

TRAVELING SALESPERSON APPROXIMATION ALGORITHM FOR REAL ROAD NETWORKS

Ojaswa Sharma ^{‡*}, Darka Mioc [‡], François Anton [‡] and Girija Dharmaraj [‡]

[‡]Department of Geomatics Engineering, University of Calgary
2500 University Drive NW, Calgary, AB, Canada, T2N 1N4
osharma@ucalgary.ca, mioc@geomatics.ucalgary.ca, gdharmar@ucalgary.ca

[‡]Department of Computer Science, University of Calgary
2500 University Drive NW, Calgary, AB, Canada, T2N 1N4
antonf@cpsc.ucalgary.ca

KEY WORDS: Traveling Salespersons Problem, Christofides algorithm, Shortest path, Transportation network, Optimal tour, GIS

ABSTRACT

Traveling salespersons problem (TSP) is one of the unsolved problems of the day that carry significant value to the transportation networks. The exact solution of a Traveling salespersons problem is not feasible. There are some good approximation algorithms that can provide an approximate solution. Here, we propose a method to extend the conventional Traveling salespersons problem to transportation networks where the travel plan can be optimized within the Geospatial Information System. We also propose an optimized web based implementation scheme that gives a faster response to the route queries. In modern geographic information systems, such a web accessible route query system can be very useful.

1 INTRODUCTION

The Traveling Salesperson Problem (TSP) is to find a minimal weight tour through all the nodes of a weighted graph i.e. to find a cycle in the graph that goes through every node once and only once with the least cost. A direct implication of TSP is in finding an optimal route in road networks. This is very important in transportation networks since it can help reduce fuel consumption, avoid traffic jams, simplify garbage collection, etc. Implementation of TSP in any commercial GIS in real time has not been done yet.

The classical TSP is a NP-hard problem (Mitchell, 1999), (Arora, 1998) and (Arora, 1996). It can be solved using one of the heuristic algorithms called the Christofides (Christofides, 1976) $O(n^{2.5})$ algorithm. The cost of the Christofides solution is at most $\frac{3}{2}$ times the cost of the optimal travelling salesperson tour. This solution is still not applicable to general transportation networks because that TSP algorithm is defined for a complete graph only. Other algorithms have been proposed (Mitchell, 1999), (Arora, 1998) and (Arora, 1996) in order to provide a polynomial time approximation scheme for the travelling salesperson problem as well as for other NP-hard Euclidean geometric problems (minimum Steiner tree, k-TSP, and k-MST). Out of these, the only algorithms applicable to general road networks is Mitchell's algorithm (see (Mitchell, 1999)), but its complexity is higher ($O(n^{20m+5})$) for an equivalent approximation ($1 + \frac{2\sqrt{2}}{m}$) of the optimal tour.

Here, we suggest a novel approach of combining the classical TSP problem solution based on Christofides algorithm (Christofides, 1976) with the transportation networks using Geospatial Information Systems. In the proposed approach, we solve the classical TSP for the city sites (nodes) assuming a complete graph and after an optimal TSP cycle is obtained we replace each successive pair of sites by their shortest paths. This generates a transportation tour which is not the optimal tour but is certainly a factor $\frac{3}{2}$ approximation one.

This algorithm has been tested by implementing it as a web based application that is easily accessible to anyone on the internet. Here, the web-interface is developed as a java applet that processes the information and communicates with the server for the final results. The dataset used here is a part of the city of Calgary downtown road network. The main java applet displays a vector map of the city roads overlaid on an orthophoto of the area. The applet allows the user to choose visiting points in any order and then calculates the most efficient path to visit all the sites and display it.

2 PRELIMINARIES

2.1 Minimum Spanning Tree

The minimum spanning tree of a weighted graph of n nodes is a tree of $(n - 1)$ edges of minimum total weight which spans all the vertices of the graph. To calculate the minimum spanning tree, Kruskal's algorithm (Kruskal, 1956) can be used.

2.2 Minimum Cost Perfect Matching

Given a set of n points (n even), and given edges between these points, with a cost associated to them; a *Perfect Matching* on the graph is a selection of $n/2$ pairwise disjoint edges. The cost of a matching is the sum of the costs of its edges. A *minimum cost perfect matching* is a perfect matching of minimum cost.

