iteration each ant will have produced a solution. The best solution (lowest cost) is used to globally update all pheromone matrices to strengthen pheromone values along the paths from original vertex locations to their new locations in this best solution. This encourages better moves during the iterations that follow. The process repeats until stopping conditions (e.g. maximum number of iterations, maximum time, acceptable cost, etc.) are met.

4.2 Initial Results

The ACO algorithm has been implemented using Java. Initial experiments have been carried out using OSCAR road centre line data for the St. David's area of West Wales. The original test data consisted of 205 edges, made up from a total of 187 vertices. This data is pre-generalized using the ArcGIS Simply Line tool; with point remove and topological error check options selected this makes use of an enhanced version of the Douglas-Peucker algorithm. A weed tolerance value of 50(m) is used in these experiments, resulting in 67 edges made up from a total of 59 vertices. The simplified data (Figure 1) acts as input to the ACO algorithm. The initial cost (in terms of constraints) for the input network is 640.5.

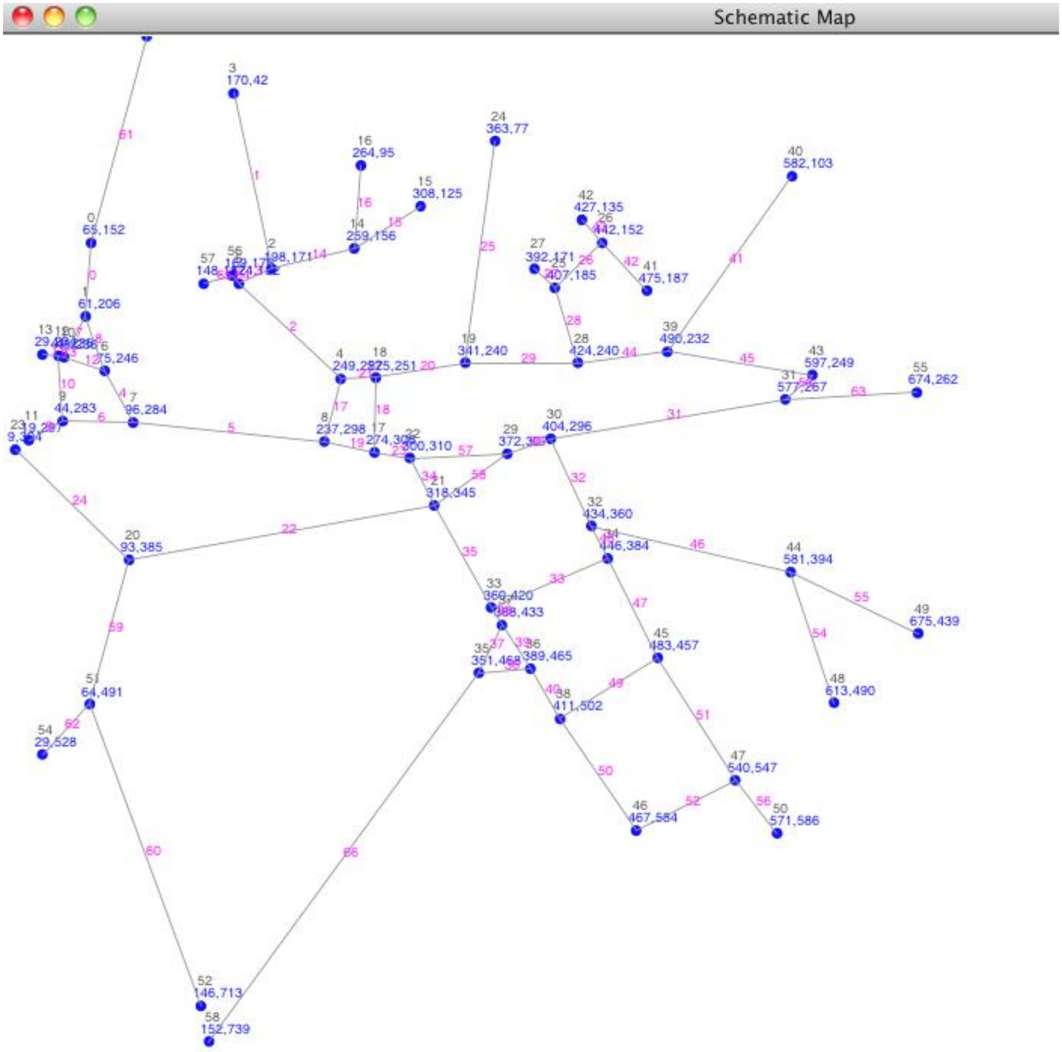

Figure 1 – Input road network. Cost = 640.5.

