A Heuristic Approach for Solving Minimum Routing Cost Spanning Tree Problem

Quoc Phan Tan

Abstract—Minimum routing cost spanning tree - MRCT is one of spanning tree optimization problems having many applications in network design. In general cases, the problem is proved as NP-hard. This paper is going to propose 2 heuristic-based algorithms used for MRCT. The general idea of those algorithms is to start at a spanning tree and step by step improve edges to obtain better spanning tree. We conducted experiment implementations for these proposals and obtained better result than the result of current approximate algorithms.

Index Terms—Routing cost spanning tree, heuristic algorithm, gradually edge-removal algorithm, gradually edge-replacement algorithm.

I. MINIMUM ROUTING-COST SPANNING TREE PROBLEM

In this section, we are going to represent some main terms related to MRCT problem, traditional approachs and their drawbacks.

Given G = (V,E,w) is an undirected connected graph having non-negative edge weights (costs); in which V is the node set, E is the edge set, w is the cost matrix. Suppose T is a spanning tree in G, the routing cost of T, denoted by C(T), is the total routing costs of all vertex pairs in T, in which the routing cost of a vertex pair (u,v) in T, denoted by $d_T(u,v)$, is the sum over edge costs on the path connecting vertex u and vertex v in T. So, by definitions, we have:

$$C(T) = \sum_{u,v \in V} d_T(u,v) \tag{1}$$

The problem requirement is to to find the one having minimum routing cost among all possible spanning trees in G[3].

Computing spanning tree routing cost of the one having n nodes in MRCT problems by definition occupies $O(n^2)$ time. However, by the definition of "routing load" below we could compute spanning tree routing cost within linear time.

Given a spanning tree T having edge set E(T). If remove an edge e from T, T is then separated into 2-subtrees of T_1 and T_2 having the node set of $V(T_1)$ và $V(T_2)$ respectively. Routing load of e is defined as follows: $I(T,e) = 2 |V(T_1)| \cdot |V(T_2)|$. The formula (1) is then equivalent to formula (2) as follows:

$$C(T) = \sum_{e \in E(T)} l(T, e).w(e)$$
(2)

The MRCT problem is proved to be of NP-hard class. Edge weights and spanning tree topology are two factors affecting on spanning tree routing cost. The spanning tree

Manuscript received May 27, 2012; revised June 15, 2012. Quoc Phan Tan is with Department of Information Technology of Saigon University, Ho Chi Minh city, Vietnam (email: phantanquoc@gmail.com). topology affects highly on the graphs in which the bias of edge weights is not too high.

Constructing a minimum routing cost spanning tree is equivalent to constructing a spanning tree so that the average length of vertex pairs is at least. The problem plays important role in applications of network system building. Specifically, peer to peer network is an example in which the ability of data transfer and all node priorities are equal (the problems origin and its applications are available in [1][3])

Example 1: Given a spanning tree as Fig.1

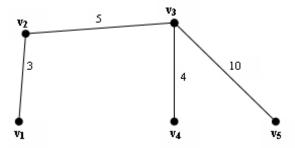


Fig. 1. This spanning tree has spanning cost value of 196

According to formula (1), we have: $d_T(v_1,v_2) + d_T(v_1,v_3) + d_T(v_1,v_4) + d_T(v_1,v_5) + d_T(v_2,v_3) + d_T(v_2,v_4) + d_T(v_2,v_5) + d_T(v_3,v_4) + d_T(v_3,v_5) + d_T(v_4,v_5) = 98.$ Since $d_T(v_i, v_i) = d_T(v_i, v_i)$, so $C(T) = 98 \times 2 = 196$.

II. APPOXIMATION APPROACHS USED FOR MRCT PROBLEMS

The first is Wong algorithm proposed by Richard Wong in 1980, Wong algorithm has 2-approximation and occupies O(nm + n²log n). Wong algorithm uses the concept of shortest path tree-SPT starting each vertex and visiting to the others. The main idea of Wong algorithm is to find SPTs having the root starting at each vertex, then select the SPT having at least cost among found SPTs. This algorithm used to establish initial solutions for metaheuristic-based algorithm in solving MRCT problem [1].

