Indian Journal of Science and Technology, *Vol 11(11), DOI: 10.17485/ijst/2018/v11i11/117069, March 2018*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Comparative Analysis of Some Modified Prim's Algorithms to Solve the Multiperiod Degree Constrained Minimum Spanning Tree Problem

Wamiliana[1*], Asmiati[1], Mustofa Usman[1], Astria Hijriani[2] and Wibi Cahyo Hastono[2]

[1]Department of Mathematics, Faculty of Mathematics and Natural Sciences, UniversitasLampung, Indonesia; wamiliana.1963@fmipa.unila.ac.id, asmiati308@yahoo.com, usman_alfha@yahoo.com
[2]Department of Computer Science, Faculty of Mathematics and Natural Sciences, Universitas Lampung, Indonesia; astria.hijriani@fmipa.unila.ac.id, wibihastono@gmail.com

Abstract

Objectives: To compare the WAC1, WAC2, and WAC3 algorithms against WADR1, WADR2, WADR3, WADR4, and WADR5 algorithms to solve the Multiperiod Degree Constrained Minimum Spanning Tree. **Methods/Statistical analysis:** WAC1, WAC2, and WAC3 are algorithms developed by modifying Prim's algorithm for the Minimum Spanning Tree (MST) and adopting the period of installation/connecting for vertices in the network. In WAC1, WAC2, and WAC3 algorithms we consider the HVT_k, the set of vertices that must be connected on k^{th} period, while in WADR5 is not. In WAC1 the vertices in HVT_k must be installed as early as possible, while in WAC2 is not given priority to be connected as soon as possible, but can be any time as long as the connection still on that current period. In WAC3, we adopt the smallest value for 2-path for vertex under consideration to be connected. All algorithm proposed used the same data as used in WADR1, WADR2,WADR3, WADR4 and WADR5. **Findings:** Since WAC1, WAC2, and WAC3, WADR5 algorithms are based on modified Prim's algorithm, then the connectivity property is maintained during the process of installation/connection not like WADR1, WADR2, WADR3, and WADR4 where based on kruskal's. Based on the same data used and connectivity property, the result shows that the performance of WAC2 is the best among the other algorithms developed. **Application/Improvements:** Considering real life application in the network installation problem, the WAC2 algorithm is one of alternative solutions since it maintains connectivity property and performs best.

Keywords: Comparative Analysis, Connectivity, Degree Constrained, Modified Prim, Multi Period

1. Introduction

There is no doubt that many network design problems usually use graph to represent the network. In network design problem we construct a network that satisfies certain requirements which is optimal according to some criterion. Graph is used to represent the network, where the vertices can represent cities/stations/computers etc. and the edges of the graph can represent roads/links safety and so on; and the criterion can be cost, output, performance etc.

Many network design problems used Minimum Spanning Tree (MST) as the backbone of the problem. In order to apply the MST into the real-life situation, some other parameters can be used as added restriction such as degree, diameter, period, and so on. To solve a minimum spanning tree problem, Prim's algorithm is one algorithm that can be used. If, in addition to the MST,there are constraints on every vertices, the problem is called as the Degree Constrained Minimum Spanning Tree (DCMST) problem.The DCMST concerned of finding an MST that satisfies specified degree restrictions on its vertices[1].

**Author for correspondence*

Moreover, if in the network's installation process must be done in some stages or periods due to other restriction such as fund limitation, weather, and so on, the DCMST becomes Multi Period Degree Constrained Minimum Spanning Tree Problem. The brief review of the method, the algorithms proposed, and the data for implementation are given in Section 2. The Results and Discussion will be given in Section 3, followed by Conclusion in Section 4.

2. Method

As already stated before, MST as one of fundamental structures, has many applications. The MST structure usually is used as the backbone of the problem. There are two well-known and widely used algorithms to solve the MST[2,3]. Even though there are some other algorithms for solving the MST such as Sollin's algorithm or Boruvka, but the previous two are commonly used. Boruvka developed an algorithm to find the most economical layout for a power-line network[4–6]. Minimum spanning trees, in general, are used in many network optimization problems as the key structure. Since G.R. Kirchoff designed electrical circuits in the 19th century, spanning trees have been considered as one of the most used subgraphs in many network design applications[7]. For a given connected weighted graph, the spanning trees can be computed in linear time. To get a minimum weight spanning tree the computational time increases slightly[8].

