Trends in

Applied Sciences Research

ISSN 1819-3579

Academic
Journals Inc.

www.academicjournals.com

Trends in Applied Sciences Research 2 (4): 348-353, 2007
ISSN 1819-3579

Graph Algorithms and Shortest Path Problems: A Case of Djikstra's Algorithm and the Dual Carriage Ways in Sokoto Metropolis

Aminu A. Ibrahim
Department of Mathematics, Usmanu Danfodiyo University, Sokoto, Nigeria

Abstract: This study deals with the problem of finding shortest paths in traversing some locations within the Sokoto metropolis. In particular, It explores the use of Dijkstra's algorithm in constructing the minimum spanning tree considering the dual carriage ways in the road network of Sokoto metropolis. The results shows a remarkable reduction in the actual distances as compared with the ordinary routing. These results indicate, clearly the importance of this type of algorithms in the optimization of network flows.

Key words: Graph algorithm, djikstra, spanning tree, shortest path, dual carriage, compute, edges, nodes

INTRODUCTION

Current developments in geographic information systems (GIS) technology, ensures that network and transportation analyses within a GIS environment are become a common practice in many application areas. But central problem in network and transportation analyses is the computation of shortest paths between different locations on a network. Sometimes these computation are to be done in real time.

For the sake of illustration, let us have a look at the case of an emergency call requesting fire service to rush to the scene of a fire outbreak; say in a market place or a bank. In practice, the fastest route can only be determined in real time. In some cases the fastest route has to be determined in a few seconds in order to ensure the safety of goods and lives. Moreover, when large real road networks are involved in an application, the determination of shortest paths on a large network can be computationally very intensive.

Graphs and graph algorithms have been used extensively over the years in searching for optimal topologies (Moore, 1959). This technique has also been very useful in the field of interconnection networks (Drager and Fettweis, 2002). In the same vein, Ibrahim (2006) studied the use of Kruskal's algorithm in the determination of traveling sales man routes using a particular manufacturing enterprise.

The main purpose of this study is to investigate the use of Graph algorithms in optimization problems involving routing along road network in Sokoto metropolis. This is justified especially given the changing phase of the town brought about physical developments and increase in the number of dual carriage ways coupled with the surge in the cost of fuel and corresponding increase in the cost of transportation. Thus, unlike the case of traveling salesman (Ibrahim, 2006) where the locations are fixed, the shortest path problem here assumes variations in both locations and the corresponding distances. With this development, a further application of Dijkstra's algorithm has been achieved in the area of transportation network.

For the purpose of clarity, some basic notions are supplied as follows:

Shortest Path Problem

The shortest paths problem involves a weighted (possibly directed) graph described by the set of edges and vertices [e, v] with a given source vertex [s]. The goal is to find the shortest existing path between [s] and any of the other vertices in the graph Each path, therefore, will have the minimum possible sum of its component edges (u, v) and weights (w (u, v)).

Spanning Tree

Is the graph obtained after improvement of the existing graph by removing all the cyclic components in it. The resulting spanning tree is therefore, always an acyclic subgraph of the parent graph.

METHOD OF COMPUTATION

In what follows, the basic method of Dikstra's algorithm is provided as outlined by Ronald (1988). This algorithm was discovered by Dijksra (1959) and independently, by whiting and Hillier(1960). It finds not only a shortest (u_0, v_0)-path, but also all the shortest paths from u_0 to all other vertices of G. The basic procedure is as follows:

Let that S is a proper subset of the vertex set V such that $u_0 \in$ S and let $\bar{S}$ denote V\S. If $P = u_0 ... \bar{u}\bar{v}$ is a shortest path from u_0 to $\bar{S}$ then clearly $\bar{u} \in S$ and the $(u_0\bar{u})$-section of P must be a shortest $(u_0\bar{u})$-path. Therefore $d(u_0, \bar{v}) = d(u_0, \bar{u}) + w(\overline{uv})$ and the from u_0 to $\bar{S}$ is given by the formula

$$d(u_0, \bar{S}) = \min_{u,v \in S} \{ \ d(u_0, u) + w(uv) \ \} \tag{1}$$

The formula (1) is the basis of Dijkstra's algorithm. It begins from the set $S_0 = \{u_0\}$ to construct an increasing sequence S_0, S_1, ...S_{v-1} of subsets of V in such a way that at the end of step i, shortest paths from u_0 to all vertices in S_i are known.

