IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555
Vol. 3, No.1, February 2013

An Efficient Algorithm for Multi-Objective Shortest Path Problem(MOSPP)

N.Nagamalleswara Rao
R.V.R & J. C. College of Engineering,
Department of Computer Scinece and Engineering
Guntur,Andhra Pradesh,Indial, nnmr_m@yahoo.com

Dr.O. Naga Raju, Prof. I. Ramesh Babu
Department of Computer Science and Engineering,Nagarjuna University,
Guntur, Andhra Pradesh, India {onrajunrt@gmail.com, rinampudi@hotmail. }

***Abstract*—The Shortest Path Problem is a popular optimization problem in operations research due to its wide range of practical applications. In most cases a single objective is considered, while also the multi-objective case has useful applications. Multi objective shortest path problem is an extension of single objective shortest path problem. The single objective shortest path problem will include only one objective whereas the multi objective shortest path will include multiple objectives like cost, time, distance etc. To determine the shortest path is one of the major problem in the field of computer science. There may be several pareto optimal solutions for a multi-objective shortest path problem. The decision maker can select the most satisfactory solution depending on the priority and nature of the problem. The Multi objective shortest path problem can be solved by means of Geoffrion's technique (converting multi objective to single objective problem) and the modified Dijkstra's algorithm will be then used to determine the shortest path. Here we use the concept of properly pareto optimal solution for MOSSP and it is stronger than pareto optimal solution using Geoffrion's approach.**

Keywords-Multi objective shortest path;properly pareto optimal solution;Geoffrion's technique;Modified Dijkstra's algorithm

1. Introduction

Jose Maria A. et al. [1] have presented an overview of the multi objective shortest path problem(MSPP) and a review of essential and recent issues regarding the methods to its solution. Rahim A et al. [2] have addressed the problem of time-dependent shortest multimodal path in complex and large urban areas. C. Chitra and P. Subbaraj [3] have presented Multi objective Optimization Solution for Shortest Path Routing Problem. The shortest path routing problem was a multi objective non linear optimization problem with constraints. This problem had been addressed by considering Quality of service parameters, delay and cost objectives separately or as a weighted sum of both objectives. Subbaraj Potti and Chitra Chinnasamy [4] have propose a new multi-objective approach, Strength Pareto Evolutionary Algorithm (SPEA), to solve the shortest path routing problem. The routing problem was formulated as a multi-objective mathematical programming problem which attempted to minimize both cost and delay objectives simultaneously. K.Karthikeyan [5] has presented a comparative study of multi objective shortest path problems. For a multi objective shortest path problem (MOSPP) of a network, there were several Pareto optimal solutions. The decision maker selected the best one or the most satisfactory solution depending on the priority and nature of the problem. In this they used the concept of Properly Pareto optimal solution for MOSSP and proved that it was stronger than Pareto optimal solution using Geoffrion's approach. source node to destination is called shortest path problem (SPP) [6].

Multi objective shortest path problem is an extension of single objective shortest path problem. The single objective shortest path problem will include only one objective whereas the multi objective shortest path will include multiple objectives like cost, time, distance etc. The Multi objective shortest path problem can be solved by means of Geoffrion's technique (converting multi objective to single objective problem) and the Dijkstra's algorithm will be then used to determine the shortest path.The single objective shortest path problem with only one objective is normally represented as,

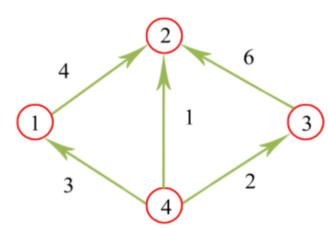


Fig1.Single Objective Shortest path problem

In the single objective shortest path problem which has been mentioned above consists of only one objective and by this single objective the shortest path can be found out. However the multi objective shortest path problem will consists of more than one objective and it should be converted to single objective and then the shortest path will be found out. The multi objective shortest path problem with multiple objectives are represented as,

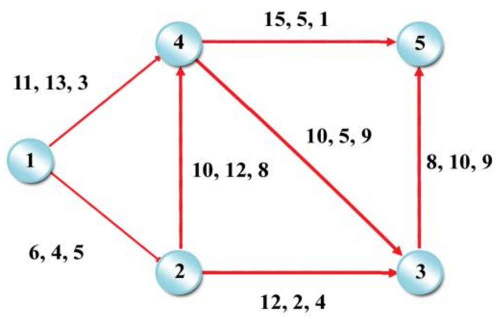


Fig 2: Multi Objective Shortest Path Problem

The multi objective shortest path problem given above consists of three objectives. The above multi objective shortest path problem can be solved by means of converting the multi objective shortest path problem into the single objective shortest path problem. Then the single objective shortest path problem can be solved by means of modified Dijkstra's algorithm. The solution obtain for the single objective shortest path problem will be an efficient solution for the multi objective shortest problem.

