Case Study of Shortest Path Algorithms and Implementation using MATLAB

Yee Yee Htun

Dr. Yee Yee Htun
Ph.D (Applied Maths), Department of Engineering Mathematics
TU (Hmawbi), Hmawbi Township, Yangon Division, Myanmar

Abstract: Shortest path problems are among the most studied network flow optimization problems with interesting application across a range of fields. In this paper, three shortest path algorithms are discussed via Dijkstra’s Algorithm (one to all pairs of nodes), Floyd Warshall’s Algorithm (all to all pairs of nodes) and Linear Programming Problems (LPP). These algorithms are also solved using Matlab software, which gives quick results for larger nodes. By this research, we can successfully study how many ways to find shortest path. Graph technique in Matlab can also be applied to be simply solved the shortest path problems. The application of Direct Graph and Undirect Graph of shortest path was implemented for the route of ferry bus, North Dagon township to TU (Hmawbi).

Keywords: Shortest path, Dijkstra’s Algorithm, Floyd Warshall’s Algorithm, Linear Programming Problems, Matlab software & Direct Graph.

1. INTRODUCTION
Finding the shortest path is an important task in network and transportation related analysis. Shortest distance problems are inevitable in road network applications, such as city emergency handling and driving system, where optimal routing has to be found. Therefore, network optimization has always been the heart of operational research. Also, as traffic conditions of a city change from time to time, there could be a huge amount of request occurring at any moment, for which an optimal path solution has to be found quickly. Hence, efficiency of an algorithm is very important to determine the shortest routes are between nodes in a network.

There are many algorithms that can be used to determine the shortest route between two nodes in a network. In this paper, two standard algorithms Dijkstra’s algorithm and Floyd Warshall’s algorithm are discussed and also solved using Matlab software. The linear programming formulation of shortest route problem solved using (0-1) binary integer programming technique is also discussed.

The dual of formulated linear programming and shortest route problem solved by algebraic method is demonstrated for small number of nodes, as it is difficult to solve for large number of nodes. In such cases, Matlab software can be the best choice. Further, the shortest distance and shortest routedetermined using Complementary Slackness Theorem,(Dr. Roopa, K.M., 2013).
2. GRAPH IN MATLAB

A directed graph with four nodes and three edges is as shown in Picture 1 in Matlab. Graph
theory functions in the toolbox apply basic graph theory algorithms to sparse matrices. A sparse matrix
represents a graph, any nonzero entries in the matrix represent the edges of the graph, and the values of
these entries represent the associated weight (cost, distance, length, or capacity) of the edge. Graph
algorithms that use the weight information will cancel the edge if a NaN or an Inf is found. Graph
algorithms that do not use the weight information will consider the edge if a NaN or an Inf is found,
because these algorithms look only at the connectivity described by the sparse matrix and not at the
values stored in the sparse matrix. (matlabexpo.co.kr, 2016).

![Picture 1. Direct Graph example](image)

Sparse matrices can represent four types of graphs:
1) Directed Graph — Sparse matrix, either double real or logical. Row (column) index indicates
   the source (target) of the edge. Self-loops (values in the diagonal) are allowed, although most of
   the algorithms ignore these values.
2) Undirected Graph — Lower triangle of a sparse matrix, either double real or logical. An
   algorithm expecting an undirected graph ignores values stored in the upper triangle of the sparse
   matrix and values in the diagonal.
3) Direct Acyclic Graph (DAG) — Sparse matrix, double real or logical, with zero values in the
   diagonal. While a zero-valued diagonal is a requirement of a DAG, it does not guarantee a
   DAG. An algorithm expecting a DAG will not test for cycles because this will add unwanted
   complexity.
4) Spanning Tree — Undirected graph with no cycles and with one connected component. There
   are no attributes attached to the graphs; sparse matrices representing all four types of graphs can
   be passed to any graph algorithm. All functions will return an error on nonsquare sparse
   matrices.

2.1 Finding the Shortest Path in a Directed Graph
1) Create and view a directed graph with 6 nodes and 11 edges.
2) Biograph object with 6 nodes and 11 edges.
3) Find the shortest path in the graph from node 1 to node 6.
4) Mark the nodes and edges of the shortest path by coloring them red and increasing the line
   width.
   This example Graph results can be shown as in Picture 2 (a).

2.2 Finding the Shortest Path in an Undirected Graph
1) Create and view an undirected graph with 6 nodes and 11 edges.
2) Biograph object with 6 nodes and 11 edges.
3) Find the shortest path in the graph from node 1 to node 6.
4) Mark the nodes and edges of the shortest path by coloring them red and increasing the line
   width.
This example Graph results can be shown as in Picture 2(b).