The first step is to determine a vertex nearest to u_0. This is achieved by computing $d(u_0, \bar{S}_0)$. Select a vertex $u_1 \in \bar{S}_0$ so that using (1) we obtain thus:

$$d(u_0, \overline{S_0}) = \min_{u,v \in S} \{ \ d(u_0, u) + w(uv) \ \} = \min_{u,v \in S_0} \{w(u_0 v)\} \tag{2}$$

That is, $d(u_0, \bar{S}_0)$ is easily computed. We now set iteratively that $S_1 = \{u_0, u_1\}$ and let P_1 denote the path u_0, u_1; this is clearly a shortest (u_0, $\bar{u}_1$)-path. In general, if the set $S_k = \{u_0, u_1, ..., u_k\}$ and the corresponding shortest paths P_1, P_2,..., P_k were determined, we compute $d(u_0, \bar{S}_k)$. Also, by (1), $d(u_0,u_{k+1}) = d(u_0,u_j)+w(u_j u_{k+1})$ for some $j \leq k$. We get a shortest (u_0,u_{k+1})-path by adjoining the edge $u_j u_{k+1}$ to the path P_j.

Illustrative Example

Consider the network in Fig. 1.

To compute the shortest path we proceed as follows:

Set $u_0 = \upsilon_0$, $t(u_0) = 0$ others are ∞
$t(\upsilon_1) = \min \{\infty, 1\} = 1$
$t(\upsilon_2) = \min \{2, t(\upsilon_1)+\alpha[u, \upsilon_2]\} = \min [2,3] = 2$
$t(\upsilon_3) = \min \{t(\upsilon_2)+1, 4\} = \min [2+1, 4] = \min [3, 4] = 3$
$t(\upsilon_4) = \min \{\infty, 4\} = 4$
$t(\upsilon s) = \min \{t(\upsilon_4)+3, 6\} = \min [4+3, 6] = \min [7, 6] = 6$
$t(\upsilon_6) = \min \{t(\upsilon_4)+2, 9\} = \min [4+2, 9] = \min [6, 9] = 6$
$t(\upsilon_7) = \min \{t(\upsilon_5)+4, 9\} = \min [6+4, 9] = \min [10, 9] = 9$
We have obtained the following result.

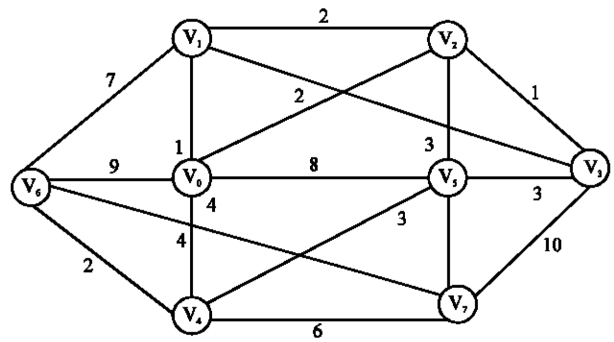

Fig. 1: A connected network

$t(\upsilon_1) = 1$
$t(\upsilon_2) = 2$
$t(\upsilon_3) = 3$
$t(\upsilon_4) = 4$
$t(\upsilon_5) = 6$
$t(\upsilon_6) = 6$
$t(\upsilon_7) = 9$
These are the minimal weights from υ_0 to each υ_i.