Figure 2 shows output produced by ACO after a total of 976,000 vertex displacements (in less than 2 seconds). The cost has been reduced to 99.5. For purposes of comparison, a Simulated Annealing (SA) solution (based on Ware et al, 2006) has also been implemented in Java. The best SA solution generated to date (shown in Figure 3) has a cost of 171.1 (after 3,450,000 vertex displacements). It was also noted that after 25,000 vertex displacements, ACO cost had reduced to 179.1 and SA cost had reduced to 333.0.

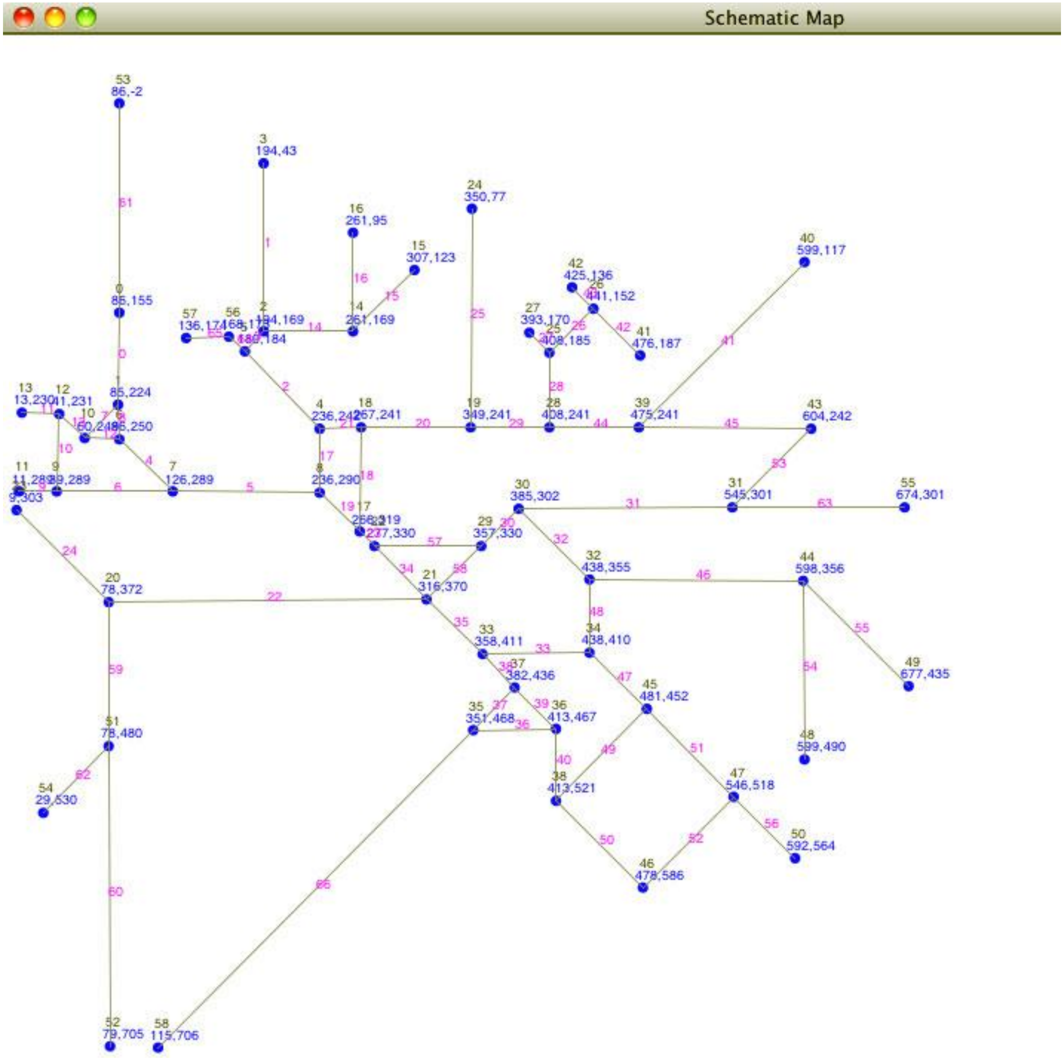

Figure 2 – Schematic map produced by ACO. Cost = 99.5, number of vertex displacements = 976,000.