The DCMST Problem is related with finding a MST while also has degree restriction on the vertices. It is showed in[9] that the problem is NP complete by reducing the degree on every vertices exactly two and making the DCMST to be a famous and highly investigated problem : the Travelling Salesman Problem (TSP). Because of NP completeness of DCMST, the heuristic methods have dominated. Some of the heuristics that had been investigated include: a number of basic MST algorithms[10]; the genetic Algorithm[11]; Simulated Annealing[12]; and Iterative Refinement[8,13]; Tabu Search[14–16] and Modified Penalty[17].

Some of the exact algorithms for solving the DCMST problem already proposed such as the Branch and Bound (BB) in which the branching procedure is an adaptation of the method of [18,19] for the TSP; the branch and bound algorithm based on an edge exchange analysis and utilized three heuristics were used[20] (the primal method[10], a heuristic[21], and a heuristic based on edge exchange). Introducing penalty by applying Lagrange multiplier πinthe Lagrangean Relaxation method and adding them to the objective function was implemented[22,23]; and finally a branch and cut method is used for the DCMST problem by generating an upper bound using the heuristics approach[20,24], the initial lower bounds are generated at the root node using the Lagrangean procedure[23] and the used of depth first search procedure was the important features of the method. The Multi Period Degree Constrained Minimum Spanning Tree (MPDCMST) Problem was introduced and investigated by using branch exchange technique as a hybrid to Lagrangean relaxation, and the method was implemented using vertices varying from 40 to 100; 10 year planning horizon; the time period for activating each terminal is uniformly distributed from 1 to 6; and set vertex 1 as central vertex[25]. In the research of design of greedy algorithm for solving the MPDCMST[26] isused one year planning horizon and divided the installation into three periods (four-month each) and four periods (three-month each). That modification of planning horizon and time period in MPDCMST was made to mimic the real situation in Indonesia where the funding for every project usually divided into three terms or periods. In the study of computational aspect of MPDCMST[27] is used not only 300 random tables problem[26], but also some problems taken from TSPLIB. Motivated by Kruskal's algorithm, WADR1 and WADR2 algorithms were developed[7], and in the searching used DFS technique with k = 2, k is the length of the node path. In the algorithms,terminology HVT_i is introduced which is a set of vertices that must be already in the networks after period i finished. The use of HVT_i is to tackle the problem that some facilities (for example hospital, police station, or other public need facilities) must be in the network earlier to handle public needs. The difference between WADR1 and WADR2 lied on the process of installation HVT_i. By setting $HVT_i = 3$, $k \le 3$, $MaxVT_i = \left\lceil \frac{n-1}{3} \right\rceil$, the WADR1 and WADR2 improved[28]. The WADR3 and WADR4 were the two modified Kruskal algorithms based on WADR1 and WADR2 by relaxing the HVT_i and introducing the best k-path, with k = 3, and WADR5 is based on Prim's[29]. The detail why the different solution occurs for WADR3 and WADR4 when the algorithms implemented with different HVT_i is given along with the illustration[30].

2.1 The Algorithms

We propose three algorithms based on Prim's algorithm which shown on the following pseudocode:

```
Initiation:V={1},T=0,n=number_of_vertex,k=1, kMax=3
begin
while k < kMax
    do
    if |HVT_k| > MaxVT
    stop
    else
    T_k = 0
        while T_k< (|HVT_k|-1)
        do
        find the shortest edge which connects with vertices in V
        store in T
        if the connecting vertex not include in HVT_k
        go to the next edge
        else
            if adding an edge constitute circuit
            choose the next edge
            else
                if adding an edge violate degree restriction
                choose the next edge
                else
                store the edge in T and the vertex incident to it in V
                T_k++
                endif
            endif
        endif
    end
while T_k< (MaxVT_k-1)
do
find the shortest edge which connects with vertices in V
store in T
if adding an edge constitute circuit
choose the next edge
else
        if adding an edge violate degree restriction
        choose the next edge
        else
        store the edge in T and the vertex incident to it in V
        T_k++
            endif
        endif
end
k++
endif
endwhile
end
```

The set of vertices that must be installed/connected on k^{th} period is notated as HVT_k. T_k is the number of edges that already installed /connected on k^{th} period, k is the current period, and the number of period for installation is notated as kMax. The maximum number of vertices that can be installed/connected on k^{th} period is notated as $MaxVT_k$. The coding process is divided into four main stages as shown on the flowchart in Figure 1.

