International Journal of Engineering and Technology Development
Vol.1, No.2, August 2013, ISSN 2337-3180(Online)

IJEST
www.journal.ubl.ac.id

The Modified CW1 Algorithm For The Degree Restricted Minimum Spanning Tree Problem

Wamiliana[#1] and Louis Caccetta[#2]
[1#]*Dept. of Mathematics, Faculty of Mathematics and Natural Science, Lampung University, Indonesia*
[2#]*Dept. of Mathematics, Curtin University of Technology, Australia*
wamiliana@unila.ac.id

(Received August 2012, Accepted May 2013)

Abstract— ***Given edge weighted graph G (all weights are non-negative), The Degree Constrained Minimum Spanning Tree Problem is concerned with finding the minimum weight spanning tree T satisfying specified degree restrictions on the vertices. This problem arises naturally in communication networks where the degree of a vertex represents the number of line interfaces available at a terminal (center). The applications of the Degree Constrained Minimum Spanning Tree problems that may arise in real-life include: the design of telecommunication, transportation, and energy networks. It is also used as a subproblem in the design of networks for computer communication, transportation, sewage and plumbing. Since, apart from some trivial cases, the problem is computationally difficult (NP-complete), a number of heuristics have been proposed. In this paper we will discuss the modification of CW1 Algorithm that already proposed by Wamiliana and Caccetta (2003). The results on540 random table problems will be discussed.***

Keywords— Minimum spanning tree; CW1 Algorithm ; Degree constrained

I. INTRODUCTION

Typically, the DCMST can be applied in cases where n vertices (or terminals/servers/road intersections) need to be connected with a minimum length of an underlying transportation mode (wires, pipes, canals or roads). However, the handling capacity of each of the vertices imposes a restriction on the number of edges (or wires/roads) that can be connected to a vertex. The DCMST may be used in the design of the road system, which has to serve a collection of suburbs/towns, and has the additional restriction that no more than certain number of roads (example: four roads) are allowed to meet at an intersection. A degree constraint in a communication network also limits the liability in the case of vertex failure. In computer networks, the degree restrictions can be used to cater for the number of line interfaces available at a server/terminal [19].

Garey and Johnson [7] showed that, apart from some trivial cases, the DCMST problem is computationally difficult (NP-complete) by reducing it to an equivalent symmetric Traveling Salesman Problem (TSP). Notice that if the degree bound $b_i = 2$, $\forall i \in V$, the problem reduces to a TSP. Thus, it is unlikely a polynomial bounded algorithm exists for solving general DCMST problems.

In this paper we will discuss the comparative modification of CW1 algorithm that already proposed by Wamiliana and Caccetta [18] to solve the DCMST problem. This paper is organized as follows: Section 2 briefly reviews some of the solution methods available in the literature; Section 3 discusses about modification we made from CW1 algorithm, Section 4 shows the implementation and in Section 5 derives the conclusion.

II. METHODS AVAILABLE IN LITERATURE

The DCMST problem has been considered by a number of authors and both heuristic and exact methods have been proposed. We give a brief account of some of this work below.

For heuristics, many variations of the Prim's and Kruskal's algorithms have been developed, for example, by Narula and Ho [12]. A Genetic Algorithm was proposed by Zhou and Gen [20]. They use the Prufer [13] number to uniquely code the spanning tree. In the method they adopt uniform crossover and perturbation mutation operators as the genetic operators, and tested the algorithm on problems with up to 50 vertices.

Simulated Annealing was proposed by Krishnamoorthy et al. [11]. Further, they also proposed a hybrid method called Problem Space Search, which is a blend between the Genetic Algorithm approach and a simple constructive search method. The algorithms were implemented and tested on problems with 30, 50, 70 and 100 vertices.

Boldon et al. [1] and Deo and Kumar [6] proposed an Iterative Refinement Method. In this method, the construction starts with finding a MST and then the edges incident to a degree violated vertex are penalized, except the smallest one. With the new weighted edges, the process of calculating a MST is repeated, and it continues until a spanning tree without degree violation is found. This method was implemented using parallel computing on a computer with 8192 processors. This can be done because the nature of the algorithm/method, where every vertex can be assigned a processor and the computational process of penalizing edges is independent (non sequential). Problems with up to 5934 vertices were solved.