2. GEOFFRION'S APPROACH

Geoffrion's technique can be basically defined as, let $\lambda > 0$ with $\sum_{i=1}^{k} \lambda_i = 1$ be the fixed value if x_0 is a solution for the minimization problem p_λ . In the above mentioned problem three objectives are specified and it can be converted to single objective by means of taking λ value as 1/3. The single objectives are represented as follows (11+13+3)/3 = 9; (15+5+1)/3 =7; (8+10+9)/3 = 9; (12+2+4)/3 = 6; (6+4+5)/3 = 5; (10+12+8)/3 = 10; (10+5+9)/3) = 8 ; Thus the single objective shortest path problem obtained by the Geoffrion's approach for the above mentioned multi objective shortest path problem can be given as,

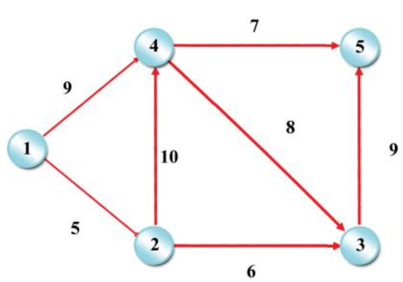


Fig 3: The converted single objective shortest path problem

The single objective shortest path problem can be solved by means of modified Dijkstra's algorithm. The modified dijkstra's algorithm.

3. MODIFIED DIJKSTRA'S ALGORITHM

Dijkstra's algorithm is a solution to the single source shortest path problem in graph theory. It works on both directed and undirected graphs. However, all edges must have nonnegative weights. Assuming that all arc lengths are nonnegative, the following method, known as Dijkstra's algorithm, can be used to find the shortest path from a node (say, node 1) to all other nodes. Dijkstra's algorithm can be used to solve such shortest path problems. It finds the shortest path from a given node s to all other nodes in the network. Node s is called a starting node or an initial node. Dijkstra's algorithm starts by assigning some initial values for the distances from node s and to every other node in the network. It operates in steps, where at each step the algorithm improves the distance values. At each step, the shortest distance from node s to another node is determined. The algorithm characterizes each node by its state. The state of a node consists of two features, they are distance value and status label. Distance value of a node is a scalar representing an estimate of the its distance from node s. Status label is an attribute specifying whether the distance value of a node is equal to the shortest distance to node s or not. The status label of a node is Permanent if its distance value is equal to the shortest distance from node s. Otherwise; the status label of a node is Temporary. The algorithm maintains the step by step updates and the states of the nodes. At each step one node is designated as the current node.

To begin, label node 1 with a permanent label of 1. Then we label each node i that is connected to node 1 by a single arc with a "temporary" label equal to the length of the arc joining node 1 to node i. Each other node (except, of course, for node 1) may have a temporary label of 0. Choose the node with the largest temporary label and make this label permanent. Now suppose that node i has just become the (k +1)th node to be given a permanent label. Then node i is the kth closest node to node 1. At this point, the temporary label of any node (say, node i) is the length of the shortest path from node 1 to node i that passes only through nodes contained in the k - 1 closest nodes to node 1. For each node j that now has a temporary label and is connected to node i by an arc, replace node j's temporary label with maximum value.

The new temporary label for node j is the length of the shortest path from node 1 to node j that passes only through nodes contained in the k closest nodes to node 1. Now make the largest temporary label a permanent label. The node with this new permanent label is the (k + 1) th closest node to node 1. Continue this process until all nodes have a permanent label. To find the shortest path from node 1 to node j, work backward from node j by finding nodes having labels differing by exactly the length of the connecting arc. Of course, if we want the shortest path from node 1 to node j, we can stop the labeling process as soon as node j receives a permanent label.

Dijkstra's algorithm is a greedy algorithm which makes choices that currently seem the best and locally optimal does not always mean globally optimal. It maintains following two properties

- for every known vertex, recorded distance is shortest distance to that vertex from source vertex
- for every unknown vertex v, its recorded distance is shortest path distance to v from source vertex, considering only currently known vertices and v

Many more problems than you might at first think be able to be cast as shortest path problems, making Dijkstra's algorithm a powerful and general tool. For example:

• Dijkstra's algorithm is applied to automatically find directions between physical locations, such as driving directions on websites like Map quest or Google Maps.

• In a networking or telecommunication applications, Dijkstra's algorithm has been used for solving the min-delay path problem (which is the shortest path problem). For example in data network routing, the goal is to find the path for data packets to go through a switching network with minimal delay.

