A Minimum Spanning Tree Approximation to the Routing Problem through ‘K’ Specified Nodes

Santosh Kumar1, Elias Munapo2, Maseka Lesaoana3 and Philimon Nyamugure4

1Department of Mathematics and Statistics, University of Melbourne, Parkville, Australia
2Graduate School of Business Leadership, University of Kwa-Zulu-Natal, Westville Campus, Durban, South Africa
3Department of Statistics and Operations Research, University of Limpopo, Bag XI106, Sovenga 0727, South Africa
4Department of Applied Mathematics, National University of Science and Technology, PO Box AC 939, Ascot, Bulawayo, Zimbabwe

E-mail: 1<skumar@ms.unimelb.edu.au>, 2<munapoe@ukzn.ac.za>, 3<Maseka.Lesaoana@ul.ac.za>, 4<pnyamugure@gmail.com>

KEYWORDS Basic Arc. String. Optimal Path. Shortest Route. Network

ABSTRACT This paper presents a minimum spanning tree approach to determine a route through ‘k’ specified nodes. The path through ‘k’ specified nodes is a difficult problem for which no good solution procedure is known. The proposed method determines the route, which may be either an optimal or a near optimal path.

INTRODUCTION

The routing problem pertains to the search for a shortest route in a network connecting two designated nodes, generally called the origin and the destination. This problem has many applications and many solution approaches are available (Dantzig 1957; Bellman 1956, 1962; Peart et al. 1960; Pollack and Weibenson 1960; Beckwith 1961; Ahuja et al. 1990; Roopa et al. 2013; Mikhail and Alexander 2014; Rais et al. 2014). Kalaba (1960) pointed out that the necessity of considering the path from the origin node to the destination node must visit a set of specified nodes en route before arriving to the destination node. When the set of specified nodes is a null set, the problem reduces to an ordinary shortest route problem. (Pollack and Weibenson 1960; Ahuja et al. 1990; Zhan 1997; Kumar et al. 2013). However, when the set of specified nodes is not a null set, the required route has to pass through the set of specified nodes before arriving at the destination node. Such a route may have loops. Another extreme case of the k specified node problem is when the set of specified nodes contains all nodes, and one is required to return to the origin node after visiting all nodes. The problem reduces to a conventional travelling salesman problem (Bellman 1962; Munapo et al. 2013; Zharfi and Mirzazadeh 2013). Once again many approaches are available in the literature (Nagata and Kobayashi 2013; Andreas et al. 2014).

The requirement for a path to pass through ‘K’ specified nodes arises when one may be interested in either saving a separate trip to the given specified node or attempting to take care of a future eventuality that is likely to arise in that situation. There are many examples. The path through ‘K’ specified nodes is a mathematical simplification of much general situations encountered in all walks of life. Let us consider a few such illustrations from different walks of life:

1. Guided Tours is a multi-billion dollar business all over the world. Tours are planned and programmed meticulously. Pick up any tour company operating in any country, a common feature is that with all major tours, they always have suggestions for a few pre or post tours, providing choice to the tourist to cover those destinations while they are out with their main destination. This is an attractive offer for the tourist and also equally good for the tour company to
create an additional business from the same tourist.

2. National planning can serve as another illustration, for example, a national decision is generally not intended to address the immediate situation faced today but solution is intended to address also the likely situation going to arise tomorrow or in the near future. To be specific, consider a decision to add an extra lane to an existing road to address traffic congestion, but this additional lane is not just intended to meet the current traffic requirements but the number of additional lanes should also address future likely traffic requirements on the same sector.

Government subsidies to various industries are directly or indirectly addressing indirect long-term impact of an industry on the nation. They all are dealing with the ‘K’ specified node situation on a broad scale. Some specified situations deal with the crisp specific nodes and others can be very fuzzy.

Complexity of this problem in a crisp situation depends on the number of specified nodes. Saksena and Kumar (1966) solved the general routing problem through the k specified nodes by using the functional equation technique of dynamic programming. They assumed that \( 0 < k < n \) where \( n \) is the number of nodes in the given network with non-negative link distances.

Similarly to the routing problem through k specified nodes, earlier Arora and Kumar (1993) have considered a routing problem through a set of the specified edges.

