VOL. 10, NO 21, NOVEMBER, 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

SOME GREEDY BASED ALGORITHMS FOR MULTI PERIODS DEGREE CONSTRAINED MINIMUM SPANNING TREE PROBLEM

Wamiliana[1], Faiz A. M. Elfaki[2], Mustofa Usman[1] and M. Azram[2]
[1]Department of Mathematics, Faculty of Mathematics and Natural Science, University of Lampung, Indonesia
[2]Department of Sciences, Faculty of Engineering, IIUM, Kuala Lumpur, Malaysia
E-Mail: faizelfaki@yahoo.com

ABSTRACT

In Indonesia, the fund for a project or an activity is usually divided into terms which can be two or three terms. Therefore, in order to optimize the utility, the team of the project need to design the project so that the project can be used as soon as possible without violating the restriction given by the government. The Multi Periods Degree Constrained Minimum Spanning Tree Problem (MPDCMST) is a problem that concerns about finding a minimum networks installations for a certain commodity so that the networks does not violate the reliability restriction whilst also satisfying the fund limitation in every stages of installations. In this paper we proposed three algorithms by modifying Kruskal's and Prim's algorithms. We implemented our algorithms on 300 generated problems with vertex order ranging from 10 to 100, and compared them with those that were already in the literature. The result shows that the algorithms proposed perform better.

Keywords: multi periods, degree constrained, installations, networks.

1. INTRODUCTION

Until very recently graph theoretic concept has been proven to be useful as a tool for representing problems that arises in daily life. Fueled by the fast improvement of computer technology, theory graph gives a good impact in the development of science and technology. The graph G(V, E) can be viewed as a network where vertices can represent cities, computers, houses, station, etc., while the edges of the graph can represent roads, cables, canals, pipes, train track, etc. Therefore, in designing or representing a network, one can just use graph as a tool.

Given a connected weighted graph G(V, E), where every edge eij associated with a cost $c_{ij} \geq 0$, one fundamental problem that commonly arises is to find a minimum cost subgraph of G(V, E) that connects every vertex of the graph. Indeed, that subgraph must constitute a spanning tree of G(V, E), i.e the problem is to find the minimum spanning tree (MST) of graph G(V, E). There are two well known algorithms for solving the MST, Prim's and Kruskal' algorithms, but the first algorithm for solving MST was proposed by Boruvka in 1926 [1].

One problem that used MST as a backbone is Degree Constrained Minimum Spanning Tree (DCMST) Problem. The Degree Constrained Minimum Spanning Tree (DCMST) typically arises in the design of telecommunication, transportation and energy networks. It is concerned with finding a minimum-weight (distance or cost) spanning tree that satisfies specified degree restrictions on its vertices. The degree restrictions typically represent the capacity of a center (node) in the network.

The problem is, apart from some trivial cases, computationally difficult (NP-complete) [2].

The constraint on the degree restriction occurs when one needs to know the reliability of the networks, for example networks connection. The degree constrained restricts the number of computer interfaces (in computer networks, for example). 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. [3], for example, used the DCMST as a subproblem in the design of a centralized computer network; and [3] also provides several examples of optimization problems that are faced in the process of designing computer communication networks.

The DCMST problem had been highly investigated and both exact and heuristics algorithms already proposed, for example exact methods had been investigated included Langrangean relaxation by [3] and [4], branch and bound by [4, 5, 6], and the branch and cut method by [7], while heuristics included many variations of Prim's and Kruskal's algorithms [5]. Genetic Algorithm by [8], Simulated Annealing [9, 10], iterative refinement by [11, 12], Modified Penalty by [13, 14], Tabu Search by [15, 16, 17].

The Multi Period Degree Constrained Minimum Spanning Tree (MPDCMST) Problem was introduced by [5]. They adopt hybrid methods between Lagrangean Relaxation and branch exchange, and used 10-year planning horizon and the time period for activating each terminal is uniformly distributed. They implemented the method using graphs with order varying from 40 to 100 vertices.