Caccetta and Wamiliana [3] and Wamiliana [17], proposed Modified Penalty Methods (MP1 and MP2) as variants of Iterative Refinement methods. Implemented on some benchmark problems, the methods perform better than Simulated Annealing method.

Wamiliana and Caccetta [18], [19] proposed Tabu Search method for solving the DCMST problem. They solved up to 2160 problems with n ranging from 10 to 500. The methods are quite competitive.

Exact methods include branching algorithms and the Lagrangean relaxation procedure. The branch and bound method for solving the DCMST problem has been investigated by Narula and Ho [12],Savelsbergh and Volgenant [15], and Volgenant [16].

Narula and Ho [12] used a branching procedure which is an adaptation of the method due to Held and Karp [9, 10] for the traveling salesman problem. They solved Euclidean and random table problems with up to 100 vertices.

Savelsbergh and Volgenant [15] used 2 heuristics (AH and CH) in a branch and bound method. The heuristic AH (Analysis Heuristic) is based on the edge exchange analysis, and the CH heuristic is a generalization of the one used by Volgenant and Jonker[15] for the traveling salesman problem. The CH heuristic is related to a heuristic developed by Christofides [5]. Both heuristics are used once in each subset of the branch and bound tree. This branch and bound method was implemented and tested on Euclidean and random table problems with up to 70 vertices.

The application of the Lagrangean Relaxation method has been investigated by Gavish[8] and Volgenant[16].Gavish [8] solved Euclidean problems with up to 200 vertices. Volgenant [16] in addition to using Lagrangean relaxation also used the ascent procedure to define the value of the multiplier. He solved Euclidean and random table problems with up to 150 vertices.

Caccetta and Hill [2] proposed a method based on Branch and Cut method. The relaxed LP subproblems are solved using the CPLEX package. The violating constraints of type (3) and the connectivity constraints are found by two search procedures, one a local search and the other a global search. They used depth-first search strategy in the branching and the best bound found so far is updated using the standard sensitivity analysis procedure. They tested their algorithms to 3150 random table problem with n ranging from 100 to 800.

2.The modified CW1 algorithm.

As in Wamiliana and Caccetta [18], the CW1 algorithm starts by first finding the MST. This gives us a lower bound (LB) whilst The Modified Kruskal algorithm gives the initial feasible solution, which is Degree Constrained Spanning Tree (DCST), and also acts as an upper bound (UB). CW1 starts from the upper bound, which is feasible and work towards optimality. The moves are the set of edges that are incident with the leaves (vertices of degree 1) in the $G\backslash T$. Tabu tenure is set to be 0.1 n, where n is the number of vertices in the graph. The maximum number of iterations is $0.2n$. The stopping criteria are the *tolerance* and maximum number of iterations, where *tolerance* = 1 % of *gap* (*gap* = UB − LB). Note UB is revised as better feasible solutions are obtained.

The aspiration condition is applied if a degree violation is detected. All possible edge exchanges among the edges of T incident to the violated vertex i and the edges of G not in T involving the neighbors of i, are examined. If the searching doesn't yield a better solution, then we record the current best solution, put the currently used moves into tabu status and restarted again.

Modified CW1 algorithm (MCW1) in general uses the same terminology as the basic algorithm CW1. In this algorithm all steps in CW1 are adapted with some slight modifications, where the two initial basics feasible solutions are generated using Modified Prim and Modified Kruskal. The initial basic feasible solution that has the best quality solution will be chosen first as the upper bound. Then, after a certain number of iterations, if the search could not gain a solution within the tolerance specification, we restart the process and use the other feasible solution generated. The best solution is recorded.

In addition to changing the way of finding the initial basic feasible solutions, MCW1 modifies CW1 by introducing comparative routine as follow:

begin
Put the recently used edges in the set Tabu move.
Check the Tabu tenure.
if
The requirement satisfies, remove first two elements
fromTabumove
else
continue.
if
The $T_{obj} \leq T_{controlobj}$,
set$T$ as $T_{control}$ and T_{obj} as $T_{controlobj}$.
Put the recently used edges in Tabu tenure.
Goto the main algorithm
else
Put recent used edges in Tabu tenure
continue.
if
The number of iteration is ≥ number for restart with other solution,
Keep the current solution , print thesolution,
Set $T_{control} = T(DIV)$, $T_{controlobj} = T(Div)_{obj.}$
Increase the iteration number by 1
Go to the main algorithm
else continue.
Select the next move.
If
The move is empty, remove i from vertex list,
Check the vertex list.
If vertex list is empty, give message, " nothing can be improved".
end
else (vertex list $\neq \varnothing$),
Increase the iteration number by 1
Do the Moves Selection Strategy Routine in CW1 main algorithm
end
end

III. COMPUTATIONAL RESULTS.

We implemented our heuristic using the C programming language on a Silicon Graphic Indy machine, running in 150MHz. In the implementation we do make the assumption that the degree restriction for every vertex is the same.