![Graph](https://example.com/graph.png)

Picture 2. Direct Graph and Undirect Graph in Matlab for 6 nodes(www.mathwork.com)

### 3. SHORTEST PATH ALGORITHMS

#### 3.1 Dijkstra’s Algorithm

Dijkstra’s algorithm considers two sets:

i) set P, which at any specific point consists of all the nodes that were encountered by the algorithm;

ii) set S, a precedence set, which at any specific point consists of the precedent node for each node in the network. Apart from these sets, the algorithm utilizes the following distance information.

$q_{ij}$, for $i, j=1, 2, 3, 4…n$ and $i≠j$, denote the weight of the directed edge (arc) from vertex $i$ to vertex $j$.

If there is no arc from $i$ to $j$, then $q_{ij}$ is set to be infinity. $t_j$, for $j=1, 2, 3, 4…n$ and $j≠s$ where $s$ is the start index. Also,

$$t_j = q_{ij} \text{ for } j=2,3,4…n$$  \hspace{1cm} (1)

In each iteration, the sets P and S as well as the set of all $t_j$, for $j=1, 2, 3, 4…n$ and $j≠s$, that are output from the previous iteration are taken as inputs.

Initially $P = \{s\}$. $S$ is a set of size $n$ populated with

i) 0 if $t_j = \text{infinity}$, ii) $s$ if $t_j = \text{finite value}$

The steps involved in each iteration for finding the shortest distance are summarized below:

**Step 1:** Identify minimum among the computed $t_j$ values. Let $t_k$ be the minimum. Add $k$ to the set $P$.

**Step 2:** Now $P = \{1, k\}$. For each of the nodes not in $P$ and with finite $q_{kj}$, for $j=1, 2, 3, 4…n$ and $j≠s$, recalculate $t_j$ using the below expression:

$$t_j = \min\{ t_j, t_k + q_{kj} \}$$  \hspace{1cm} (2)

Only if $(t_k + q_{kj}) < t_j$, then update the $j$th entry in $S$ to $k$. Continue the iterations until the end node, $e$, is added to the set $P$.

Similarly, the steps to trace the shortest path between nodes $s$ and $e$, using Dijkstra’s algorithm are given below:

**Step 1:** Take node $e$ as the last node in the shortest path

**Step 2:** Find the $e$th entry in the set $S$, let this be $x$. Add this prefix node $x$ to the partially constructed shortest path.

**Step 3:** Check whether $x$ is equal to $s$. If so, go to Step 4; else go to Step 3.

**Step 4:** The required shortest path from node $s$ to node $e$ is thus constructed.

Picture 3 shows an example of using Dijkstra’s Algorithm.
3.2Floyd Warshall’s Algorithm:

Floyd Warshall’s Algorithm is a graph analysis algorithm to find the shortest route between any two nodes in a network with positive or negative edge weights with no negative cycle. This algorithm uses the dynamic programming technique to solve the shortest path problem between all pairs of nodes (all-to-all) in a directed network. It represents the network as a square matrix with n-rows and n-columns, and at the end of the algorithm each (i,j) of the matrix gives the shortest distance from node i to node j. If there is a direct link between node i to node j, then the value at (i,j) is finite, otherwise it is infinite, i.e., d(i,j) = ∞.

The steps to trace the shortest path between two nodes, say i and j using Floyd-Warshall’s algorithm are given below:
Step 1: Take node j as the last node in the shortest path.
Step 2: Find the value S[i, j] from the precedence matrix Sn, let it be x. Add this Prefix node x to the partially constructed shortest path.
Step 3: Check whether x is equal to i. If so, go to step (4); else, set j = x and go to the step 3.
Step 4: The required shortest path from node i to node j is constructed.
To determine the shortest distance and shortest paths between all pairs of nodes in a transportation network as shown in Picture 4, using Iteration (3): Set k=3. Consider third column and third row of D3 as pivot column and pivot row respectively. Except d (1, 3), all the entries in the pivot column are infinity and also except d (3, 5) and d (3, 7), all the entries in the pivot row are infinity. Further, apply transitivity property to obtain the following results:

(i) Since, d(1,4)=17, d(1,5)=14 and d(1,7)=32. So, d (1, 7) = 32 cannot be improved.

(ii) Set precedence matrix S2 as S (1, 4) = 3, S (1, 5) = 3. The changes are as shown in the matrix D3 and S3.

Continuing in this way, the final matrix in the last iteration where none of the entries in the d (i,j) can be improved by transitivity property, because all the elements in the last row are infinity. Finally, the shortest distance between any two nodes is determined from the matrix D7 as shown in Picture 5.