[18] proposed modification of MPDCMST problem by changing the planning horizon only one year and divided the installation process into three periods (four months each). This modification made due to real condition in Indonesia where the fund for a project usually divided into two or three terms of payments. [19] Improved the algorithm developed by [18] and tested the algorithm on some problems taken from TSPLIB. [20] Improved the algorithm developed by Junaidi by adopting m path, $1 \leq m \leq 2$. They implemented the algorithm on a data set used by [16]. [17] Improved the algorithms proposed in [20] by changing the $MaxVT_i$ and HVT_i value, and set $1 \leq m \leq 3$.

We will organize our paper as follow: in section 1, Introduction, we briefly discuss the background and history of the Multi Period Degree Constrained Minimum Spanning Tree Problems. In section 2 we discuss about the problem considered, in section 3 we discuss about the algorithms, and in section 4 we show the results followed by Conclusion.

2. THE PROBLEMS

The Multi Period Degree Constrained Minimum Spanning Tree (MPDCMST) is an enhanced problem of Degree Constrained Minimum Spanning Tree (DCMST). In the Degree Constrained Minimum Spanning Tree (DCMST) problem we concerned of finding a minimum-weight spanning tree whilst satisfying degree requirements on the vertices. 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 [14]. The DCMST also used as a subproblem in the design for computer network, communication network, transportation network, and so on. [21], for example, used the DCMST as a subproblem in the design of a centralized computer network; and [3] also provides several examples of optimization problems that are faced in the process of designing computer communication networks. In DCMST Problem we concern of finding an MST that satisfy degree restriction on it vertices. The MILP formulation for DCMST is given below by [14]:

$$Minimise \quad \sum_{i}^{n}\sum_{j}^{n} c_{ij}x_{ij} \tag{1}$$

subject to

$$\sum_{i,j} x_{ij} = n-1 \tag{2}$$

$$\sum_{i,j\in V'} x_{ij} \leq |V'|-1, \quad \forall \varnothing \neq V' \subseteq V \tag{3}$$

$$1 \leq \sum_{j=1,i\neq j} x_{ij} \leq b_i \qquad i = 1,2,...,n \tag{4}$$

$$x_{ij} = 0 \text{ or } 1, \qquad 1 \leq i \neq j \leq n. \tag{5}$$

where c_{ij} is the weight (or distance or cost) of the edge (i,j), b_i is the degree bound on vertex i and n is the number of vertices. Constraint (2) ensures that $(n-1)$ edges are selected. Constraint (3) is the usual subtour elimination constraints. Constraint (4) specifies the degree restriction on the vertices. The last constraint (5) is just the variable constraint, which restricts the variables to the value of 0 or 1. x_{ij} is 1 if the edge x_{ij} is selected or included in the tree T and 0, otherwise.

In the DCMST problem, all vertices are connected in one period, there is no priority level for specific vertices. In MPDCMST, the process of connecting/installing vertices is devided into periods that implies that at a certain period only part of the set of vertices already connected, except the last period where all vertices must be connected. Therefore the MPDCMST problem that we consider can be stated as follow:

Given graph G(V, E) with every edge of E has associated with a weight cost c_{ij}, k is the number of periods, find the minimum cost of connecting all the vertices in the graph that satisfy two conditions which are:

a) every vertex v_i has property that $d(v_i) \leq b_i$

b) every element on HVT_k must be installed in the k^{th} period or before.

HVT_k is set of vertices that must be installed in k^{th} period or before, and $|HVT_k|$ is the number of element in the set HVT_k. Be noted that, if $|HVT_k| > MaxVT_k$ (the maximum number of vertices that can be installed on k^{th} period), then the problem becomes infeasible.