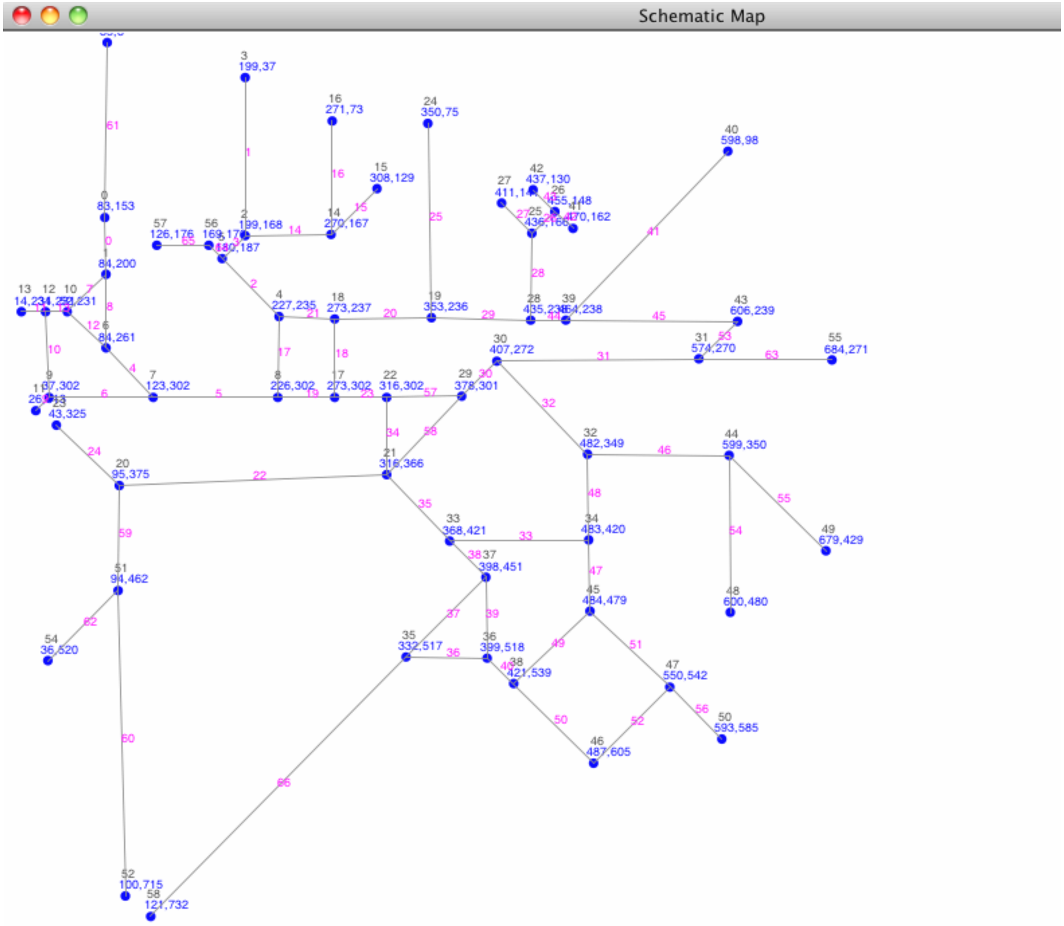

Figure 3 – Schematic map produced by SA. Cost = 171.1, number of vertex displacements = 3,450,000.

5. Conclusion

This paper has presented an ACO solution to the problem of map schematization. It has shown that ACO can be successfully applied to a simple road network. When compared to a SA solution, initial results suggest that ACO is able to reduce overall cost at a faster rate (i.e. few displacements) and produces a better result overall (in evaluation cost terms). It should be noted however that the performance of both approaches is very dependent on a variety of input parameters, and the next phase of research will seek to optimize these parameters to ensure a fair comparison. Future work will concentrate on testing the ACO on more realistic data and carrying out a more rigorous evaluation of constraints and output (involving user testing). ACO will also be applied to other map generalization problems (e.g. Richards et al, 2010).

References

Dorigo M, Maniezzo V and Colorni A, 1996, Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, 26(1):29–41.

Goss S, Aron S, Deneubourg J-L and Pasteels JM,1989, Self-organized shortcuts in the Argentine ant, *Naturwissenschaften*, 76:579–581.

Richards N, Ware JM, Thomas N and Ware JA, 2010, Automated map generalization: application of simulated annealing to river symbolization, *2010 International Conference on Artificial Intelligence* (Las Vegas, July 2010), to appear.

Ware JM, Anand S, Taylor GE and Thomas N, 2006, Automatic Generation of Schematic Maps for Mobile GIS Applications, *Transactions in GIS*, 10(1):25-42.