The second is an algorithm based on the idea of General Start proposed by author group of Bang Ye Wu and Kun-Mao Chao [3]; this algorithm has 3/2 approximation and occupies $O(n^4)$. The author group also proposed Polynomial Time Approximation Scheme – PTAS enabling us to find out a spanning tree having routing cost approximate within 1+e times of the best spanning tree routing cost, where is e is desired quality. The algorithm occupies $O(n^{2\lceil \frac{1}{2} \rceil - 2})$

The third is Add algorithm proposed by Vic Grout in 2005. Add algorithm occupies O(n log n). Add algorithm considers vertex degree as primary condition to construct spanning tree, instead edge weights. Add algorithm says: Find the vertex v

having the most numbers of unvisited vertices incident to v, then insert all edges incident to v into T so that T has no cycle inside, this process repeats until all vertices in graph have been inserted into T. The algorithm commonly used in homogeneous-graph and nonhomogeneous graph (from here, we call it as uniformed distribution graph i.e. a kind of graph having insignificant bias of edge weights [4].

The fourth is Campos algorithm proposed by the author group of Rui Campos and Manual Ricardo in 2008; the algorithm has 2-approximation and occupies $O(m + n \log n)$.

This is also considered as the fastest 2-approximation algorithm in present; Campos algorithm combines the ideas of Add, Prim, Dijstrak based Prim algorithm [6].

It says possibly: above approximation algorithms could not find out highly exact solutions but it has advantage of time cost and ensuring solution quality when applied into MRCT problem.

III. HEURISTIC APPROACH FOR MRCT PROBLEM

In this section, we are proposing 2-heuristic algorithms to solve MRCT problem.

A. Gradually Edge-Replacement (H1)

The idea of edge-replacement makes use of Prims or Kruskal algorithms to find the minimum spanning tree of graph, then replace gradually each edge of spanning tree with a better edge.

Step 1: Find a minimum spanning tree T in G.

Step 2:Insert respectively edge e in the edge set of E-T into T, certainly $T \cup e$ will form a cycle, in this new cycle, we find the best edge e' so that $T-e \cup e'$ has better cost than the cost of T; if there exists edge e', then replace T with $T \cup e - e'$.

Repeat step 2 until in a loop there is no edge replacement could be done in the spanning tree.

Example 2: Given a graph G as Fig.2:

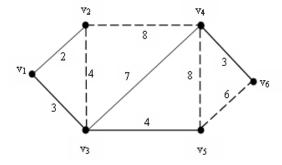


Fig. 2. The obtained spanning tree has the routing cost value of 250 following Edge-Replacement algorithm.

If we apply Edge-Replacement algorithm into fig.3, the first obtained minimum spanning tree have the set of edges $\{(v_1,v_2), (v_1,v_3), (v_3,v_5), (v_4,v_6), (v_5,v_6)\}$. Apply each step of Edge-Replacement algorithm: When replace the first 4-edges $(v_1,v_2), (v_1,v_3), (v_2,v_3), (v_2,v_4)$ in T, we also obtain a spanning tree having a routing-cost value of 266; When we replace 5^{th} -edge (v_3, v_4) with edge (v_5, v_6) , and we then obtained the routing-cost value of 250. This is just the desired spanning tree.

B. Gradually Edge-Removal (H2)

Step 1:Find the best SPT by Wong algorithm among all possible SPTs in G:

Step 2: Remove gradually each e of T; with each e find the best e' within $E-\{e\}$, suppose T'= T-e+e'. If T' better than T then replace T=T'.

Repeat step 2 until in a loop there is no edge removal could be done in the spanning tree.

IV. EXPERIMENTS

This section is going to compare experimental results of proposed algorithms against Wong, ADD, CAMPOS.

A. Experimental System

All proposed algorithms were implemented in C++ under DEV CPP compiler on the computer powered by a 2.26Ghz processor and 4 GB RAM.

We first conducted experiments on general graphs, took the obtained results into consideration on some special graphs such as homogeneous graphs, graphs with uniform edge distribution and graphs with non-uniform edge distribution.

Experiment data were generated randomly. The graph size we used in experiments has the number of nodes in range [20..200] and the number of edges in range [50..2400]. The routing costs obtained by the algorithms in experimental table are displayed as ½ of the value obtained from the formula (2).