For all vertex orders we run the program using the gap value of 1 % and maximum iteration number as min {0.20n, 50}. For the degree condition, we restrict our implementation only for degree bound 3. We choose this bound, since our early computational work revealed that for degree bound greater than 3 the MST is usually feasible and hence optimal. We provide results on 2160 random problems generated as follows:

- Number of vertices range from 10 to 500 with an increment of 10 for up to 100vertices and an increment of 50 for larger graphs.
- The edge weights are generated Randomlyfrom uniform distribution from 1 to 1000.
- For a given n, graphs are generated withdensity p = 1 which mean that we use complete graph of order n.
- For a given n, 30 random problems are generated.

Test Data

For all simulation problems, first we generate 30 problems for every vertex order. We use time as the seed when generating a problem (data) and assign that data a name so that next time when we will retest, we use the same data. This is very important step because otherwise we will lose the same data since our seed is time, which will never be the same.

The following tables detail the computational results for MCW1 algorithms. The average performance of MCW1 improves the results of CW1, in terms of the statistic $\frac{H - LB}{LB}$ by approximately 0.3 %. The following tables and figures detail the results.

TABLE 1.
THE PERFORMANCE OF CW1 FOR THE GRAPHS WITH 10 TO 100 VERTICES WITH INCREMENTS IN 10, P = 1,TOLERANCE 1%, $B_I = 3 \ \forall I$

Number of vertices	Average (MK-LB)/LB	(CW1-LB)/LB
10	0.131578	0.066067
20	0.066052	0.057933
30	0.076046	0.067111
40	0.070054	0.063430
50	0.073348	0.061223
60	0.075702	0.067070
70	0.087226	0.074880
80	0.075585	0.070758
90	0.085593	0.077978
100	0.074109	0.065123

TABLE 2.
THE PERFORMANCE OF CW1 FOR THE GRAPHS WITH 50 TO 500 VERTICES WITH INCREMENTS IN 50, P = 1, TOLERANCE 1%, B_I = 3 $\forall I$

Number of vertices	Average (MK-LB)/LB	(CW1-LB)/LB
50	0.073348	0.061223
100	0.074109	0.065123
150	0.080862	0.065292
200	0.077879	0.066724
250	0.091188	0.074507
300	0.085995	0.074507
350	0.085866	0.074589
400	0.088672	0.074669
450	0.090023	0.083421
500	0.087555	0.081576

TABLE 3.
THE PERFORMANCE OF MCW1 FOR THE GRAPHS WITH 10 TO 100 VERTICES WITH INCREMENTS IN 10,P = 1, TOLERANCE 1%, $B_I = \forall I$

Number of vertices	Average (MK-LB)/LB	(CW1-LB)/LB
10	0.098559	0.0657
20	0.066052	0.0538
30	0.076046	0.0651
40	0.071673	0.0616
50	0.073348	0.0568
60	0.075702	0.0657
70	0.087226	0.0732
80	0.075585	0.0693
90	0.085593	0.0753
100	0.074109	0.0622

TABLE 4.
THE PERFORMANCE OF MCW1 FOR THE GRAPHS WITH 50 TO 500 VERTICES WITH INCREMENTS IN 50, P = 1, TOLERANCE 1%, B_I = 3 $\forall I$

Number of vertices	Average (MK-LB)/LB	(CW1-LB)/LB
50	0.073348	0.0568
100	0.074109	0.0622
150	0.080862	0.063076
200	0.077879	0.063765
250	0.091188	0.073438
300	0.085995	0.072527
350	0.085866	0.073528
400	0.088672	0.072438
450	0.090023	0.07991
500	0.087555	0.07953

IV. CONCLUSION

In terms of effectiveness of the Tabu Search aspects of the algorithm we note that on average CW1 improves the initial upper bound in terms of the statistic $\frac{H-LB}{LB}$ by approximately 1.5%. However, the computational results show that employing a different initial feasible solution improves the quality of the solution. The MCW1 algorithm improves the CW1 results, in terms of the statistic $\frac{H-LB}{LB}$ by an average of approximately 0.3% with the highest improvement of 0.5% occurring at n=50.

