Science and Technology Indonesia
e-ISSN:2580-4391 p-ISSN:2580-4405
Vol. 8, No. 4, October 2023

Research Paper

Solving the Shortest Total Path Length Spanning Tree Problem Using the Modified Sollin and Modified Dijkstra Algorithms

Wamiliana[1]*, Reni Permata Sari[2,3], Astri Reformasari[1], Jani Suparman[1], Akmal Junaidi[4]

[1]Department of Mathematics, Faculty of Mathematics and Natural Science, Universitas Lampung, Bandar Lampung, 35145, Indonesia
[2]Doctoral Program, Faculty of Mathematics and Natural Sciences, Universitas Lampung, Bandar Lampung, 35145, Indonesia
[3]Faculty of Science and Technology, Universitas Nahdatul Ulama, Lampung, 34192, Indonesia
[4]Department of Computer Science, Faculty of Mathematics and Natural Science, Universitas Lampung, Bandar Lampung, 35145, Indonesia
***Corresponding author**: wamiliana.1963@fmipa.unila.ac.id

Abstract
In a weighted connected graph, the shortest total path length spanning tree problem is a problem when we need to discover the spanning tree with the lowest total cost of all pairwise distances between its vertices. This problem is also known as the minimum routing cost spanning tree (MRCST). In this study, we will discuss the Modified Sollin and Modified Dijkstra Algorithms to solve that problem which implemented on 300 problems are complete graphs of orders 10 to 100 in increments of 10, where every order consists of 30 problems. The results show that the performance of the Modified Dijkstra and the Modified Sollin Algorithms are slightly similar. On orders 10, 20, 30, 60, and 80, the Modified Dijkstra Algorithm performs better than the Modified Sollin, however on orders 40, 50, 70, 90, and 100, the Modified Sollin performs better.

Keywords
Shortest Total Path Length Spanning Tree Problem, Modified Sollin Algorithm, Modified Dijkstra

Received: 31 June 2023, Accepted: 5 September 2023
https://doi.org/10.26554/sti.2023.8.4.684-690

1. INTRODUCTION

Graph theory is one of the mathematical branches that can be used to represent problems in daily life. Graf $G(V, E)$, $V \neq \emptyset$, V is the set of vertices and E denotes the set of edges that connect the vertices. Because of its flexible structure where there is no specific rule in drawing a graph (where it is possible to draw an edge as a curve, a line, etc.), a graph emerges as one of the important tools for representing problems. A wide range of sciences and technology employ graph theory for instance: in computer science graphs are used in database designing, software engineering, and network design (Elumalai, 2020; Singh, 2014). The graph structure plays a crucial role in database architecture since it provides quick implementation using various graph functionalities and features. The vertices indicate transformations and the edges represent the data flows on the data flow diagram during the requirements specification process. A program's control flow during testing uses directed graphs to address the order of executed instructions, whereas, during the design phase, graphical design is utilized to describe relationships between modules. In social science de Nooy (2009) adapted the graph structure to do the social networks analysis, where the social network is viewed as a graph in a graph theoretical approach, which is made up of a collection of vertices that represent social actors and a set of edges that reflect one or more social links among them; while in life science, Nøjgaard (2020) explored graph theoretical concepts, especially on formalizing and addressing the issues raised, including self-assembling protein design, evolutionary biology, chemical compounds, etc. A leaf-labeled tree was used in biology to illustrate a phylogeny (Brandes and Cornelsen, 2009; Huson and Bryant, 2006). A phylogeny or phylogenetic tree is a branching diagram that depicts the paths of diverse species, organisms, or genes descended from a common ancestor. In agriculture, Kannimuthu et al. (2020) used the graph coloring algorithm to help farmers decide what crops to grow while protecting their investment in agricultural cultivation and balancing crop demand. In this algorithm, the land region is represented by the vertices, and crops are represented by the colors, and the first requirement is to color the vertices so that there are no two adjacent vertices that have the same color. Kawakura and Shibasaki (2018) used a spanning tree, one of the useful graph theory concepts, to create methods for both close and far agriculture laborers' observations engaged in cropping tasks.