2.3 Eulerian walk

A walk that includes each edge of a multigraph exactly once is called an Eulerian walk. A closed Eulerian walk is one that starts and finishes at the same vertex (Downey, 2005).

Theorem 2.1. *A multigraph with no isolated vertices is Eulerian if and only if it is connected and either all or all but two of its vertices have even degree. If every vertex has even degree, then all Eulerian walks are closed; if two vertices have odd degree, then every Eulerian walk starts at one of these vertices and finishes at the other (Downey, 2005).*

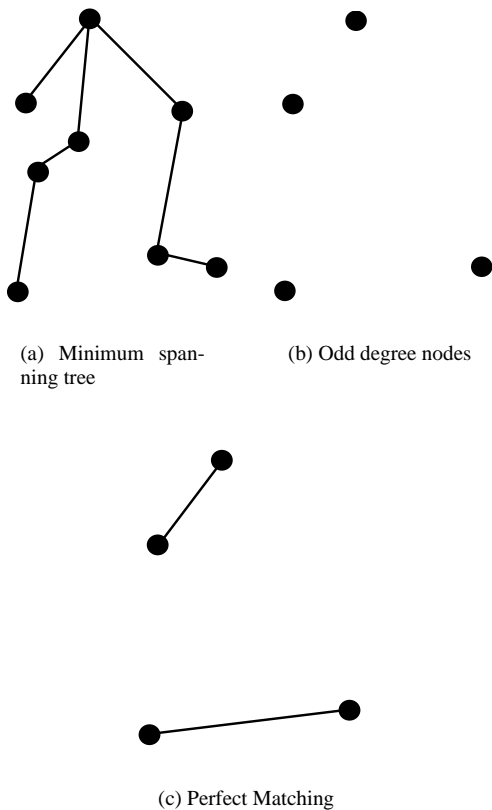


Figure 1: Perfect matching example

2.4 All-Pairs Shortest Path

The shortest distance between any pair of vertices in the shortest-path spanning tree, as long as the path giving the shortest path does not pass through the root of the spanning tree (Skiena, 1990). The problem can be solved using Dijkstra’s algorithm (Dijkstra, 1959) or Floyd’s algorithm (Floyd, 1962). The Floyd’s algorithm also works in the case of a weighted graph where the edges have negative weights (Weissteinf, 2005).

3 THE ROAD NETWORK TSP ALGORITHM

The idea behind modifying the classical Christofides algorithm to solve the Traveling Salesman’s problem arises from the fact that the transportation networks are not complete graphs and hence the algorithms developed for graphs are not applicable to transportation network. Here, we first present the original Christofides algorithm and then explain how it is modified to work on the transportation networks. Given a complete graph G , the Christofides algorithm (Christofides, 1976) is described here:

1. Find the minimum spanning tree (Kruskal, 1956) of the given graph G .
2. Identify all odd-degree nodes in the minimum spanning tree.
3. Do a minimum cost perfect matching on the odd degree nodes in the minimum spanning tree (see figure 1).
4. Construct the multigraph (see upper right graph in figure 2) where the set of vertices is the set of vertices of the minimum spanning tree, and the set of edges is obtained by adding the minimum cost perfect matching edges to the minimum

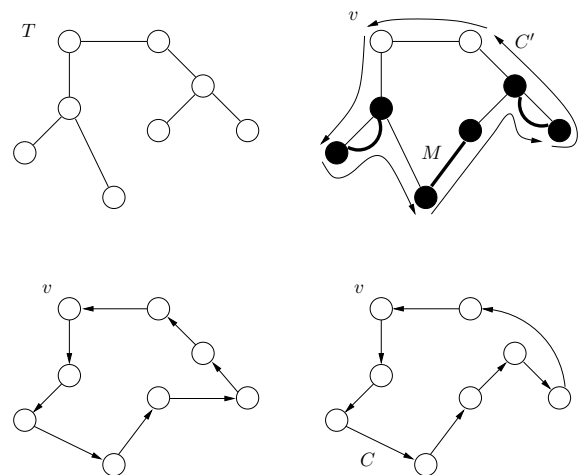


Figure 2: The Christofides Algorithm (From (Fleischer, 2003))

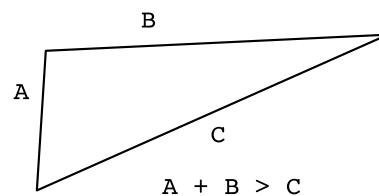


Figure 3: The Triangle Inequality

spanning tree edges (if an edge between the two same vertices appears both in the perfect matching and in the MST, there will be two edges in the multigraph).