There are two very well-known algorithms for finding a MST of a graph, Prims's and Kruskal's algorithms. Krukal's algorithm was proposed by [22], and Prim's algorithm proposed by [23]. These two algorithms not only can be used for finding Minimum Spanning Tree (MST) Problem as they are supposed to be, but also can be used as tools that can be modified to solve problems using MST as a backbone such as Degree Constrained Minimum Spanning Tree (DCMST) problems, Multi Period Degree Constrained Minimum Spanning Tree (MPDMST) Problems, Bounded Diameter Minimum Spanning Tree Problems, Most Reliable Minimum Spanning Tree Problems, and many more.

In this section we discuss about WDAR1 and WADR2 algorithms that already propose by [20]. The reason for doing this because our algorithms developed based on these two algorithms, and we compare the algorithms proposed with those two algorithms. In these algorithms they used one year planning horizon and divided the period of payments every four months (three periods). T is the tree formed after installations, V as the set of vertices in Tree T, and L(T) is the total cost. MaxVTi is the maximum number of vertices that can be installed (connected to the main networks) on ith period. We use MaxVTi = $\left\lceil \frac{n-1}{3} \right\rceil$ as in [17], and at the end of installation $\sum$MaxVTi = n-1. HVTi is the set of vertices that must be installed before or on ith period, $|$HVTi$|$<MaxVTi.

The following pseudocode shows WDAR1 algorithm as proposed by [20].

```
begin
        Determine vertices in HVT_i
        Sort edges connected with vertices in HVT_i
        whosepath length k in ascending order
        Set i = 1
        if | HVT_i | ≥ | MaxVT_i | , Stop
        else do
        while { the number of edges in T is < n -1}
        Choose the smallest edge in the sorting and
connect it with T
          if the connection of that edge constitutes cycle
           Remove the edge and choose the next available
        If the connection violates the degree restriction
        Remove and choose the next available edge
                         end
                 end
             end
        end
        i = i + 1
       if i > 3
    end
 end
```

3. THE ALGORITHMS

We develop two algorithm based on WDAR1, WDAR2, and named them as WDAR3 and WDAR4 by relaxing the HVTi in the installation process. In WDAR1 and WDAR2, if a certain vertex v_i already connected/installed in the previous period, then the program will ask the user what other vertices should be installed, maybe due to its importance, because there is possibility to installed more vertices if MaxVTi > HVTi. In WDAR3 and WDAR4 we relax that condition so that the program will choose the best possibility in the process of connecting/installing the vertices. Moreover, in WDAR4, we change the searching process by not selecting the next vertex with minimum edge to be installed, but searching for the best 3- path that connect to the temporary networks to be installed.

In addition to those of Modified Kruskal's algorithm, we develop WDAR5 algorithm based on other greedy algorithm, which is Prim's algorithm, and use the same terminology as in WDAR's algorithms. The main difference between WDAR5 and those of Kruskal's modifications lies in the installation process where in the WDAR5, the networks always constitute only one tree, not a forest, while in others, that is possible to constitute a forest during the installation process, but at the end the networks constitute a tree. Table-1, show the characteristics of the algorithms developed

4. DATA FOR IMPLEMENTATION

We use the data set used by [19], and [20]. The data generated as follows:

- Number of vertices range from 10 to 100 with an increment of 10 for up to 100.
- The edge weights are generated randomly from uniform distribution from 1 to 1000.
- For a given *n*, graphs are generated with different density p. In this paper we generate data with p = 1 (complete graph)
- For a given *n*, 30 random problems are generated.

5. RESULTS

We implemented our heuristic using the C++ programming language running on dual core computer, with 1.83 Ghz and 2 GB RAM. We do make the assumption that the degree restriction for every vertex is the same, which are 3. For the degree constrained, we do restrict our implementation only for degree bound 3. We chose this degree bound, since our early computational work revealed that for degree bound greater than 3 the MST is usually feasible for DCMST, and hence optimal for DCMST problem. We developed the source code in such a way so that the user can give input on HVTi in every period. As we know HVT_i consists of set of vertices that must be installed before or on the ith period. This set of vertices can be important for society such as hospital, fire station, etc. Suppose that the network is energy networks (electricity) or water, then a hospital must get priority to be installed (connected to the main network) first. Therefore, the setting user input for HVT_i in the source code makes the program as a tool in hand for the user.