B. General Graph

Generating general graph

General graphs G = (V, E, w) were generated as follows: we first constructed randomly a spanning tree of n = |V| nodes and n-1 edges then inserted randomly other m-(n-1)valid edge; all edge weights of graphs are random integers in range [1..2500].

| TABLE | I A: GENERAL GRAPH |
|-------|--------------------|
| ADD | CAMBOS |

| Test | WONG | ADD | CAMPOS | H1 | H2 |
|------|---------|----------|---------|---------|---------|
| 1 | 3408 | 5358 | 3725 | 3416 | 3408 |
| 2 | 8760 | 15670 | 8569 | 8552 | 8552 |
| 3 | 29915 | 51472 | 30063 | 29799 | 29799 |
| 4 | 14784 | 41531 | 15026 | 14784 | 14784 |
| 5 | 40242 | 101790 | 47270 | 39945 | 39945 |
| 6 | 185248 | 429124 | 254654 | 182349 | 182349 |
| 7 | 1145919 | 4350050 | 1420996 | 1128132 | 1128132 |
| 8 | 3178505 | 11375802 | 3765028 | 3099462 | 3099462 |
| 9 | 3374998 | 17056890 | 3865164 | 3360491 | 3357145 |
| 10 | 5485453 | 23448926 | 6303738 | 5704751 | 5474075 |
| 11 | 1384422 | 7064096 | 1483611 | 1372739 | 1372739 |
| 12 | 2964078 | 16402988 | 3611100 | 2934906 | 2957986 |
| 13 | 5311194 | 19142486 | 5903426 | 5214430 | 5214430 |
| 14 | 6605587 | 27951407 | 8284471 | 6567690 | 6567690 |
| 15 | 1908398 | 6901936 | 2336558 | 1923726 | 1881204 |

When comparing H₁ and H₂ over against WONG, ADD, CAMPOS through 15 general graph tests, we obtained result as Table I B.

TABLE I B: EXPERIMENTAL RESULTS

| H1 | WONG | | ADI | ADD | | OS |
|------------|----------|----|----------|-----|----------|-----|
| пі | Quantity | % | Quantity | % | Quantity | % |
| Better | 73 | 15 | 100 | 15 | 100 | 73 |
| Equivalent | 7 | 0 | 0 | 0 | 0 | 7 |
| Worse | 20 | 0 | 0 | 0 | 0 | 20 |
| H2 | Quantity | % | Quantity | % | Quantity | % |
| Better | 13 | 87 | 15 | 100 | 15 | 100 |
| Equivalent | 2 | 13 | 0 | 0 | 0 | 0 |
| Worse | 0 | 0 | 0 | 0 | 0 | 0 |

C. Homogeneous Graph

Generating homogeneous graph

Homogeneous graphs were generated as follows: we first chose a random value as a homogeneous value for edges, suppose $\Delta \in [1..2500]$. Then constructed a random spanning tree of n = |V| nodes and n-1 edges, and finally inserted randomly other m-(n-1) valid edges. All edges in G were attached with a positive value $\Delta \pm \mu$ in which μ is a small integer.

TABLE II A: HOMOGENEOUS GRAPH

| | TABLE II A. HOMOGENEOUS GRAFFI | | | | | | | |
|------|--------------------------------|---------|---------|---------|---------|--|--|--|
| Test | WONG | ADD | CAMPOS | H1 | H2 | | | |
| 16 | 72818 | 74094 | 106012 | 71642 | 71714 | | | |
| 17 | 98150 | 102307 | 131853 | 96971 | 96758 | | | |
| 18 | 144751 | 148596 | 189264 | 159118 | 143076 | | | |
| 19 | 342910 | 341099 | 433157 | 352204 | 335517 | | | |
| 20 | 474658 | 566614 | 590918 | 471100 | 466340 | | | |
| 21 | 560523 | 564263 | 698049 | 560838 | 548735 | | | |
| 22 | 520997 | 549681 | 636465 | 545616 | 508497 | | | |
| 23 | 117392 | 117676 | 158210 | 128354 | 115736 | | | |
| 24 | 329528 | 340386 | 405366 | 338660 | 323524 | | | |
| 25 | 6461300 | 7127108 | 7133250 | 6603778 | 6372566 | | | |
| 26 | 3183150 | 3369136 | 3813918 | 3237188 | 3107894 | | | |
| 27 | 1345428 | 1393342 | 1774572 | 1393770 | 1315424 | | | |
| 28 | 2242772 | 2414476 | 2699136 | 2219240 | 2207774 | | | |
| 29 | 879674 | 885659 | 1268719 | 867713 | 869739 | | | |
| 30 | 2952464 | 2929879 | 3782156 | 3205149 | 2914822 | | | |

When comparing H_1 and H_2 over against WONG, ADD, CAMPOS through 15 tests of general graphs, we obtained result as Table II B.