• It is also used for solving a variety of shortest path problems arising in plant and facility layout, robotics, transportation, and VLSI design.

4. ALGORITHM STEPS

The dijkstra's algorithm includes three steps they are initialization, iteration, halting. These three steps are described in the following manner. For the initialization, set the current distance to the initial vertex as 1 and for all other vertices, set the current distance to 0. All vertices are initially marked as unvisited and so set the pointer for all vertices to null. For iteration, find an unvisited vertex which has the longest distance to it and mark it as visited. For each unvisited vertex which is adjacent to the current vertex add the distance to the current vertex to the weight of the connecting edge, if this is greater than the current distance to that vertex, update the distance and set the parent vertex of the adjacent vertex to be the current vertex. For Halting, successfully halt when the vertex we are visiting is the target vertex and if at some point, all remaining unvisited vertices have distance 0, then no path from the starting vertex to the end vertex exits.

The various steps like initialization, Distance Value Update and Current Node Designation Update, Termination Criterion which are involved in the dijkstra's algorithm can be specified with the following table.

Step1. Initialization

- Assign one as the distance value to node s, and label it as Permanent. [The state of node s is (1, p)].
- Assign to every node a distance value of 0 and label them as Temporary. [The state of every other node is (0, t)].
- Designate the node s as the current node.

Step2. Distance Value Update and Current Node Designation Update

Let i be the index of the current node.

(1) Find the set J of nodes with temporary labels that can be reached from the current node i by a link (i, j). Update the distance values of these nodes.

- For each $j \in J$, the distance value d_j of node j is updated as follows
- New $dj = \max\{dj, di + cij\}$ (1)

Where c_{ij} is the cost of link (i, j), as given in the network problem.

(2) Determine a node j that has the largest distance value dj among all nodes $j \in J$, find j∗ such that

$$\max_{j \in J} \quad dj = dj^* \tag{2}$$

(3) Change the label of node j* to permanent and designate this node as the current node.

Step3. Termination Criterion

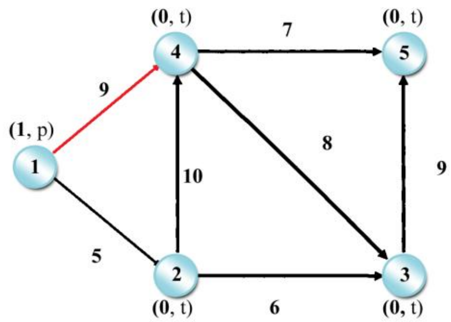
- If all nodes that can be reached from node s have been permanently labeled, then stop it was done.
- If we cannot reach any temporary labeled node from the current node, then all the temporary labels become permanent and it was done.
- Otherwise, go to Step 2.
-

APPLICATIONS

- Traffic information systems use Dijkstra's algorithm in order to track the source and destinations from a given particular source and destination.
- OSPF- Open Shortest Path First, used in Internet routing. It uses a link-state in the individual areas that make up the hierarchy. The computation is based on Dijkstra's algorithm which is used to calculate the shortest path tree inside each area of the network.

Example for modified dijkstra's algorithm:

Take the previous example to find the shortest path by using modified dijkstra's algorithm. As mentioned above, node 1 is the current node and is marked as permanent. All the other nodes are marked as the temporary nodes. It is given as follows

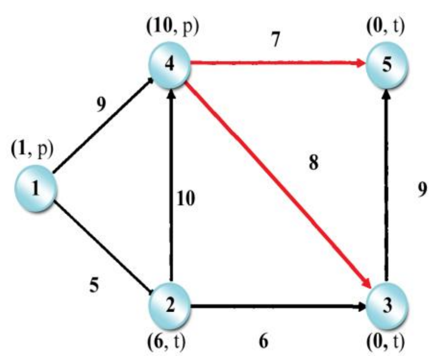


Node 1 can be considered as the current node. The state of the current node will be (1, p) and all other nodes have the state of (0, t). Nodes 2 and 4 can be reached from the node 1.Update the distance values for these nodes by using equations (1) and (2).

$$d2 = \max(0,1+5) = 6 \quad (1)$$

$$d4 = \max(0,1+9) = 10 \quad (2)$$

Node 4 has the maximum value and so it becomes the current node. The label of node 4 is permanent. Node 2 remains temporary.

