International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 10, October - 2013

Empirical Analysis of Minimum Spanning Tree for Directed graph

Ms. Dharmishtha.R.Chaudhari *, Ms. Reema Patel **
SVNIT Surat

Abstract

We consider the problem of finding the cost of Minimum weighted spanning tree for directed graph. This problem is a classical problem in the field of dynamic graph algorithms. Edmond proposed polynomial time algorithm for above problem which gives reduced cost compare to Prim's algorithm. Our contributions include the results on the Empirical analysis of Edmond's algorithm.

1. Introduction

A tree is a connected graph with no cycles. A spanning tree is a subgraph of G which has the same set of vertices of G and is a tree. In general the problem of finding a minimum spanning tree for a weighted directed graph is difficult but solvable. There are a lot of differences between problems for directed and undirected graphs, therefore the algorithms for undirected graphs cannot usually be applied to the directed case. In this paper we examine one of the solution for minimum spanning tree of a directed graph .Minimum spanning trees are useful in constructing networks, by describing the way to connect a set of sites using the smallest total amount of wire.

2. Minimum weighted spanning tree for directed graph

A minimum spanning tree of a weighted graph G is the spanning tree of G whose edges sum to minimum weight.Let G(V,E) be a directed graph with a distinguished root vertex r and real valued cost C(v,w) on each edge (v,w). We denote number of vertices by n and number of edges by m. We assume that every vertex of G is reachable from root vertex r. A minimum spanning tree of G is spanning tree rooted at r (a set of n-1 edges containing paths form r to every vertex) of minimum total edge cost. Edmonds[1] devised a polynomial time algorithm for finding a minimum spanning tree. Edmond's correctness proof uses concepts of linear programming.

3. Prim's Algorithm for Minimum Spanning Tree

Firstly we discuss Prim's Algorithm [2] to find out Minimum cost spanning tree for weighted directed graph. Prim's algorithm is described as below:

The Prim-MST(G)
    Select an arbitrary vertex s to start the tree from.
    While (there are still non-tree vertices)
        Select the edge of minimum weight between a tree Add the selected edge and vertex to the tree Tprim.

This creates a spanning tree, since no cycle can be introduced, but is it minimum? PRIM algorithm, which solves the *undirected* MST problem, is shorthanded to solve the directed counterpart. The following example Fig.1 exhibits that the iterative greedy decision may no longer sequentially give the optimal solution because the Total cost of the below graph we get using it is equal to 16, while the actual Minimum spanning Tree cost we get is equal to 12. We must consider the other solution rather than Prim's approach to find out MST for directed graph.

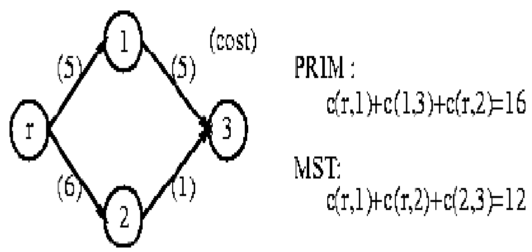

Fig.1 Total cost by Prim's algorithm and MST cost

4. Solving the MST Problem for Directed graph

Edmonds [1], Chu and Liu [3], and Bock [4] have independently given efficient algorithms for finding the MST on a directed graph. The Chu-Liu and Edmonds algorithms are virtually identical; the Bock algorithm is similar but stated on matrices instead of on graphs. Furthermore, a distributed algorithm is given by Humblet [5]. In the sequel, we shall briefly illustrate the Chu-Liu/Edmonds algorithm, following by a comprehensive example (due to [1]). Reader can also refer to [6] [7] for an efficient implementation, *O(mlogn)* and *O(n^2)* for dense graph, of this algorithm.

Chu-Liu/Edmonds Algorithm

1. Discard the arcs entering the root if any; For each node other than the root, select the entering arc with the smallest cost; Let the selected *n-1* arcs be the set *S*.
2. If no cycle formed, *G(N,S)* is a MST. Otherwise, continue.
3. For each cycle formed, contract the nodes in the cycle into a pseudo-node *(k)*, and modify the cost of each arc which enters a node *(j)* in the cycle from some node *(i)* outside the cycle according to the following equation.
   c(i,k)=c(i,j)-(c(x(j),j)-min_{j}(c(x(j),j))
   where *c(x(j),j)* is the cost of the arc in the cycle which enters *j*.

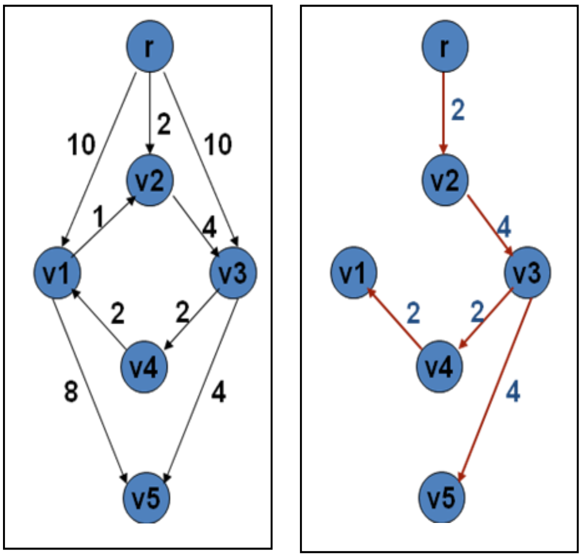

Fig.2 (A) Original Graph(G) (B) MST for G using Admond's Algorithm(Total Cost of MST=14)