5. perform an Eulerian walk on the graph from the previous step (see lower left graph in figure 2).

3.1 Extension of the Christofides Algorithm

A simpler but less accurate algorithm exists to solve the Traveling Salesman’s problem with Triangle-Inequality. This algorithm creates an Euler tour on the minimum spanning tree and shortcuts any node already visited that produces a tour of length at most twice the shortest tour. Here, we propose an approach to extend the 1.5 optimal Christofides algorithm that is also based on triangle inequality. The Christofides algorithm differs from the 2-approximation algorithm in the way that it makes use of the minimum cost perfect matching to reduce the overall length of the tour. This approach divides the process of finding an optimal transportation route into two parts. Firstly, the TSP cycle is calculated using the Christofides algorithm assuming a complete graph with the nodes (road intersections) and then we remove this assumption and convert this to a transportation route. Given a Transportation network T , the algorithm is:

1. Let G be the complete graph having nodes V from graph T .
2. Find the minimum spanning tree of the given graph G .
3. Identify all odd-degree nodes in the minimum spanning tree.
4. Do a minimum cost perfect matching on the odd degree nodes in the minimum spanning tree.
5. Construct the multigraph where the set of vertices is the set of vertices of the minimum spanning tree, and the set of edges is obtained by adding the minimum cost perfect matching edges to the minimum spanning tree edges (if an

edge between the two same vertices appears both in the perfect matching and in the minimum spanning tree, there will be two edges in the multigraph).

6. perform an Eulerian walk on the graph from the previous step.
7. Substitute each pair of successive nodes by the shortest path between them.

The route obtained after this can be considered as an optimal tour because the shortest path is used instead of direct path because the direct path might not exist in transportation networks. Note here that if there is a direct path between two successive nodes of the cycle then the shortest path will be the direct path. This can be proved by the triangle inequality. For the sole purpose of performance improvement, the shortest path is not calculated at runtime, rather it is retrieved from the database that contains all-pairs shortest paths.

3.2 Proof of Correctness

The Christofides algorithm assumes that the graph is a complete graph. Road networks are not generally complete graphs (unless network degenerates to fewer than four vertices). Moreover, the triangular inequality might not hold for some road networks. We will prove that replacing the edges from the complete graph by shortest paths does not increase the approximation rate if the triangular inequality holds. If the triangular inequality does not hold, the approximation factor is 2 (like for the tour based on the minimum spanning tree). In fact, the only thing that changes in Christofides's algorithm proof is that we need to prove that shortcuts always exist in road networks and that they do not increase the length of the tour. The fact the length of the tour is not increased comes from the triangular inequality. It is obvious that we can always choose as a crossover the shortest path along the road network. Now let us prove the approximation

Theorem 3.1. *The above TSP algorithm produces a $\frac{3}{2}$ approximation of the optimal Travelling Salesperson tour of a set V of vertices (crossings) of a road network R that satisfies the triangular inequality.*

Proof. Let T be a minimum spanning tree of V along (i.e. that is a subgraph of) R . Let T^* be a travelling salesperson optimal tour for V along R . Let c^* be the cost of T^* , and c be the cost of T . Removing any edge from T^* gives a spanning tree of V . Thus, c^* is larger than or equal to c . If we connect the odd nodes of T (the only vertices are the odd nodes of T) in the order of their cyclic appearance on T^* , we obtain a cycle of cost at most c^* (because of the triangular inequality). This cycle has an even number of nodes (because the number of odd-degree vertices in a graph is even) and edges (because it is a cycle). If we choose every second edge on the cycle, we partition its edges into two disjoint sets. Each set is a perfect matching of the odd nodes of T . Thus, one of the two sets has cost at most $\frac{c^*}{2}$. Therefore the cost of the minimum cost perfect matching P is at most $\frac{c^*}{2}$. Since $c \leq c^*$, the cost of the multigraph overlay of T and P is at most $\frac{3}{2}c^*$. Finally, by the triangular inequality, introducing shortcuts can only lower the cost. \square