TABLE II B: EXPERIMENTAL RESULTS

| TABLE II B. LAI ERIMENTAE RESCEIS | | | | | | | |
|-----------------------------------|----------|-----|----------|-----|----------|-----|--|
| H1 | WONG | | AD | ADD | | OS | |
| пі | Quantity | % | Quantity | % | Quantity | % | |
| Better | 5 | 33 | 10 | 67 | 15 | 100 | |
| Equivalent | 0 | 0 | 0 | 0 | 0 | 0 | |
| Worse | 10 | 67 | 5 | 33 | 0 | 0 | |
| H2 | Quantity | % | Quantity | % | Quantity | % | |
| Better | 15 | 100 | 15 | 100 | 15 | 100 | |
| Equivalent | 0 | 0 | 0 | 0 | 0 | 0 | |
| Worse | 0 | 0 | 0 | 0 | 0 | 0 | |

D. Edge Distribution Factor

Generating graphs under edge distribution

Graphs with uniform edge distribution are the graphs in which node degrees are equivalent or insignificant difference.

Graphs with uniform edge distribution we used were generated as follows: we first determined a parameter $r = 2 \times [m/n] + 1$ called as the average number of edges of a node then constructed randomly a spanning tree of n = |V| and n - 1 edges so that all node degrees did not exceed r, we next

inserted randomly other m - (n - 1) valid edges and assured that every node degree did not exceed r, edge weights in the graph were generated randomly in range [1..2500].

| Test | WONG | ADD | CAMPOS | H1 | H2 |
|------|---------|----------|---------|---------|---------|
| 31 | 166232 | 441740 | 174190 | 165052 | 165052 |
| 32 | 242436 | 1110048 | 262110 | 242708 | 241096 |
| 33 | 552034 | 2330760 | 658246 | 547204 | 547204 |
| 34 | 754534 | 5906702 | 852214 | 751180 | 751180 |
| 35 | 1591536 | 11757760 | 2015229 | 1586588 | 1586588 |
| 36 | 1931829 | 14010963 | 2036603 | 1921256 | 1921256 |
| 37 | 193600 | 1386026 | 242028 | 193492 | 193492 |
| 38 | 7111108 | 36043876 | 7705378 | 7068811 | 7087501 |
| 39 | 2308062 | 10356986 | 2735218 | 2297704 | 2297704 |
| 40 | 5640618 | 19951434 | 7332050 | 5632972 | 5632972 |
| 41 | 935328 | 3294241 | 1271022 | 935328 | 935328 |
| 42 | 2259932 | 8717570 | 2813048 | 2243368 | 2246560 |
| 43 | 1408134 | 5740514 | 1472156 | 1408134 | 1408134 |
| 44 | 3157851 | 12159738 | 3625373 | 3147656 | 3147656 |
| 45 | 660361 | 3617234 | 790670 | 659906 | 654767 |

When comparing H₁ and H₂ over against WONG, ADD, CAMPOS through 15 tests of graphs under edge distribution, we obtained result as Table III B.

TABLE III B: EXPERIMENTAL RESULTS

| H1 | WONG | | AD: | ADD | | CAMPOS | |
|------------|----------|----|----------|-----|----------|--------|--|
| пі | Quantity | % | Quantity | % | Quantity | % | |
| Better | 12 | 80 | 15 | 100 | 15 | 100 | |
| Equivalent | 2 | 13 | 0 | 0 | 0 | 0 | |
| Worse | 1 | 7 | 0 | 0 | 0 | 0 | |
| H2 | Quantity | % | Quantity | % | Quantity | % | |
| Better | 13 | 87 | 15 | 100 | 15 | 100 | |
| Equivalent | 2 | 13 | 0 | 0 | 0 | 0 | |
| Worse | 0 | 0 | 0 | 0 | 0 | 0 | |

E. Non-Uniform Edge Distribution

Generating graph under non-uniform edge distribution Graphs with non-uniform edge distribution G = (V,E,w) were generated as follows: we first selected random k nodes $([n/2] + 1 \le k \le n-1)$ then assigned each node an integer $r \in \{1,2\}$ indicates that node degrees cannot be bigger than r; we next constructed a spanning tree and its edges as we did in general graphs. Note that in the case of not being able to construct enough m edges, we would repeat the process (this kind of graph is named as asymmetric graph).

