© Kamla-Raj 2014 J Economics, 5(3): 307-312 (2014)

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

Santosh Kumar[1], Elias Munapo[2], 'Maseka Lesaoana[3] and Philimon Nyamugure[4]

[1]Department of Mathematics and Statistics, University of Melbourne, Parkville, Australia
[2]Graduate School of Business Leadership, University of Kwa-Zulu-Natal, Westville Campus, Durban, South Africa
[3]Department of Statistics and Operations Research, University of Limpopo, Bag X1106, Sovenga 0727, South Africa
[4]Department 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 two extreme cases deal with loop free paths and in all other cases loop is a possibility (Kumar et al. 2013).

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

Address for correspondence:
Santosh Kumar
Department of Mathematics and Statistics,
University of Melbourne,
Parkville, Australia
E-mail: skumar@ms.unimelb.edu.au

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_j'. 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_j' and the shortest route from the node 'r_j' 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_j'. 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:

$D^r(i, j)$ to be the distance between the ordered pair (i, j), where i denotes the origin and J denotes the destination, and the index r indicates the specified nodes that occur on the optimal route, $r = 0, 1, 2, \ldots$

f^{ξ}_i is the minimum distance from the specified node i to the final destination, passing through at least ξ distinct specified nodes (the initial node i 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].

$$f_i^{\xi} = \min_j [D^r(i, j) + f_j^{\xi-r-1}], [(j = 1,2,\ldots) \; and \; j \neq i] \quad (1)$$

Here,

f^0_i is the minimum distance from the specified node i 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].

$$f_i^1 = \min_j [D(i, j) + f_j^0] \quad (2)$$

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 i 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 $\boldsymbol{i}$, 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, …, 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 1: K specified node patterns

No. of specified nodes (r) and their identification	*Possible routes*	*No of shortest route problems solved / Total number of new problems solved / No of evaluations*	*Complexity pattern 2 (number of specified nodes r) + $^rC_2$*
1; r_1	O -> r_1 -> N	2 / 2 / 1 = 1!	2(1) + 0 = 2
2; r_1, r_2	O -> r_1 -> r_2 -> N	3	
	O -> r_2 -> r_1 - N	2 / 5 / 2 =2!	2(2) + $^2C_2$ = 5
3; r_1, r_2, r_3	O -> r_1 -> r_2 -> r_3 -> N	4	
	O -> r_1 -> r_3 -> r_2 -> N	2	
	O -> r_2 -> r_1 -> r_3 -> N	1	
	O -> r_2 -> r_3 -> r_1 -> N	1	
	O -> r_3 -> r_1 -> r_2 -> N	1	2(3) + $^3C_2$ = 9
	O -> r_3 -> r_2 -> r_1 -> N	0 / 9 / 6 = 3!	
…	…	…	…
r; $r_1, r_2, \ldots, r_r$	O -> r_1 ->r_2 -> … ->r_r-> N	(r+1) …0 / 2 (r) + $^rC_2$/r! / r!	2(r) + $^rC_2$

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 $s^1 \leq s$. 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) and (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 O to 9, passing through the set of specified nodes {2, 4, 6, 8}.

Solution

The shortest path using the Step 1 is given by:

{O-> 7 -> 5 -> 6 -> 3 -> 2 -> 4 -> 9}. (3)

The length of the above string is 14. Note there are alternative paths, for example, the path 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^1={8}.