In this paper, the routing problem with k specified nodes has been reconsidered and solved by using the minimum spanning tree (MST) of the given network. The MST approach was used recently by Munapo et al. (2014) to find the travelling salesman tour. Several other similar work were done (Abhilasha 2013; Öncan et al. 2013; Saravanan and Muthusamy 2014). This problem and its variants have several applications in vehicle routing as applied recently by Peng et al. (2013) and Zhifeng et al. (2014). The functional equation approach used by Saksena and Kumar (1966) for general routing problem through the k specified nodes has been briefly explained in Section 2. Some terms needed to develop the algorithm using the minimum spanning tree have been discussed in Section 3. The algorithm developed in this paper is presented in Section 4 and two numerical illustrations have been discussed in Section 5. Finally a few suggestions for further work have been provided in Section 6.

**The Problem**

Let the given network consist of nodes, which for convenience are denoted in any order by a sequence of numbers 0, 1, 2, ..., N-1, N. Here 0 denotes the origin node and N denotes the destination node. Let the set of specified nodes (through which the route must pass) contain \( k \) elements, where \( 0 < k < N \). Consider the case of \( k = 1 \), where there is one specified node, and let this node be denoted by ‘\( r \)’. In this case the problem reduces to two shortest route problems, that is, finding the shortest route from the node ‘0’ to the node ‘\( r \)’, and the shortest route from the node ‘\( r \)’ to the final destination node ‘N’. The sum of these two shortest routes will give the required shortest path from ‘0’ to ‘N’ passing through the node ‘\( r \)’. One can easily see that the combinations will increase when \( k > 1 \). Saksena and Kumar (1966) developed a functional equation using the principle of optimality, which is briefly presented here. They defined:

\[
D(i, j) \quad \text{to be the distance between the ordered pair} \quad (i, j), \quad \text{where} \quad i \quad \text{denotes the origin and} \quad j \quad \text{denotes the destination, and the index} \quad r \quad \text{indicates the specified nodes that occur on the optimal route,} \quad r = 0, 1, 2, \ldots
\]

\[
f_r^i \quad \text{is the minimum distance from the specified node} \quad i \quad \text{to the final destination, passing through at least} \quad r \quad \text{distinct specified nodes (the initial node} \quad i \quad \text{is not to be counted as one of the specified nodes even if it is repeated en route).}
\]

These definitions together with the principle of optimality result in the functional equation [1].

\[
f_r^i = \min \{D(i, j) + f_r^{i+1}, (j = 1, 2, \ldots) \text{and} \ j \neq i\} \quad (1)
\]

Here,

\[
f_r^0 \quad \text{is the minimum distance from the specified node} \quad i \quad \text{to the final destination, without passing through any other specified node. The initial calculation was given by Saksena and Kumar (1966) as in equation [2].}
\]

\[
f_r^i = \min \{D(i, j) + f_r^0\} \quad (2)
\]

**A REVISIT TO THE PROBLEM: COMPLEXITY AND DEFINITIONS**

**Problem Complexity**

Similar to the travelling salesman problem, the path through k specified nodes has complexity as a function of the number of specified nodes. This is explained in Table 1.
From Table 1, it is clear that as the value of k increases, the complexity of the problem increases with respect to the number of shortest route problems solved and also the number of evaluations required, before arriving at the solution. The number of evaluations is increasing in a factorial manner (Mohammard 2014; Zakir 2014).

Therefore a good approximate solution in a linear time would be desirable in many practical situations.

Terms and Definitions

**Basic and Non-basic Arcs:** An arc connecting two nodes j and j is said to be basic if \( x_{ij} = 1 \), that is, it belongs to the minimum spanning tree (MST) solution. If \( x_{ij} \neq 1 \), then that arc is not a part of the MST and it would be a non-basic arc. Many methods exist to find the MST of a given network, (see for example, Abhilasha 2013; Ahuja et al. 1990; Garg and Kumar 1968).

**Degree of a Node:** The number of basic arcs emanating from a node in the MST is the degree of that node. In a loop free route, the degree of all en route nodes is 2, except the start and the end nodes of that route.

**Neighbouring Arcs and Neighbouring Nodes:** A node j is said to be a neighbouring node to a node i, if the nodes and j are connected by a single arc. Any arc emanating from a neighbouring node is known as a neighbouring arc.

**A String in a MST** is a collection of arcs where the degree of all intermediate nodes is two and the end nodes have a degree 1 with respect to that string. In other words, more than one string may start from the same node and more than one string may have a part of the string common to them. Thus, the MST of a given graph may have several strings.