3.3 Analysis

The complexity of the Christofides algorithm is polynomial since the minimum cost perfect matching can be computed in $O(n^{1.5} \log^5 n)$ using Varadarajan's algorithm (see (Varadarajan,

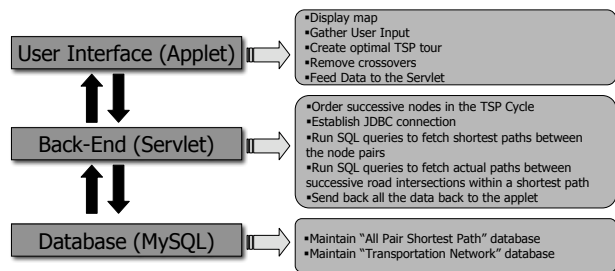


Figure 4: The functionality

1998)), while the Euler tour can be computed in linear time (because the multigraph is Eulerian), the Minimum Spanning Tree is $O(n \log n)$ and the shortcutting of the Euler tour is linear. However, if we take into account the size of the output, an output sensitive algorithm analysis leads to a polynomial $O(n^{1.5} \log^5 n)$ time in the sum of the sizes of the Minimum Spanning Tree and the minimum cost perfect matching. The all shortest path computation complexity is $O(n^{2.575})$ using (Zwick, 2002), but it is a preprocessing step that is performed once for a given road network.

4 TSP IMPLEMENTATION IN SPATIAL DATABASE

In this prototype implementation, the web-interface is developed as a java applet that processes the information and communicates with the server for the final results.

The main java applet displays a vector map of the city roads overlaid on an orthophoto of the area. The applet allows the user to choose visiting points in any order and then calculate the most efficient path to visit all of the sites and display it.

The Interface of the application is fairly simple to use and understand. The backend of this application is a database driven Java enterprise application. The applet gathers the user input (i.e., the sites to visit) and solves the traveling Salesperson's problem assuming a complete graph constituted by the highlighted points. The best cycle is calculated using the Christofides algorithm and the crossovers are removed. The application must somehow calculate the actual road path that it does by replacing the successive site pairs in the best cycle, by their shortest paths. For faster execution and quick response of the application, these shortest paths are not calculated on the fly, rather a database of all-pairs shortest paths" (Zwick, 2002) is created using the Floyd's algorithm (Floyd, 1962). The best cycle obtained after solving the traveling salesperson problem, is sent to the server to get the actual optimal road path.

From a programming standpoint, there are a few optimizations done here. When the user moves the mouse over the applet, the nearest site, to the mouse, is highlighted. To optimize the search for the nearest site among a list of sites, the whole city is divided into parts using Binary space partitioning (QuadTree partition). This is far more efficient than doing a linear search.

The data preparation was a crucial step here. The raw inputs to this application are a vector road map of the city of Calgary in ASCII format, a raster base map and the map projection information. We developed several applications for data processing and input to the spatial database. These applications extract the implicit information about the road network connectivity and store it in spatial database that can be used by the applet to do the two basic tasks: use the information for solving the traveling salesperson's problem and displaying the roads (Please note here that

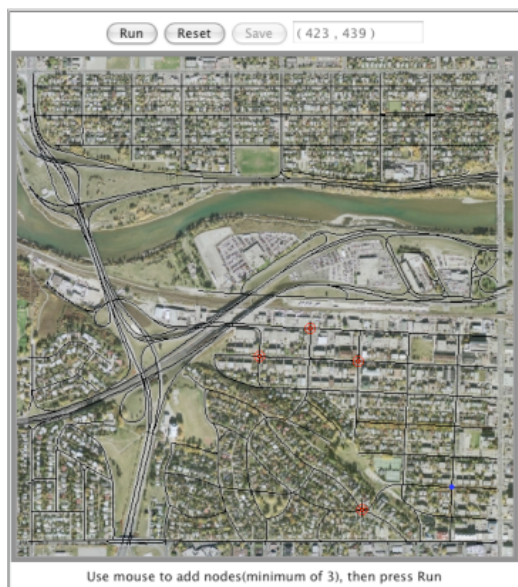


Figure 5: Sites to be visited selected on the map

curved roads can be displayed and processed appropriately).