TABLE IV A: Non-Uniform Edge Distribution

| Test | WONG | ADD | CAMPOS | H1 | H2 |
|------|---------|---------|---------|---------|---------|
| 46 | 2381866 | 3403130 | 2636460 | 2360598 | 2360598 |
| 47 | 3683138 | 6418194 | 3911224 | 3667189 | 3667189 |
| 48 | 199826 | 388175 | 206870 | 195836 | 195836 |
| 49 | 291837 | 518199 | 329238 | 291457 | 291457 |
| 50 | 1264912 | 1843978 | 1276668 | 1253872 | 1252822 |
| 51 | 2606062 | 4779404 | 2761861 | 2603768 | 2603768 |
| 52 | 1544662 | 3076550 | 1729774 | 1540650 | 1540650 |
| 53 | 553184 | 942300 | 619302 | 552906 | 552906 |
| 54 | 1572986 | 2884650 | 1684184 | 1571468 | 1571468 |
| 55 | 2338095 | 3819570 | 2463925 | 2323776 | 2323865 |
| 56 | 713830 | 1193228 | 822416 | 706892 | 706892 |
| 57 | 515746 | 1029195 | 565535 | 506533 | 506533 |
| 58 | 203599 | 485323 | 219841 | 203599 | 203599 |
| 59 | 942060 | 1674603 | 1023093 | 939226 | 939226 |
| 60 | 295842 | 552842 | 327948 | 293685 | 294642 |

When comparing H₁ and H₂ over against WONG, ADD, CAMPOS through 15 test of graphs under non-uniform distribution, we obtained result as Table IV B.

TABLE IV B: EXPERIMENTAL RESULTS

| H1 | WONG | | ADD | | CAMPOS | |
|------------|----------|----|----------|-----|----------|-----|
| пі | Quantity | % | Quantity | % | Quantity | % |
| Better | 14 | 93 | 15 | 100 | 15 | 100 |
| Equivalent | 1 | 7 | 0 | 0 | 0 | 0 |
| Worse | 0 | 0 | 0 | 0 | 0 | 0 |
| H2 | Quantity | % | Quantity | % | Quantity | % |
| Better | 14 | 93 | 15 | 100 | 15 | 14 |
| Equivalent | 1 | 7 | 0 | 0 | 0 | 1 |
| Worse | 0 | 0 | 0 | 0 | 0 | 0 |

F. Summarized Table

When comparing H_1 and H_2 over against WONG, ADD, CAMPOS through 60 tests in total, we obtained result as Table V.

TABLE V: SUMMARIZED IN GRAPHS (60 TEST)

| H1 | WONG | | ADD | | CAMPOS | |
|------------|----------|----|----------|-----|----------|-----|
| пі | Quantity | % | Quantity | % | Quantity | % |
| Better | 42 | 70 | 55 | 92 | 60 | 100 |
| Equivalent | 4 | 7 | 0 | 0 | 0 | 0 |
| Worse | 14 | 23 | 5 | 8 | 0 | 0 |
| H2 | Quantity | % | Quantity | % | Quantity | % |
| Better | 55 | 92 | 60 | 100 | 60 | 100 |
| Equivalent | 5 | 8 | 0 | 0 | 0 | 0 |
| Worse | 0 | 0 | 0 | 0 | 0 | 0 |

V. CONCLUSION AND FUTURE WORKS

We have proposed 2 heuristic-based algorithms to solve MRCT problem, we have also compared the results of these algorithm with the results of algorithms WONG, ADD, CAMPOS and the obtained results are possitive.

We are now studying metaheuristic-based algorithms for MRCT problem, and the idea of improving a spanning tree step by step (neighbor exploring) like above 2 algorithms would be main idea of metaheuristic-based.

REFERENCES

- [1] R. Wong, Worst-case analysis of network design problem heuristics, SIAM J. Algebra. Discr, 1:51–63, 1980.
- [2] Alan W Johnson, Generalized Hill Climbing Algorithms For Discreter Optimization Problems, 1996.
- [3] Bang Ye Wu, Kun-Mao Chao, Spanning Trees and Optimization Problems, Chapman&Hall/CRC, 2004, 13–139.
- [4] V. Grout, Principles of cost minimization in wireless networks, Journal of Heuristics 11 (2005), 115–133.
- [5] Lin Lin, Mitsuo Gen, Node-Based Genetic Algorithm for Communication Spanning Tree Problem, IEICE TRANS. COMMUN, vol.E89–B, no.4 APRIL 2006, 1091-1098.
- [6] Rui Campos, Manuel Ricardo A fast Algorithm for Computing Minimum Routing Cost Spanning Trees, Computer Networks, vol. 52, Issue 17, 2008, 3229-3247.