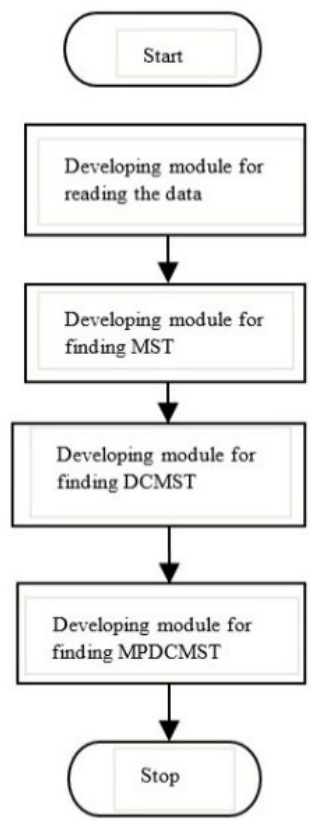

Figure 1. Four main stages of the process.

2.2 Data for Implementation

The module for reading the data is developed to read the data from the source. On the implementation 300 random table problems are used. One problem on the data represents a complete graph with specific order. The order of graphs are 10 to 100 with increment of 10 and for every vertex order there are 30 problems for simulation. Therefore,there are total 300 problems to be tested[29,30]. The module for finding MST is designed to solve the MST of the problem using Prim's algorithm. In this research Prim's algorithm is used instead of Kruskal' algorithm because Prim's algorithm maintains the conectivity of the network during installation proccesses.The module for finding DCMST is the improvement of module for finding MST with the restriction on every vertices. Here, we add a degree rectriction on every vertex by setting $d_i \leq 3$. The degree of vertex i is d_i. To find the MPDCMST is the last module which is the improvement of the module for finding DCMST by adding number of periods for installing the network, and vertex priority to be installed on a certain period.

We developed three algorithms based on modified Prim's algorithm. The first algorithm (WAC1) is the simplest one. The algorithm just follows the original Prim's algorithm. The modification made by adding the degree restriction on edge insertion processes and checking the element on HVT_k on every period. In this algorithm the vertices on HVT_k are given priority to be connected/installed as early as possible. The set of $HVT_k$is given in Table 1. For n = 10, by setting v_1 as the root, V={v_1}, and using k = 1,2,3 and HVT_1 = {2}, HVT_2={3}, and HVT_3 = {4}, the result in every period installation is given in Figure 2.

Figure 2. Stage of installation for every period of WAC1 Algorithm.

Table 1. Data file 22.dat (10 vertices)

Edge	e_{12}	e_{13}	e_{14}	e_{15}	e_{16}	e_{17}	e_{18}	e_{19}	$e_{1,10}$
Weight	740	572	447	835	427	807	362	832	120
Edge	e_{23}	e_{24}	e_{25}	e_{26}	e_{27}	e_{28}	e_{29}	$e_{2,10}$	e_{34}
Weight	221	109	276	741	987	352	368	403	505
Edge	e_{35}	e_{36}	e_{37}	e_{38}	e_{39}	$e_{3,10}$	e_{45}	e_{46}	e_{47}
Weight	921	757	884	369	886	545	639	253	750
Edge	e_{48}	e_{49}	$e_{4,10}$	e_{56}	e_{57}	e_{58}	e_{59}	$e_{5,10}$	e_{67}
Weight	251	187	857	807	926	781	605	112	559
Edge	e_{68}	e_{69}	$e_{6,10}$	e_{78}	e_{79}	$e_{7,10}$	e_{89}	$e_{8,10}$	$e_{9,10}$
Weight	411	473	743	882	693	851	509	434	828

Note that the box between every pair of vertices represents the distance/cost/weight. For instance, the weight of edge e_{24} (weight from v_2 to v_4) is smaller than e_{12}.

For the second algorithm (WAC2), the vertices on HVT_k are not given priority to be connected as soon as possible, but can be any time as long as the connection still on that certain period. Figure 3 gives the the result obtained on every period of WAC2 algorithm. For the third algorithm (WAC3) we adopt the Depth First Search technique as in[30] by applying the smallest value for2-path. Figure 4 shows the result obtained on very period of WAC3 algorithm.

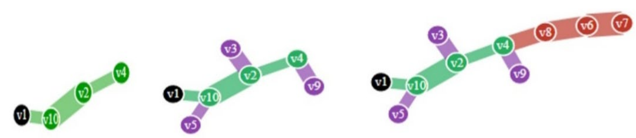

Figure 3. Stage of installation for every period of WAC2 Algorithm.

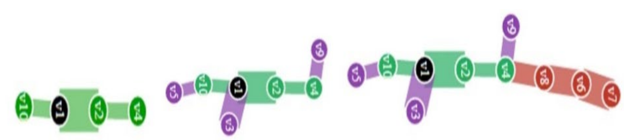

Figure 4. Stage of installation for every period of WAC3 Alogrithm.