Figure-1, show the results on our algorithms (WADR3, WADR4, and WDAR5) and compare them to the algorithms proposed by [20].

6. CONCLUSIONS

From the diagram we can see that the solution of WDAR1 and WDAR2 are far beyond the lower bound of the problem (DCMST). Performance of WDAR3, WDAR4 and WDAR5 are better than WDAR1 and WDAR2, but among all of the algorithms, WDAR4 is the best. Therefore, if we concern that during installation process the vertices already installed still connected in one component, then we choose WADR5, but if we concern of installing a multi period network in which there is no priority of the vertices should be installed in every period, then we should choose WDAR4. In general we can conclude that using the best -3 path and randoming the edges installed in every period performs the best.

Table-1. The characteristics of the algorithms developed.

No.	Name	Algorithms' characteristics				
		Based algorithm	**Start searching**	**Period**	**Edge searching**	**Installation process of HVTi**
1	WADR1	Kruskal	After HVTi connected/ installed	I & II	The minimum edge in the sorting list	If a certain vertex in HVTi already installed in the previous period, the program will ask the next priority vertex to be installed.
2	WADR2	Kruskal	After HVTi connected/ installed	I & II	The minimum edge in the sorting list	If a certain vertex in HVTi already installed in the previous period, the program did not ask the next priority vertex to be installed.
3	WADR3	Kruskal	Not depend on HVTi	I & II	The minimum edge in the sorting list	If a certain vertex in HVTi already installed in the previous period, the program did not ask the next priority vertex to be installed.
4	WADR4	Kruskal	Not depend on HVTi	I & II	Use the best 3-path	If a certain vertex in HVTi already installed in the previous period, the program did not ask the next priority vertex to be installed.
7	WADR5	Prim	Not depend on HVTi	I, II & III	Use the best 3-path	If a certain vertex in HVTi already installed in the previous period, the program did not ask the next priority vertex to be installed.

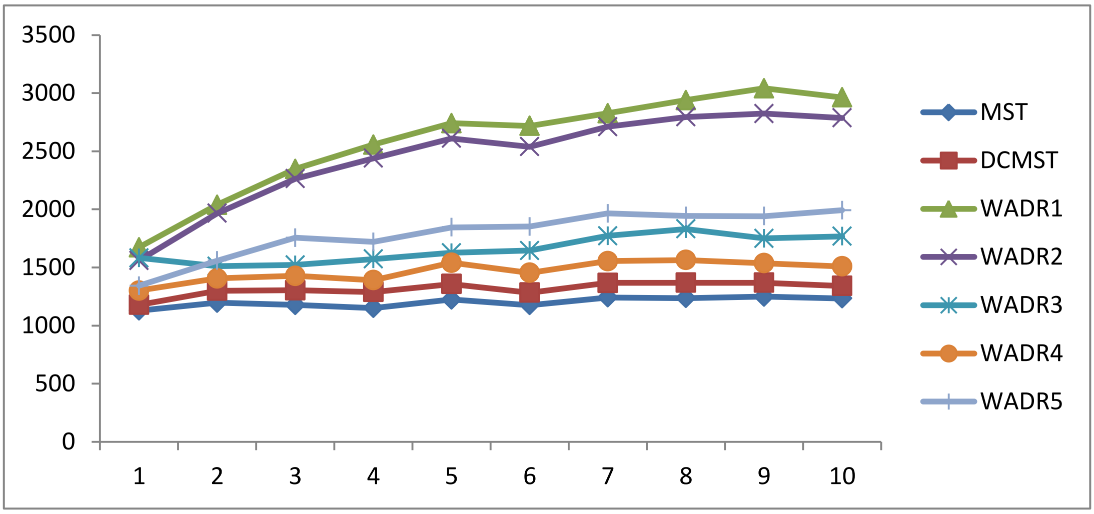

Figure-1. Compare WDARA1, 2, 3, 4, 5, DCMST and MST.