APPLICATION

We now consider the way and manner these concepts can be used to establish shortest routes in traversing various locations in the Sokoto metropolis using the dual carriage ways. The Fig. 2 is obtained from map of Sokoto metropolis where the arcs represent the dual carriage ways while the nodes represent the roundabouts.

Using roundabout V_1; connecting Manuri road and Sultan palace as the starting point along the dual carriage ways depicted by the weighted arcs of Fig. 3. The routing layout up to the node V_{10}; connecting Gusau road.

Computation of Shortest Paths Using Fig. 3 with the Nodes Represented by Roundabouts in the Dual Carriage Ways

- Set $u_0 = v_0$, $t(u_0) = o$, other are ∞.
- $t(v_1) = \min\{\infty, 2000\} = 2000$
- $t(v_2) = \min\{\infty, 1500\} = 1500$
- Other are ∞. Thus, $u_1 = v_1$
- $t(v_2) = \min\{1500, t(u_1) + \alpha(u_1, v_2)\} = \min[1500, 3000] = 1,500$
- $t(v_3) = \min\{t(v_2) + 2000, 5500\} = \min[1500 + 2000, 5500] = \min[3500, 5500] = 3500.$
- $t(v_4) = \min\{t(v_3)+1500, 6000\} = \min[3500+1500, 7000] = \min[5000, 7000] = 5000.$
- $t(v_5) = \min\{t(v_4) + 1000, 5000\} = \min[5000 + 1000, 5000] = \min[6000, 5000] = 5000.$
- $t(v_6) = \min\{t(v_5) + 500, 4500\} = \min[5000 + 500, 4500] = \min[5500, 4500] = 4500.$
- $t(v_7) = \min\{t(v_6) + 1000, 3500\} = \min[4500 + 1000, 3500] = \min[5500, 3500] = 3500.$
- $t(v_8) = \min\{t(v_7) + 1000, 3500\} = \min[3500 + 1000, 3500] = \min[4500, 3500] = 3500.$
- $t(v_9) = \min\{t(v_8) + 2000, 5500\} = \min[3500 + 2000, 5500] = \min[5500, 5500] = 5500.$
- $t(v_{10}) = \min[t(v_9) + 500, 6500] = \min[5500 + 500, 6500] = \min[6000, 6500] = 6000.$