2. CONSTRAINED SPANNING TREE PROBLEM

A connected graph containing no cycle is a tree, and a spanning tree T of an undirected graph G is a subgraph of G, a tree with all of its vertices present. If graph G is a weighted graph, the minimum spanning tree (MST) of graph G is a spanning tree of graph G whose total weight/cost is minimum. Spanning tree is a concept in graph theory that is used in many real-life applications, and MST is one concept in graph theory that is utilized as a backbone in numerous network design issues (Sari et al., 2022). When Borůvka (1926) resolved the issue of building Moravia's electricity network in the Czech Republic, he proposed the first algorithm to solve the MST. However, the two popular algorithms for solving MST are Prim's algorithm by Prim (1957) and Kruskal's algorithm by Kruskal (1956). The MST problem is frequently encountered in network design applications when various graph criteria like diameter, distance, degree, flow, connectedness, etc. must be fulfilled. For example, the bounded diameter MST is a combinatorial optimization problem that optimizes a tree weight while keeping the hop diameter. This optimization challenge is useful for designing a computer network with the lowest possible cost and the shortest possible network delay to achieve service quality while decreasing the likelihood of communication failure (Segal and Tzfaty, 2022). The Degree Constrained MST occurs when designing a network where the vertex/node has a maximum bound of the number of interconnecting channels. If, besides degree, the period is added as a constraint, the problem becomes a multi-period degree-constrained MST. Kawatra (2002) solved this problem for digraphs, while Wamiliana et al. (2020; 2015a; 2015b) solved it for an undirected graph. The period is added as a restriction because in real-life problems it is possible that developing or building a network is done stage by stage due to some conditions such as weather, limited funds, and others.

The other problem that uses a spanning tree as the backbone is the shortest total path length spanning tree (STPL) problem. This problem occurs when we must determine the tree with the lowest communication costs i.e., the tree that has the smallest total distance for all pairs of vertices computed across the whole network. A STPL or MRCST problem represents a spanning tree T, one of all spanning trees of G so that $C_r(T)^* = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}$, $i \neq j$, where c_{ij} is the cost of vertex (i, j), and T^* is the spanning tree whose the minimal routing cost is the minimal among all spanning tree in G (Campos and Ricardo, 2008). For more precise, the STPL problem is defined as follows: given $G(V, E)$, $V=\{v_1, v_2, v_3, \cdots, v_n\}$, G undirected, $V \neq \emptyset$, E is the set of edges that connect the vertices in V, G is connected, and for every $e_{ij} \in E$ there is a nonnegative cost c_{ij} associated with it, $d(i, j)$ is the distance between the vertices i and j in G, the STPL or MRCST problem is to find a spanning tree T^* so that $C_r(T^*)= \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}$, $i \neq j$ T^* is the spanning tree in G that produces the shortest total distance between every two vertices. Figure 1 below illustrates the STPL problem.