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

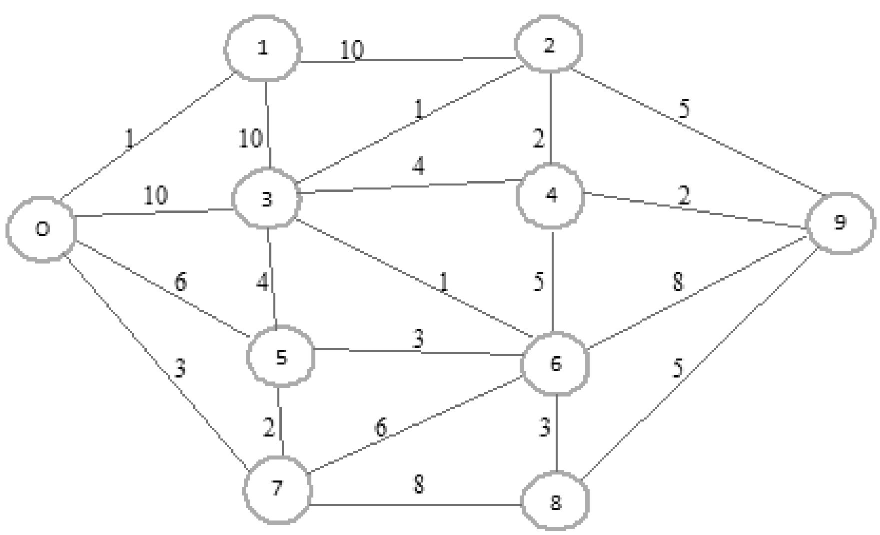

Fig. 1. Given connected graph and arc distances

{(8,6), (6,3), (3,2), (2,4), (4,9), (6,5), (5,7), (7,O), (O,1)} (4)

From the step 4, the union of the sets [3] and [4] is given by [5], where the common links have been underlined.

{(8,6),(6,3),(3,2),(2,4),(4,9),(6,5),(5,7), (7,O),(O,1)} (5)

Using the step 5, starting from the node O, we form as many strings as possible. These strings are given below:

String 1: O -> 1

String 2: O ->7 -> 5 -> 6 -> 3 -> 2 -> 4 ->9

String 3: O ->7 -> 5 -> 6 -> 8

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 has been covered in other strings.

String 3 is common to string 2, except the node 8. Therefore, we have a choice of either forming a loop 6 -> 8 -> 6 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] forming a loop and as [7] without a loop.

Path with a loop: O ->7 -> 5 **-> 6 -> 8 -> 6** -> 3 -> 2 -> 4 -> 9, cost 14 +6 = 20 (6)

Path without a loop: O ->7 -> 8 -> 6 -> 3 -> 2 -> 4 -> 9, cost 14 – 5 + 8 +3 =20 (7)

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.

Alternative 1: **O -> 1 -> O** -> 7 -> 5 **-> 6 -> 8 -> 6** -> 3 -> 2 -> 4 -> 9, cost 20 + 2 = 22 (8)

Alternative 2: **O -> 1 -> O** -> 7 -> 8 -> 6 -> 3 -> 2 -> 4 -> 9, cost 20 +2 = 22 (9)

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 moving 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

Abhilasha R 2013. Minimum cost spanning tree using Prim's Algorithm. *International Journal of Advanced Research in Computer Science and Management Studies*,1(1): 15-20.

Ahuja RK, Mehlhorn K, Orlin J, Tarjan RE 1990. Faster algorithms for the shortest path problem. *Journal of the Association of Computing Machinery*, 37(2): 213-223.

Andreas B, Mourad EO, Gerold J, Volkmar S, Anand S 2014. Exact and heuristic algorithms for the travelling salesman problem with multiple time windows and hotel selection. *Journal of the Operations Research Society*. Doi:10.1057/jors.2014.17

Arora H, Kumar S 1993. Paths through K-specified edges in a linear graph. *Opsearch*, 30(2): 163-173.

Beckwith RE, 1961. Dynamic Programming and Network Routing: An Introduction to the Technique of Functional Equations. *Appeared in the Proceedings of the Dynamic Programming Workshop*. Boulder, Colorado.

Bellman R 1956. On an optimal routing problem. *Q Appl Math*, 16: 87-90.

Bellman RE 1962. Dynamic programming treatment of travelling salesman problems. *J of ACM*, 9(1): 61-63.