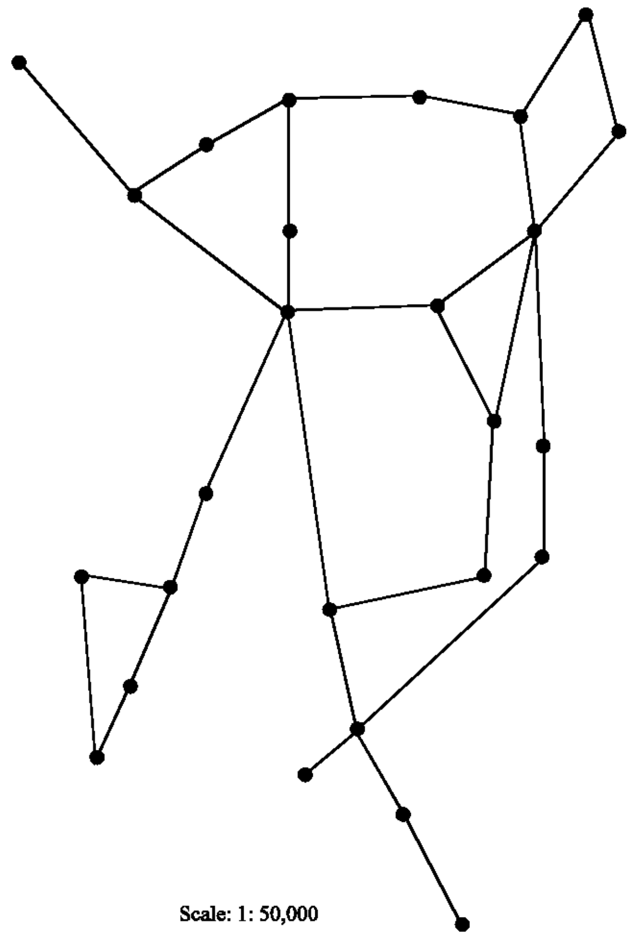

Fig. 2: Road network connectivity of Sokoto metropolis

The algorithm stops and we obtained,

$t(v_1) = 2000_m$	$t(v_6) = 4500_m$
$t(v_2) = 1500_m$	$t(v_7) = 3500_m$
$t(v_3) = 3500_m$	$t(v_8) = 3500_m$
$t(v_4) = 5000_m$	$t(v_9) = 5500_m$
$t(v_5) = 5000_m$	$t(v_{10}) = 6000_m$

These are the minimal weights from u_0 to each v_i. That is, the total distance from Sokoto water works to Federal Government College Sokoto reduces to only 6000 m after applying Dijktras algorithm in the corresponding routing using dual carriage ways of the Sokoto metropolis.

Similarly, if we change the starting point to begin from Sultan Abubakar college Sokoto as V_1 and up to Sokoto Water Works have the following:

Computation of Shortest Paths by Making Appropriate Adjustments in Fig. 3 so that V_1 is Sultan Abubakar College while V_{10} is the Sokoto Water Works

- Set $u_0 = v_0$, $t(u_0) = o$, other are ∞.
- $t(v_1) = \min\{\infty, 1000\} = 1000$