3. Results and Discussion

We implemented our heuristic using the C++ programming language running on dual core computer, with 1.83 Ghz and 2 GB RAM. We used the same elements on HVT_k[30] as in Table 2.

We compare our algorithms with WADR1, WADR2, WADR3, WADR4,and WADR5[29,30]. Please note that WADR1, WADR2 WADR3, and WADR4 are algorithms developed by modifying Kruskal's algorithm. Therefore, during the process of installation is possible the network constitute a forest (not maintains the connectivity) eventhough at the end all vertices are connected in the network. Besides WAC1, WAC2 and WAC3, the WADR5 is the algorithms that developed based on Prim's algorithm. Figure 5 shows the result.

Table 2. The list of vertices in HVT_i, i = 1,2,3[30]

n	HVT_1	HVT_2	HVT_3
10	{2}	{3}	{4}
20	{2}	{3}	{4}
30	{2,3}	{4,5}	{6,7}
40	{2,3,4}	{5,6,7}	{8,9,10}
50	{2,3,4,5}	{6,7,8,9}	{10,11,12,13}
60	{2,3,4,5,6}	{7,8,9,10,11}	{12,13,14,15}
70	{2,3,4,5,6,7}	{8,9,10,11,12,13}	{14,15,16,17,18,19}
80	{2,3,4,5,6,7,8}	{9,10,11,12,13,14,15}	{16,17,18,19,20,21,22}
90	{2,3,4,5,6,7,8}	{9,10,11,12,13,14,15}	{16,17,18,19,20,21,22}
100	{2,3,4,5,6,7,8,9}	{10,11,12,13,14,15,16,17}	{18,19,20,21,22,23,24,25}

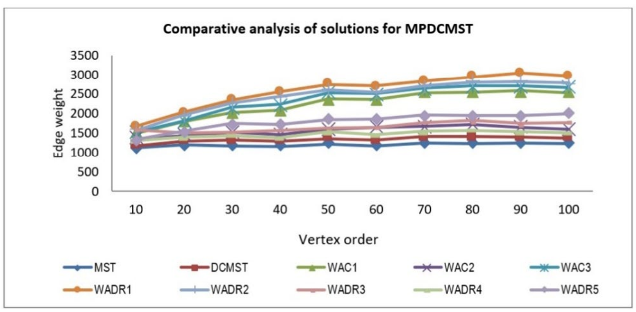

Figure 5. Comparative analysis of some heuristics for the MPDCMST.

4. Conclusion

From the above discussion we see that the average solutions of WAC1, WAC2, and WAC3 heuristics are better than WADR1 and WADR 2. The heuristics that performs better than WAC1 and WAC3 are WADR3, WADR4, WADR5 and WAC2. But, WADR3 and WADR4 are algorithms developed based on Kruskal's algorithm in which during installation process, disconnectivity of the networks is permissible, while in WAC1, WAC2, and WAC3 the connectivity is maintained. WADR5, which is based on Prims's algorithm performs better than WAC1 and WAC3, but WAC2 performs better than WADR5. The result shows that on this comparison, if we are considering of maintaining connectivity in the whole process of installation, then WAC2 heuristicis the best.

5. Acknowledgement

This research was supported by The Directorate General of Higher Education (DGHE), Ministry of Research Technology and Higher Education of The Republic of Indonesia under contract # 582/UN26.21/KU/2017. The authors would like to thank DGHE for that support.

6. References

1. Wamiliana. Combinatorial methods for degree constrained minimum spanning tree problem. Department of Mathematics and Statistics, Curtin University and Technology: Australia; 2002.
2. Kruskal JB. On the shortest spanning tree of a graph and the travelling salesman problem. Proceedings of the American mathematical society. 1956; 7(1):48–50. crossref
3. Prim RC. Shortest connection networks and some generalizations. The Bell System Technical Journal. 1957; 36(6):1389–401. crossref
4. Graham RL, Hell P. On the history of the Minimum Spanning Tree Problem. IEEE Annals of the History of Computing. 1985; 7(1):43–57. crossref
5. Boruvka O. Jistém Problému Minimálním. Práce Moravské přírodovědecké společnosti. 1926; 3:37–58.
6. Boruvka O. Příspěvek k řešení otázky ekonomické stavby Elektrovodních sítí. Elektronický Obzor. 1926; 15:153–4.
7. Wamiliana, Sakethi D, Yuniarti R. Computational aspect of WADR1 and WADR2 algorithms for the multi period

degree Constrained Minimum Spanning Tree Problem. Proceeding SNMAP, Bandarlampung; 2010. p. 208–14.