3.3 Algebraic Method for solving the dual Linear Programming Problem

The dual linear programming problem can also be solved using algebraic method for only small number of variables. However, solving the above dual Linear Programming Problem through algebraic method, by introducing slack variables which gives better result compared to any other software packages. By using the first and final tableaus of algebraic method, the dual problem can also be solved using Matlab software. For example, if the solutions obtained from Matlab software are given below:

\[ y_1 = -10.4979, \quad y_2 = 3.6415, \quad y_3 = -0.4979, \quad y_4 = 6.5021, \quad y_5 = 3.5021, \quad y_6 = 5.5021, \quad y_7 = 11.5021 \]

The value of Z = 22 gives the shortest distance from node 1 to node 7. By considering the solutions that satisfy the above constraints the following routes: 1-3, 3-4, 3-5, 4-7, 5-7, 5-6 and 6-7 are obtained. From these sequence of routes 1-3, 3-4, 4-7, the shortest route 1—3—4—7, which is of distance 22 units from node 1 to node 7 is traced. Similarly, other alternate shortest routes that can be obtained are: 1—3—5—7 and 1—3—5—6—7 respectively.

The shortest route can also be determined using Complementary Slackness Theorem. As the sequence of routes 1-2, 3-2, 2-7, 2-4, 4-6 and 4-5, do not satisfy the constraints in the dual problem, from the Complementary Slackness Theorem it follows that \( x_{12} = x_{32} = x_{24} = x_{27} = x_{45} = x_{46} = 0 \).

Substituting these variable values in the primal problem, the following systems of equations are obtained:

\[ x_{13} = 1; \quad x_{13} - x_{34} - x_{35} = 0; \quad x_{34} + x_{47} = 0; \quad x_{35} - x_{56} - x_{57} = 0; \quad x_{56} - x_{67} = 0; \quad x_{47} + x_{57} + x_{67} = 1 \] (3)

By solving above system of linear equations using Gauss Elimination Method, the system in echelon form becomes:

\[ x_{34} + x_{35} = 1; \quad x_{35} + x_{47} = 1; \quad x_{47} + x_{56} + x_{57} = 1; \quad x_{47} + x_{57} + x_{67} = 1 \] (4)

In the above system of equations, there are 4 equations (r=4) with 6 unknowns (n=6) and two free variables \((x_{35}, x_{56})\). Hence, the possible choices are: \((0,0),(0,1),(1,0),(1,1)\). Each of these possible choices may or may not be the solution points because the dependent variables have the restriction, \( x_{ij} = 0 \) or 1. (Dr. Roopa, K.M., 2013).
4. IMPLEMENTATION OF SHORTEST PATH (BETWEEN NORTH DAGON-MAWATA BUS STOP AND TU (HMAWBI))

The case was assigned as to find shortest path between Mawata Bus Stop of North Dagon township and TU(Hmaewbi). By solving this case, the author can apply low cost and minimum time to arrive TU(Hmaewbi) from her home. There are 9 main Bus Stations as focal points. But there are a lot of Bus Stops Between each focal Bus Stations. Unit are per miles to be considered. The m-script of this case is shown in Picture 6.

Table-1 Nodes Assignment

<table>
<thead>
<tr>
<th>No.</th>
<th>BUS STOPS</th>
<th>NODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ma wa ta</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Bay lie</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Aung migalar highway station</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>8 miles</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Saw bwar gyi kyone</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Htaut kyant</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>Toll gate</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>Hmawbi market</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>TU Hmaewbi</td>
<td>9</td>
</tr>
</tbody>
</table>

Picture 6. M-script of Implementation Case

5. RESULTS

As a result of implementing shortest path in Matlab, author can make effort of cost and time to go to office from home everyday. The Result is to use Bus No (99) between Mawata Bus stop to Sawbwargyi Goan Bus stop, and again to TU(Hmaewbi) with Bus No (37). The minimum cost for a route is 700 MMK. The results from Matlab program are shown in Picture 7 (a) and (b).
6. CONCLUSION AND RECOMMENDATION

This paper has presented the results of implementing or application of shortest path algorithms in real case. Effectiveness of Matlab software can also be proved in this paper. All three algorithms of shortest path finding methods are studied and compared. It is evident that Dijkstra’s algorithm takes a relatively lesser time than Floyd and Binary integer programming in finding shortest route. However, Dijkstra’s algorithm is the better option for identifying the shortest path in larger networks such as railway, water, power distribution and gas pipeline networks. For simplicity, the author mainly applied Direct Graph methods that used Dijkstra’s algorithm in background of Matlab. This research paper carried out mainly case study of three shortest path finding concepts and analysis with real world application.

7. REFERENCES