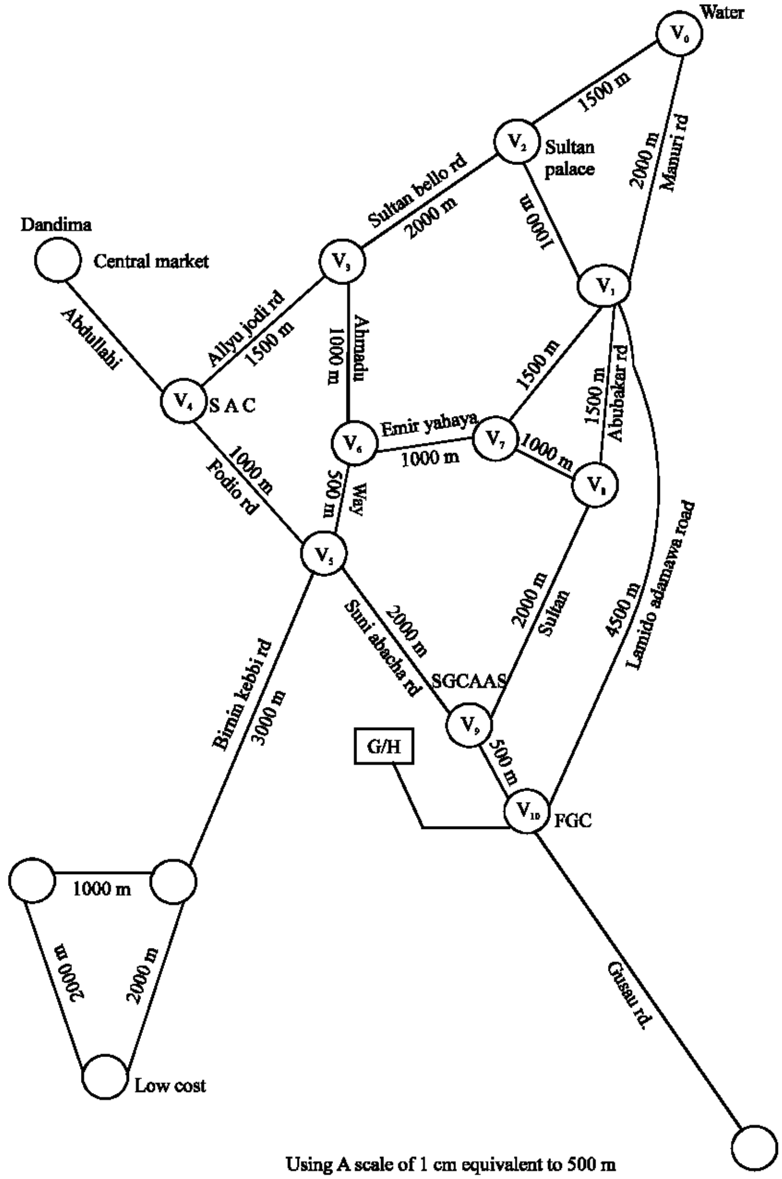

Fig. 3: Road network connectivity of Sokoto metropolis

- $t(v_2) = \min\{\infty, 1500\} = 1500$
- Other are ∞. Thus, $u_1 = v_1$
- $t(v_2) = \min\{1500, t(u_1) + \alpha(u_1, v_2)\} = \min[1500, 2500] = 1{,}500$
- $t(v_3) = \min\{t(v_2) + 1000, 500\} = \min[1500 + 1000, 500] = \min[2500, 500] = 500.$
- $t(v_4) = \min\{t(v_3)+1000, 1500\} = \min[500+1000, 1500] = \min[1500, 1500] = 1500.$
- $t(v_5) = \min\{t(v_4) + 1000, 2500\} = \min[1500 + 1000, 2500] = \min[2500, 2500] = 2500.$
- $t(v_6) = \min\{t(v_5) + 2000, 2000\} = \min[2500 + 2000, 2000] = \min[4500, 2000] = 2000.$
- $t(v_7) = \min\{t(v_6) + 500, 2500\} = \min[2000 + 500, 2500] = \min[2500, 2500] = 2500.$
- $t(v_8) = \min\{t(v_7) + 4500, 3000\} = \min[2500 + 4500, 3000] = \min[7000, 3000] = 3000.$

- $t(v_9)$ = min {$t(v_8)$ + 1000, 4000} = min [3000 + 1000,4000] = min [4000, 4000] = 4000.
- $t(v_{10})$ = min [$t(v_9)$ + 1500, 5000] = min [4000 + 1500, 5000] = min [5500, 5000] = 5000.

The algorithm stops and we obtained,

$t(v_1) = 1000_m$	$t(v_6) = 2000_m$
$t(v_2) = 1500_m$	$t(v_7) = 2500_m$
$t(v_3) = 500_m$	$t(v_8) = 3000_m$
$t(v_4) = 1500_m$	$t(v_9) = 4000_m$
$t(v_5) = 2500_m$	$t(v_{10}) = 5000_m$

These are The minimal weights from u_0 to each v_i

CONCLUSION

It is can be established from the a fore mentioned that Dijstra's Algorithm is a useful graph theoretic mechanism for optimisaton processes of network connectivity. Present results have exposed the versatility of this theoretic tool in carrying out minimization process involving construction and for itinerancy in conveyance of goods and services in different locations of a town based on the existing road network.

REFERENCES

Dijkstra, E.W.A., 1959. Note on two problems in connection with graphs. Numerische Math., 269-271.

Drager, T. and G.P. Fettweis, 2002. Energy savings with appropriate interconnection networks in parallel DSP. 3 Lolloquim des schwerpunklprogramms der dculschen forscungsgescllschaft VIVA: 35-42, chemnitz, Germany.

Ibrahim, A.A., 2006. Graph algorithm and travelling salesman problrm: A case of kruscal's algorithm and some distribution centres of a manufacturing company, abacus. J. Math. Assoc. Nigeria, 33: 237-241.

Moore, E.F., 1957. The shortest path through a maze. Proc. Iternat. Symp. Switching Th., Part II, Harvard Univ. Press, pp: 285-202.

Ronald, G., 1988. Graph theory the Benjamin/Cummings Publishing Company, Inc. California, pp: 332.

Whiting, P.D. and J.A. Hillier, 1960. A Method for finding the shortest route through a road network. Operat. Res. Q., 11: 37-40.