8. Deo N, Kumar N. Computation of constrained spanning trees: A unified approach. Network Optimization, Lecture Notes in Economics and Mathematical Systems. 1997; 450:194–220. crossref
9. Garey MR., Johnson DS. Computers and intractability. A guide to the theory of np completeness. Freemann: San Francisco USA; 1979. p. 1–13.
10. Narula SC, Cesar AHO. Degree-constrained minimum spanning tree. Computer and Operation Research. 1980; 7(4):239–49. crossref
11. Zhou G, Gen M. A note on genetics algorithms for degree-constrained spanning tree problems. Networks. 1997; 30:91–5. crossref
12. Krishnamoorthy M, Ernst AT, Sharaila YM. Comparison of algorithms for the degree constrained minimum spanning tree. Journal of Heuristics. 2001; 7(6):587–611. crossref
13. Boldon B, Deo N, Kumar N. Minimum weight degree-constrained spanning tree problem: Heuristics and Implementation on an SIMD parallel machine. Parallel Computing. 1996; 22:369–82. crossref
14. Caccetta L, Wamiliana. Heuristics algorithms for the degree constrained minimum spanning tree problems. Proceeding of the International Congress on modeling and Simulation (MODSIM), Canberra; 2001. p. 2161–6.
15. Wamiliana, Caccetta. Tabu search based heuristics for the degree constrained minimum spanning tree problem. Proceeding of South East Asia Mathematical Society, Yogyakarta; 2003. p. 133–40.
16. Wamiliana, Cacetta L. The modified CW1 algorithm for the degree restricted minimum spanning tree problem. Proceeding of International Conference on Engineering and Technology Development, Bandarlampung; 2012. p. 36–9.
17. Wamiliana. Solving the degree constrained minimum spanning tree using tabu and penalty method. Journal Teknik Industry. 2004; 6(1):1–9.
18. Held M, Karp RM. The travelling salesman problem and minimum spanning trees. Operation Research. 1970; 18:1138–62. crossref
19. Held M, Karp RM. The traveling salesman problem and minimum spanning trees Part II. Mathematical Programming. 1971; 1:6–25. crossref
20. Savelsbergh M, Volgenant T. Edge exchange in the degree-constrained minimum spanning tree. Computer and Operation Research. 1985; 12:341–8. crossref
21. Christofides N. Worst-case analysis of a new heuristic for the travelling salesman problem. Carnegy-Mellon University: Pittsburg USA; 1976. p. 1–11.
22. Gavish B. Topological design of centralized computer networks formulations and algorithms. Networks. 1982; 12:355–77. crossref
23. Volgenant A. A lagrangean approach to the degree-constrained minimum spanning tree problem. European Journal of Operational Research. 1989; 39(3):325–31. crossref
24. Caccetta L, Hill SP. A branch and cut method for the degree constrained minimum spanning tree problem. Networks. 2001; 37:74–83. crossref
25. Kawatra R. A multi period degree constrained minimum spanning tree problem. European Journal of Operational Research. 2002; 143:53–63. crossref
26. Wamiliana, Sakethi D, Akmal J, Baskoro ET. The design of greedy algorithm for solving the multi period degree constrained minimum spanning tree problem. Journal MIPA FMIPA University of Lampung. 2005; 11(2):93–6.
27. Junaidi A, Wamiliana, Sakethi D, Baskoro ET. Computational aspect of greedy algorithm for the multi period degree constrained minimum spanning tree problem. Journal SAINS MIPA. 2008; 14(1):1–6.
28. Wamiliana, Amanto, Usman M. Comparative analysis for the multi period degree constrained minimum spanning tree problem. Proceeding The International Conference on Engineering and Technology Development (ICETD); 2013. p. 39–43.
29. Wamiliana, Elfaki FAM, Usman M, Azram M. Some greedy based algorithms for multi periods degree constrained minimum spanning tree problem. ARPN Journal of Engineering and Applied Sciences. 2015; 10(21): 10147–52.
30. Wamiliana, Usman M, Sakethi D Yuniarti R, Cucus A. The hybrid of depth first search technique and Kruskal's algorithm for solving the multi period degree constrained minimum spanning tree problem. The 4th International Conference on Interactive Digital Media (ICIDM); 2015 Dec. p. 1–4.