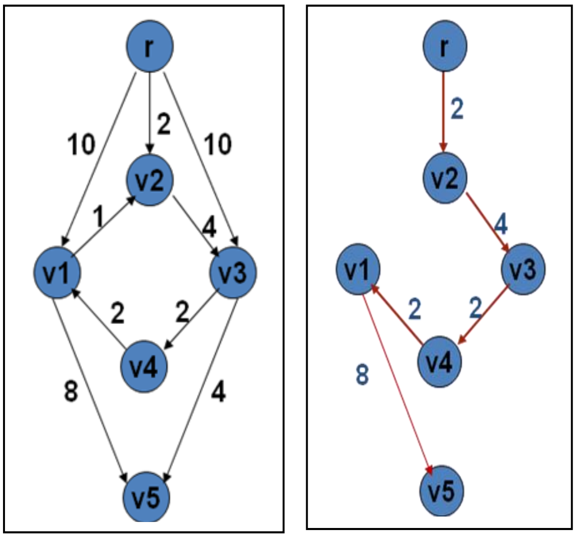

Fig.3 (A) Graph(G) (B)MST of G using Prim's Algorithm (Total Cost of MST=18)

4. For each pseudo-node, select the entering arc which has the smallest modified cost; Replace the arc which enters the same *real* node in *S* by the new selected arc.
5. Go to step 2 with the contracted graph.

The key idea of the algorithm is to find the replacing arc(s) which has the minimum *extra cost* to eliminate cycle(s) if any. The given equation exhibits the associated extra cost. Fig. 2(A) shows the example graph for which we would like to find out the MST. In the given weighted graph r represent root node and v1,v2,v3,v4,v5 other nodes in the graph. The corresponding Minimum spanning tree using Edmond's algorithm is shown in Fig. 2(B). The total cost for MST we find out here is 14. For the same graph G, We determine the total cost using Prim's algorithm that is=18 as shown in Fig. 3(B). Edmond's gives us the optimum solution to find out MST for directed graph compared to Prim's algorithm.

3. Empirical Analysis from our implementation of Chu-Liu/Edmonds Algorithm

The directed graph has been created in Turbo c to perform the timing analysis of Edmond's algorithm. The start time and end time is measured in terms of Millie seconds. The Link list data structure is used to implement the algorithm. The flowchart of entire implementation is shown in Fig. 4. The analysis of the algorithm is shown in Table 1. As varying the number of nodes the time variation is linear.

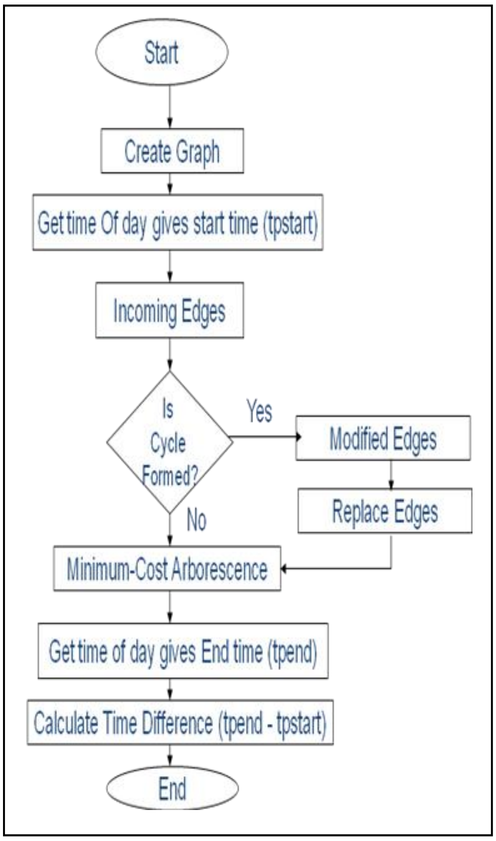

Fig. 4 Flowchart of Implementation of Edmond's algorithm

TABLE I
TIME IN MSEC PER N

n	Time (msec)
4	57
6	187
8	199
10	234
12	240
17	245
20	350
50	635
100	867
500	2500

6. Conclusion

In the previous section it is shown that algorithms for directed graphs cannot be directly applied to a directed case. We can apply the better techniques to get the optimum solution for directed graph. The future work can be find out the minimum spanning forest using the above approach.

10. References

[1] J. Edmonds, ``Optimum branchings'', J. Research of the National Bureau of Standards, 71B, 1967, pp.233-240.
[2] E. Lawler, ``Combinatorial optimization: networks and matroids'', Saunders College Publishing, 1976.
[3] Y. J. Chu and T. H. Liu, ``On the shortest arborescence of a directed graph'', Science Sinica, v.14, 1965, pp.1396-1400..
[4] F. Bock, ``An algorithm to construct a minimum spanning tree in a directed network'', Developments in Operations Research, Gordon and Breach, NY, 1971, pp. 29-44.
[5] P. Humblet, ``A distributed algorithm for minimum weighted directed spanning trees'', IEEE Trans. on Communications, v.COM-31, n.6, 1983, pp.756-762.
[6] R. E. Tarjan, ``Finding Optimum Branchings'', Networks, v.7, 1977, pp.25-35.
[7] P.M. Camerini, L. Fratta, and F. Maffioli, ``A note on finding optimum branchings'', Networks, v.9, 1979, pp.309-312. "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
[8] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
[9] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
[10] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1997