Suppose that we have a graph G and one of its spanning

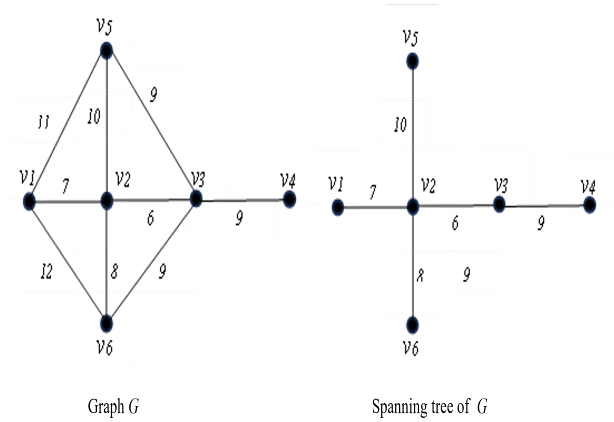

Figure 1. Example of Graph G and One of its Spanning Trees T

tree T as shown in Figure 2. The STPL of T is the total length of the distance of every two vertices. The distances of every pair of vertices are: $d(v_1, v_2)$= 7, $d(v_1, v_3)$= 13, $d(v_1, v_4)$= 22, $d(v_1, v_5)$= 17, $d(v_1, v_6)$= 15, $d(v_2, v_3)$= 6, $d(v_2, v_4)$= 15, $d(v_2, v_5)$= 10, $d(v_2, v_6)$= 8, $d(v_3, v_4)$= 9, $d(v_3, v_5)$= 16, $d(v_3, v_6)$= 14, $d(v_4, v_5)$= 25, $d(v_4, v_6)$=23, $d(v_5, v_6)$=18. Since $C_r = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}$, $i \neq j$, then the value of C_r is double the sum of the distance of every pair of vertices. Thus, the value of STPL in the example in Figure 1 is $C_r = 2 \times (7 + 13 + 22 + 17 + 15 + 6 + 15 + 10 + 8 + 9 + 16 + 14 + 25 + 23 + 18) = 2 \times 218 = 436$.

Some researchers already investigated the shortest total distance spanning tree, and since this problem is an NP-hard problem the heuristics are more proposed. The bee colony algorithm was investigated by Singh (2008), Tan (2012b), and Singh and Sundar (2011), while Hieu et al. (2011) investigated the ant colony algorithm. Julstrom (2001; 2005), and Tan (2012a) proposed a genetic algorithm to tackle the problem. Julstrom (2005), also coded the tree in Blob code and demonstrated that in genetic algorithms, the tree represented in Blob code performed better than the tree coded as an edge-set, as proposed by Raidl and Julstrom (2003). Fischetti et al. (2002) showed that in addition to network design, trees with low routing costs are important in biological computation, where they can be used to find acceptable genomic sequence alignments. Masone et al. (2019) offered a broad and thorough understanding of the topic while also laying the groundwork for the next research activities such as the evolution of the proposed heuristic's evolution inside a framework for metaheuristics.

3. RESULTS AND DISCUSSION

3.1 The Modified Dijkstra Algorithm

Before starting the Modified Dijkstra Algorithm, we need to do preprocessing. The preprocessing process runs the Dijkstra Algorithm to determine the shortest path for every two vertices so that the number of trees obtained is $\frac{n(n-1)}{2}$ where the order of the graph is n. Next, construct a table that gives the list of

© 2023 The Authors.