REFERENCES

[1] Wamiliana. 2004a. Combinatorial Methods for Degree Constrained Minimum Spanning Tree Problem, Doctoral Thesis, Department of Mathematics and Statistics, Curtin University and Technology, Australia.

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

[3] Gavish, B. 1992. 'Topological Design of Centralized Computer Networks the Overal Design Problem ', European Journal of Operations Research, vol. 58, pp.149-172.

[4] Volgenant, A. 1989. 'A Lagrangean Approach to The Degree-Constrained Minimum Spanning Tree Problem'. European Journal of Operational Research 39, pp. 325-331.

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

[6] Savelsbergh,M., and T. Volgenant. 1985. 'Edge Exchange in The Degree- Constrained Minimum Spanning Tree'. Computer and Operation Research 12, pp.341-348.

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

[8] Zhou, G., and M. Gen. 1997. 'A Note on Genetic Algorithms for Degree Constrained Spanning Tree Problems'. Network, Vol. 30, pp. 91-95.

[9] Krishnamoorthy, M., A.T. Ernst, and Y.M., Sharaiha. 1998. Algorithms for the Degree-Constrained Minimum Spanning Tree. Proceeding of Fourth International Conference on Optimization Techniques and Application, pp. 859-866

[10] Krishnamoorthy, M., A.T. Ernst, and Y.M., Sharaiha. 2001. 'Comparison of Algorithms for the Degree Constrained Minimum Spanning Tree'. Journal of Heuristics, Vol. 7 no. 6, pp. 587-611.

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

[12] Deo N. and N. Kumar. 1997. 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.

[13] Wamiliana. 2002. 'The Modified Penalty Methods for The Degree Constrained Minimum Spanning Tree Problem'. Journal Sains dan Teknologi, Vol. 8, pp.1-12.

[14] Wamiliana. 2004b. 'Solving the Degree Constrained Minimum Spanning Tree Using Tabu and Penalty Method', Jurnal Teknik Industri, pp.1-9.

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

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

[17] Wamiliana and L. Caccetta. 2013. 'The Modified CW1 Algorithm for The Degree Constrained Minimum Spanning Tree Problem'. International Journal of Engineering Science and Technology, Vol.1, No.2, pp. 32-36.

[18] Wamiliana, Dwi Sakethi, Akmal J., and Edy Tri Baskoro. 2005. "The Design of Greedy Algorithm for Solving The Multi Period Degree Constrained Minimum Spanning Tree Problem", Jurnal Sains dan Teknologi. Vol.11, pp. 93-96.

[19] Junaidi Akmal, Wamiliana, Dwi Sakethi, and Edy Tri Baskoro. 2008. Computational Aspects of Greedy Algorithms for Solving The Multi Period Degree Constrained Minimum Spanning Tree Problem. J. Sains MIPA, Edisi Khusus Tahun, Vol. 14, No. 1, pp. 1-6.

[20] Wamiliana, Dwi Sakethi, and Restu Yuniarti. 2010. Computational Aspect of WADR and WADR2 Algorithms for The Multi Period Degree Constrained Minimum Spanning Tree Problem, Proceeding SNMAP 2010, pp. 208-214.

[21] Gavish, B. 1985. Augmented Lagrangian based algorithms for centralized

[22] Network design. IEE Transaction of Communication, vol. 33, pp.1247-1257.

[23] [22] Kruskal, J.B. 1956. On the Shortest Spanning Tree of a Graph and the Traveling Salesman Problem. Proc. Amer. Math. Soc, 7, 48-50.

[24] [23] Prim, R.C. 1957. Shortest Connection Networks and Some Generalizations. Bell System Technical Journal, 36, pp.1389-1401.