REFERENCE

[1] Boldon, B., N. Deo and N. Kumar, Minimum Weight degree-constrained spanning tree problem: Heuristics and Implementation on an SIMD parallel machine. *Parallel Computing*, **vol**. **22**, 369 – 382, 1996.

[2] Caccetta, L. and S.P. Hill, A Branch and Cut method for the Degree Constrained Minimum Spanning Tree Problem, *Networks,* **vol. 37**, pp.74-83, 2001.

[3] Caccetta, L. and Wamiliana, Heuristics Approach For The Degree Constrained Minimum Spanning Tree, Proceeding of The International Modeling and Simulations, pp. 2161-2166, Canberra, 2001.

[4] Caccetta L. and Wamiliana,2004. "A First Level Tabu Search Method for the DegreConstrained Minimum Spanning Tree", *Journal of Quantitative Methods*, Vol.1 No.1, pp. 20-32.

[5] Christofides, N., *Worst-case Analysis of a New Heuristic for the TravelingSalesman Problem*, Carnegy-Mellon University, Pittsburgh, USA, 1976.

[6] Deo N. and N. Kumar, Computation of Constrained Spanning Trees: A Unified Approach. *Network Optimization* (Lecture Notes in Economics and Mathematical Systems, Editor: Panos M. Pardalos, et al.), Springer-Verlag, Berlin, Germany, pp. 194 – 220, 1997.

[7] Garey, M.R., and D.S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*. Freemann, San Francisco, 1979.

[8] Gavish, B., Topological Design of Centralized Computer Networks Formulations and Algorithms. *Networks* **vol. 12**, pp.355 –377, 1982. pp.111 – 134,1995.

[9] Held, M. and R.M. Karp. The traveling salesman problem and minimum spanning trees. *Operation Research*, **vol**. **18,** pp.1138-1162, 1970.

[10] Held, M. and R.M. Karp. The traveling salesman problem and minimum spanning trees. Part II. *MathematicalProgramming*, **vol.1,** pp.6-25, 1971.

[11] Krishnamoorthy, M., A.T. Ernst, and Y.M., Sharaiha Comparison of Algorithms for the Degree Constrained Minimum Spanning Tree. *Journal of Heuristics*, **Vol. 7** no. 6, pp. 587 – 611, 2001.

[12] Narula,S.C., and C.A. Ho , Degree-Constrained Minimum Spanning Tree. *Computer and Operation Research* ,**vol.7**, pp.239-249, 1980.

[13] Prufer, H., NeuerbeweiseinessatzesüberPermutationen. *Arch. Math. Phys*, **vol. 27**, pp.742-744, 1918.

[14] Savelsbergh,M., and T. Volgenant, Edge Exchange in The Degree- Constrained Minimum Spanning Tree. *Computer and Operation Research***12**, pp.341-348, 1985.

[15] Volgenant A., and R. Jonker, A branch and bound algorithm for the traveling salesman problem based on the 1–tree relaxa*European Journal of Operational Research*, **vol. 9**, pp. 83-89, 1982

[16] Volgenant, A., A Lagrangean Approach to TheDegree-Constrained Minimum Spanning Tree Problem. *European Journal Of Operational Research***vol. 39**, pp.325 - 331, 1989.

[17] Wamiliana, The Modified Penalty Methods for The Degree Constrained Minimum Spanning Tree Problem, JurnalSainsdanTeknologi, pp.1-12, **Vol. 8,** 2002.

[18] Wamiliana and Caccetta, 2003. "Tabu SearchBased Heuristics for the Degree Constrained Minimum Spanning Tree Problem", Proceeding of South East Asia Mathematical Society Conference, pp. 133-140.

[19] Wamiliana, 2004. Combinatorial Methods for TheDegree Constrained Minimum Spanning Tree Problem, Doctoral Thesis, Dept. of Mathematics and Statistics, Curtin University of Technology.
[20] Zhou, G., and M. Gen, A Note on Genetic Algorithms for Degree Constrained Spanning Tree Problems. *Network* **vol. 30**, pp.91 – 95, 1997.