**Network Modification Theorem:** For a given MST solution, the number of basic arcs emanating from that node can be altered by adding a constant to all arcs emanating from that node without altering the MST, but creating more number of alternative MSTs.

**Motivation and Proof:** Consider that the MST solution of a network has a node with more than two basic arcs in that solution. By adding a proper constant to all arcs emanating from that node, one can create an alternative neighbouring arc to qualify to be basic, thus creating an alternative MST. If an alternative MST solution can create a string among the specified nodes, then the set of specified nodes has a defined start and end. Thus, the number of shortest route problems and evaluations will no longer be required. For detailed proof, see Munapo et al. (2013).

**METHODOLOGY**

**MST Based Approach to find the Route through the Specified Nodes**

Let the network be denoted by \( N(n,l) \), where n is a set of nodes whose elements are \( \{O, 1, 2, \ldots, N\} \) and l is an arc set with elements \( \{l_{ij}, i,j = O,1,2,\ldots,N\} \). Let the set of specified nodes denoted by ‘s’ be a subset of the set of nodes n.

The steps of the algorithm are the following:

**Step 1:** Find the shortest path without imposing specified nodes condition by any known

<table>
<thead>
<tr>
<th>No. of specified nodes (r) and their identification</th>
<th>Possible routes</th>
<th>No of shortest route problems solved / Total number of new problems solved / No of evaluations</th>
<th>Complexity pattern 2 (number of specified nodes r) + ( C_r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1; ( r_1 )</td>
<td>( O \rightarrow r_1 \rightarrow N )</td>
<td>2 / 2 / 1 = 1!</td>
<td>2(1) + 0 = 2</td>
</tr>
<tr>
<td>2; ( r_1, r_2 )</td>
<td>( O \rightarrow r_1 \rightarrow r_2 \rightarrow N )</td>
<td>3 / 5 / 2 = 2!</td>
<td>2(2) + ( C_2 ) = 5</td>
</tr>
<tr>
<td>3; ( r_1, r_2, r_3 )</td>
<td>( O \rightarrow r_1 \rightarrow r_2 \rightarrow r_3 \rightarrow N )</td>
<td>4 / 9 / 6 = 3!</td>
<td>2(3) + ( C_3 ) = 9</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>(r+1)</td>
<td>...</td>
</tr>
<tr>
<td>r; ( r_1, r_2, \ldots, r_r )</td>
<td>( O \rightarrow r_1 \rightarrow r_2 \rightarrow \ldots \rightarrow r_r \rightarrow N )</td>
<td>( 0 / 2 ) / (r) + ( C_r ) / r!</td>
<td>( 2(r) + ( C_r ) )</td>
</tr>
</tbody>
</table>

Table 1: K specified node patterns
method. Let the links on this path be denoted by the set \( SP \) (shortest path).

**Step 2:** Check if the shortest path obtained from Step 1 has visited all nodes in the set \('s\). If yes, terminate the search process and go to Step 6. Else redefine the set of specified nodes which have not been covered by the shortest path. Let the specified nodes that are not on the shortest path be denoted by \( s' \), where \( s' \leq s \). Go to Step 3.

**Step 3:** Find the MST of the given network, starting from a node in the set \( k' \). Go to Step 4. This connected graph will be comprised of all nodes, specified and others. Let the links in the MST be denoted by the set \( MST \).

**Step 4:** Find the union of the two sets (\( SP \) and \( MST \)) obtained from Steps 1 and 3. The set of links in the union set will definitely have links forming a path joining the origin and the destination nodes. This union set will contain all the specified nodes, all other nodes and selected links. All nodes and links in the union set may not be required for the desired path through the \( k \) specified nodes.

**Step 5:** Rearrange the links in the union set to form strings. For example, the links \{1,2,1,7,2,3,3,4,2,4\} will form three strings: \{1 -> 2 -> 3 -> 4, 1 -> 7 and 1 -> 2 -> 4\}. One of these strings will be the shortest path joining the origin and destination.

**Step 6:** Delete a string if does not contain any node from the set of specified nodes \( s \) and it does not contain both the origin node and the destination node. Remove loops if beneficial.

**Step 7:** Prepare the final path as a string joining the origin node to the destination node through the set of specified nodes as discussed by Munapo et al. (2013).