Dantzig GB 1957. Discrete-variable extremum problems. *Opns Res*, 5(2): 266-277.

Garg RC, Kumar S 1968, Shortest connected graph through dynamic programming technique. *Maths Magazine*, 41(4): 170-173.

Kalaba R 1960. On some communicating network routing problems. Combinatorial Analysis. *Proc Symp Appl Math*, 10.

Kumar S, Munapo E, Ncube O, Sigauke C, Nyamugure P 2013. A minimum weight labeling method for determination of a shortest route in a non-directed network. *Int J Syst Assur Eng* Manag, DOI 10, 1007/s 198-012-0140-7.

Mikhail B, Alexander P 2014. Heuristic for a Real-life Truck and Trailer Routing Problem. *2nd International Conference on Information Technology and Quantitative Management, ITQM 2014, Procedia Computer Science*, Vl(31): 778-792.

Mohammad A, Ritika G, Shivalika S 2014. Travelling salesman problem using genetic algorithm. *International Journal of Latest Trends in Engineering and Technology (IJLTET)*. 3(3):183-190.

Munapo E, Kumar S, Lesaoana M, Nyamugure P 2014. A spanning tree based approach to determine a near optimal tour for the travelling salesman problem in a completely connected network. *Communicated for Publication Opsearch* 2014.

Nagata Y, Kokayashi S 2013. A powerful genetic algorithm using edge assembly crossover for the Travelling Salesman Problem. *INFORMS Journal on Computing*, 25(2): 346-363.

Öncan T, Zhang R, Punnen AP 2013. The minimum cost perfect matching problem with conflict pair constraints. *Computers and Operations Research* 40(4): 920-930. DOI 10.1016/j.car.2012.10.022.

Peart RM, Randolph PH, Bartlett TE 1960. The shortest route problem. *Opns Res*, 8(6): 866-868.

Peng Hai-Xia, Zhao Hai, LiDa-Zhou, Lin Chuan 2014. Data fusaggregation algorithm based on dynamic minimal spanning tree routing protocol. *Acta Phys Sin*, 63(9): 090-206.

Pollack M, Wiebenson W 1960. Solutions of the shortest-route problem – A review. *Opns Res*, 8(2): 224-230.

Rais A, Alvelos F, Carvalho MS 2014. New mixed integer-programming model for the pickup-and-delivery problem with trans-shipment. *European Journal of Operations Research*, 235(3): 530-539.

Roopa KM, Apoorva HR, Srinivasu VK, Viswanatah MC 2013. A Study of Different Algorithms For Shortest Route Problem, International Journal of Engineering Research and Technology, 2(9). From <www.ijert.org> .

Saksena JP, Kumar S 1966. The routing problem with 'K' specified nodes. *Opns Res*, 14(5): 909-913.

Saravanan M, Muthusamy M 2014. A Hybrid Optimised Weighted Minimum Spanning Tree for the Shortest Intrapath Selection in Wireless Sensor Network. *Mathematical Problems in Engineering*. 2014 (2014), article ID 713427. From <http://dx.doi.org/10.1155/2014/713427>.

Zakir HA 2014. The ordered clustered travelling salesman problem, a hybrid genetic algorithm. *The Scientific World Journal*, doi:10.1155/2014/258207.

Zhan FB 1997. Three fastest shortest path algorithms on real road networks: Data structures and procedures. *Journal of the Geographic Information and Decision Analysis*, 1(1): 70-82.

Zharfi V, Mirzazadeh A 2013. A Novel Metaheuristic for Travelling Salesman Problem. Journal of Industrial Engineering, ID 347825. From <http//dx.doi.org/10.1155/2013/347825>.

Zhifeng L, Enjian Y, Weisong H, Zheng P 2014. A vehicle routing problem solution considering alternative stop points. *Procedia-Social and Behavioral Sciences*, 138: 584-591.