The spatial database also holds the all pairs shortest path information. This is accessed by the servlet to process shortest path queries.

5 RESULTS AND APPLICATIONS

Figures 5 and 6 are snapshots of the system that was developed using the transportation network of the city of Calgary, Canada. The vector road map is overlaid on the Ortho-photograph of the area. The downtown area of the city has been chosen to test the algorithm as it has parallel roads with many intersections. This helped us to generate some complex cases. These results clearly show the application of the algorithm on the transportation networks. This application can be accessed at:

<http://oaswa.geomatics.ucalgary.ca/tsp>

Having the TSP solved for transportation networks, enables us to optimize many transportation tasks. One such example is supplying the medicaments to the pharmacies. This involves visiting all the pharmacies in question. This job can be done faster and with lesser fuel consumption if an optimal route is followed. Some other potential applications of TSP in transportation context are postal delivery, tourist visits, garbage collection and pizza delivery.

6 CONCLUSIONS

In this article we have presented implementation of Christofides TSP algorithm within a Geospatial database. The implementation of the Christofides algorithm for the travelling salesperson problem is applied on road networks. Up to our knowledge, this is the first actual implementation of a TSP algorithm in the case of transportation networks.

7 ACKNOWLEDGMENT

The authors wish to thank the City of Calgary for the use of their digital data of Calgary's downtown. We also wish to acknowledge Intergraph Inc. for the Registered Research Laboratory (RRL) grant. Thanks also to two of the graduate students,



Figure 6: The optimal route highlighted in white

Matthew Reid and Anoop Pullivelli, at the Department of Geomatics Engineering, University of Calgary for their fruitful discussions and valuable help.

REFERENCES

- Arora, S., 1996. Polynomial time approximation schemes for Euclidean TSP and other geometric problems. In: 37th Annual Symposium on Foundations of Computer Science (Burlington, VT, 1996), IEEE Comput. Soc. Press, Los Alamitos, CA, pp. 2–11.
- Arora, S., 1998. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. ACM* 45(5), pp. 753–782.
- Christofides, N., 1976. Worst case analysis of a new heuristic for the traveling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, (Pittsburgh, PA).
- Dijkstra, E., 1959. A note on two problems in connection with graphs. In: *Numerische Math*, pp. 269–271.
- Downey, R., 2005. Eulerian Graphs. World Wide Web. http://www.mcs.vuw.ac.nz/courses/MATH214/2005T1/notes_2.pdf.
- Fleischer, R., 2003. IT Fundamentals CSIT570. World Wide Web. <http://www.cs.ust.hk/~rudolf/Courses/Msc03/Script/week2.pdf>.
- Floyd, R. W., 1962. Algorithm 97: Shortest path. *Commun. ACM* 5(6), pp. 345.
- Kruskal, J. B., 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. Amer. Math. Soc.* 7, pp. 48–50.
- Mitchell, J. S. B., 1999. Guillotine subdivisions approximate polygonal subdivisions: a simple polynomial-time approximation scheme for geometric TSP, k -MST, and related problems. *SIAM J. Comput.* 28(4), pp. 1298–1309 (electronic).
- Skiena, S., 1990. All pairs shortest paths. In: *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. § 6.1.2, Addison-Wesley, Los Alamitos, CA, pp. 228–229.

Varadarajan, K. R., 1998. A divide-and-conquer algorithm for min-cost perfect matching in the plane. In: FOCS '98: Proceedings of the 39th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, p. 320.

Weisstein, E. W., 2005. All-Pairs Shortest Path, from MathWorld—A Wolfram Web Resource. World Wide Web. <http://mathworld.wolfram.com/All-PairsShortestPath.html>.

Zwick, U., 2002. All pairs shortest paths using bridging sets and rectangular matrix multiplication. J. ACM 49(3), pp. 289–317.