**RESULTS**

The researchers present two examples to illustrate various features of the proposed algorithm.

*Example 1:* Reconsider the example solved by Saksena and Kumar (1966).

The problem is to find the shortest path joining the origin nodes 0 to 9, passing through the set of specified nodes \{2, 4, 6, 8\}.

**Solution**

The shortest path using the Step 1 is given by:

\[
{\text{O-> 7 -> 5 -> 6 -> 3 -> 2 -> 4 -> 9}.} \quad (3)
\]

The length of the above string is 14. Note there are alternative paths, for example, the path \( \text{O-> 7 -> 5 -> 3 -> 2 -> 4 -> 9} \) is of the same length but the researchers choose the previous one since it has three specified nodes \{2, 4, 6\} and the alternative path has only two specified nodes \{2, 4\}.

Step 2: The specified node not visited by the selected shortest path is node 8. Thus, the set \( K' = \{8\} \).

From the Step 3, the MST starting from the node 8 will be comprised of the following links in the order of selection:

![Fig. 1. Given connected graph and arc distances](image)
A MINIMUM SPANNING TREE APPROXIMATION TO THE ROUTING

\[(8,6), (6,3), (3,2), (2,4), (4,9), (6,5), (5,7), (7,0), (O,1)\]  
\(\text{From the step 4, the union of the sets } [3] \text{ and } [4] \text{ is given by } [5], \text{ where the common links have been underlined.}
\[(8,6), (6,3), (3,2), (2,4), (4,9), (6,5), (5,7), (7,0), (O,1)\]
\(\text{Using the step 5, starting from the node } O, \text{ we form as many strings as possible. These strings are given below:}
\text{String 1: } O \rightarrow 1
\text{String 2: } O \rightarrow 7 \rightarrow 5 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 9
\text{String 3: } O \rightarrow 7 \rightarrow 5 \rightarrow 6 \rightarrow 8
\text{From the step 6, we notice that the string 1 can be deleted since the node 1 is neither the specified node nor the destination node. The other node } O \text{ has been covered in other strings.}
\text{String 3 is common to string 2, except the node 8. Therefore, we have a choice of either forming a loop } 6 \rightarrow 8 \rightarrow 6 \text{ or merging the node 8 with the path of string 2. In this case both possibilities will increase the cost equally, thus there are two answers to the problem. These two answers are given as } [6] \text{ forming a loop and as } [7] \text{ without a loop.}
\text{Path with a loop: } O \rightarrow 7 \rightarrow 5 \rightarrow 6 \rightarrow 8 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 9, \text{ cost } 14 + 6 = 20
\text{Path without a loop: } O \rightarrow 7 \rightarrow 8 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 9, \text{ cost } 14 - 5 + 8 + 3 = 20
\text{Example 2: Suppose node 1 is also a specified node. In that case the string 1 will not be discarded and the required two alternative paths would be given by [8] and [9]. Note that node 1 cannot be absorbed on any path as cost associated with a loop is only two units against 10 for moving out of node 1 other than the node } O.
\text{Alternative 1: } O \rightarrow 1 \rightarrow O \rightarrow 7 \rightarrow 5 \rightarrow 6 \rightarrow 8 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 9, \text{ cost } 20 + 2 = 22
\text{Alternative 2: } O \rightarrow 1 \rightarrow O \rightarrow 7 \rightarrow 8 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 9, \text{ cost } 20 + 2 = 22
\text{RECOMMENDATIONS AND FURTHER WORK}

It is easy to see that in the proposed approach one can easily establish lower and upper bounds on the path length. For example, the lower bound on the path length is the unconstrained path length and the upper bound is the length of the path one can get from the union set in Step 5.

It is desirable to establish a procedure of approximating the bounds and thus ensuring the optimality of the solution.

In this paper, the concept of the specified nodes has been discussed in connection with the shortest route from a given origin node to the destination node. In graph theory, many variants of the routing problem and its applications exist. For example, the nodes on the shortest path may have a time window attached to them. For a feasible solution, one has to visit the specified node within the specified time window. This simple mathematical concept has applications in patient routing in the hospitals, where windows arise due to availability of beds, theatre, doctor, radiology results etc.

ACKNOWLEDGEMENTS

The researchers are grateful to the referees for their constructive suggestions for improving the paper.

REFERENCES