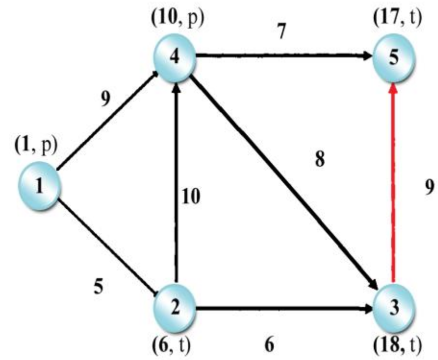


Nodes 3 and 5 can be reached from node 4. The values are updated for nodes 3 and 5.

$$d3 = \max(0,10+8) = 18 \quad (3)$$

$$d5 = \max(0,10+7) = 17 \quad (4)$$

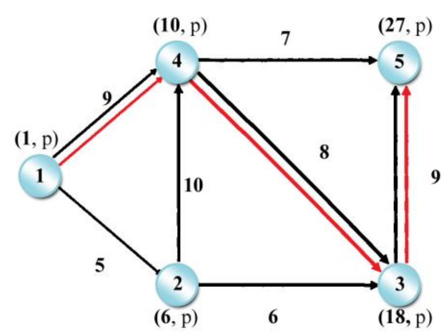
Node 3 has the maximum value and so it becomes the current node. The label of node 3 is permanent. Node 5 remains temporary.



Node 5 can be reached from node 3. The distance value is updated for node 5 as follows.

$$d5 = \max(17,18+9) = 27 \quad (5)$$

Node 5 has become permanent. Now node 2 has only temporary label. Change the temporary label of node 2 as permanent. The shortest path from node 1 to node 5 is obtained as follows.



The shortest path thus obtained by the modified dijkstra's algorithm is 1-4-3-5. The modified algorithm finds the efficient shortest path than that of the existing algorithm.

Conclusion

Thus the Geoffrion's technique is used to convert the multi objective shortest path problem into single objective shortest path problem and the modified Dijkstra's algorithm is used to find the solution for the single objective shortest path problem. The solution obtained for the single objective shortest path problem is the efficient solution for the multi objective shortest path problem.

REFERENCES

[1] Jose Maria A. Pangilinan, and Gerrit K. Janssens, "Evolutionary Algorithms for the Multi objective Shortest Path Problem "International Journal of Computer and Information Engineering, pp.322-327,2007.

[2] Rahim A. Abbaspour, and Farhad Samadzadegan, "An Evolutionary Solution for Multimodal Shortest Path Problem in Metropolises ", Vol. 7, No. 4, Dec 2010.

[3] C. Chitra, and P. Subbaraj, "Multi objective Optimization Solution for Shortest Path Routing Problem ",International Journal of Computer and Information Engineering, Vol. 4, No 2, 2010.

[4] Subbaraj Potti, and Chitra Chinnasamy, "Strength Pareto Evolutionary Algorithm based Multi-Objective Optimization for Shortest Path Routing Problem in Computer Networks ",Journal of Computer Science, Vol. 7, Issue 1, pp. 17-26, 2011.

[5] K.Karthikeyan, " A Comparative Study Of Multi-Objective Shortest Path Problems", International Journal of Engineering Science and Technology(IJEST), Vol. 3, No.4, pp. 2875-2879, April 2011.

[6] S. Ismail Mohideen and B.Rajesh, "A Comparative Analysis of Multi Objective Shortest Path Problem ", International Journal of Engineering Science and Technology, Vol. 2, pp.3241-3243, 2010.

[7] Andrea Raith, Matthias Ehrgott, "A Comparison of Solution Strategies for Bi objective Shortest Path Problems", Computers & Operations Research, Vol. 36, pp.1299–1331,April 2009.

[8] Radu Marinescu, "Efficient Approximation Algorithms for Multi-objective Constraint Optimization", In Proceedings of the Second international conference on Algorithmic decision theory,PP.150-164 ,2011.

[9] Gerrit K. Janssens, and Jose Maria Pangilinan, "An Empirical Evaluation of Martins' Algorithm for the Multi-Objective Shortest Path Problem ", In proceedings of European Simulation and Modeling Conference, pp. 252-256,2011.

[10] Dilpreet kaur, Dr.P.S Mundra, "Ant Colony Optimization: A Technique Used for Finding Shortest Path", International Journal of Engineering and Innovative Technology (IJEIT), Vol 1, No 5, May 2012.

[11] Mahdavi, N. Mahdavi-Amiri, and S. Nejati, "Algorithms for Bi objective Shortest Path Problems In Fuzzy Networks", Iranian Journal of Fuzzy Systems, Vol. 8, No. 4, pp. 9-37, 2011.

[12] R. Sophia Porchelvi, A.Nagoorgani, R.Irene Hepzibah, "An Algorithmic Approach to Multi Objective Fuzzy Linear Programming Problem ", International Journal of Algorithms, Computing and Mathematics, Vol. 3, No 4, November 2010.