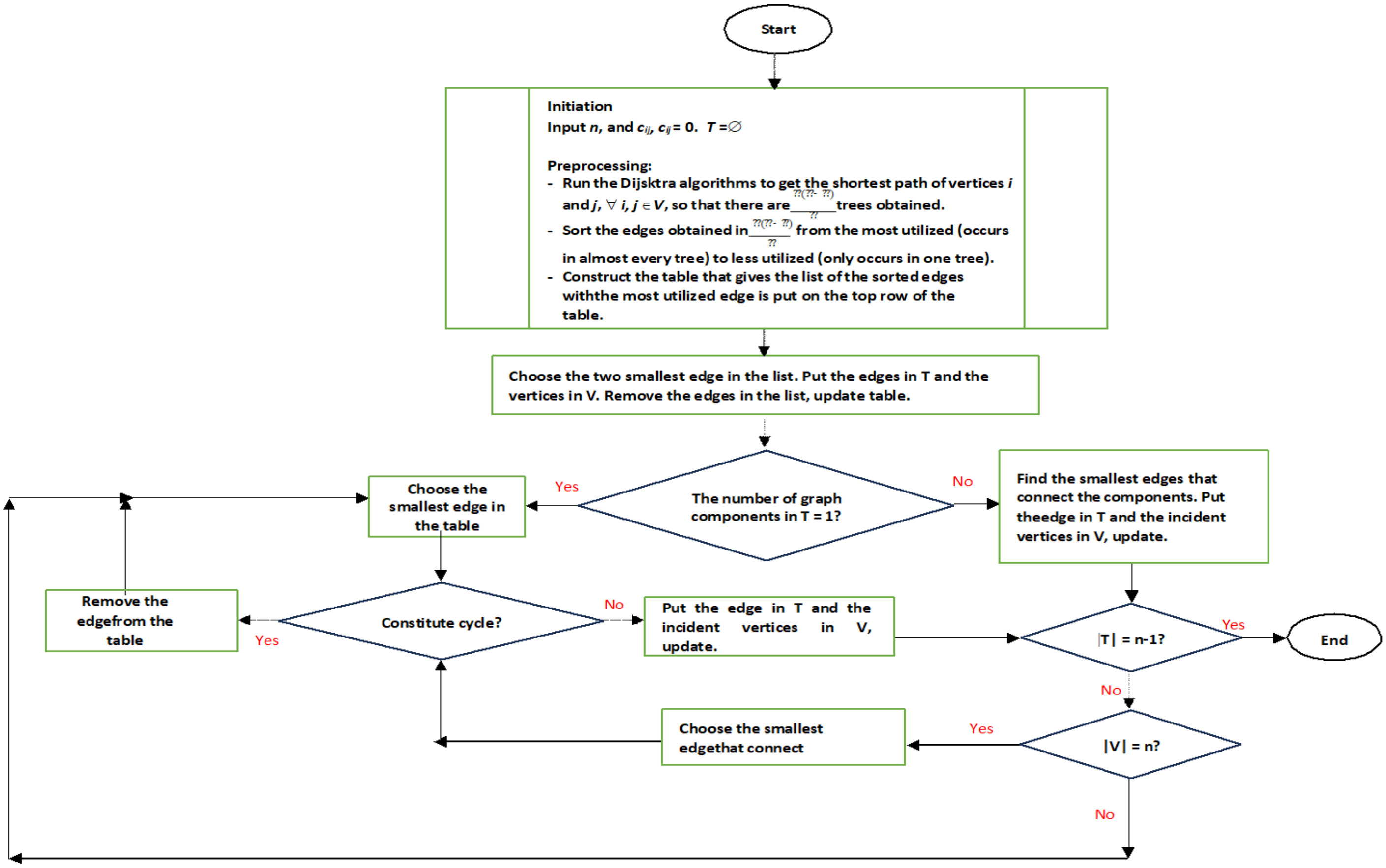

Figure 2. Flowchart Modified Dijkstra Algorithm

Table 1. The Solutions for STPL for Graphs of Order 10 to 100

Data	Vertex order 10		20		30		40		50		60		70		80		90		100	
	Mod-Dijkstra	Mod-Sollin	Mod-Dijkstra	Mod-Sollin	Mod-Dijkstra	Mod-Sollin	Mod-Dijkstra	Mod-Sollin	Mod-Dijkstra	Mod-Sollin	Mod-Dijkstra	Mod-Sollin	Mod-Dijkstra	Mod-Sollin	Mod-Dijkstra	Mod-Sollin	Mod-Dijkstra	Mod-Sollin	Mod-Dijkstra	Mod-Sollin
1	27022	28948	77870	77870	116482	116482	201060	201060	285712	285712	385978	385978	559070	521956	636448	617738	822052	661388	748488	770046
2	23092	23092	92032	92796	144080	144080	259132	235020	280408	280408	511416	421188	623082	573232	673364	709768	838642	867600	902720	819444
3	20540	20540	77204	77732	166512	166512	203146	203146	352090	365060	477242	452894	440202	450988	674122	645122	723342	786342	882802	888782
4	22600	22114	91800	91800	120390	120390	284214	287686	283752	318320	428058	419770	614922	540916	545598	508722	760322	693226	628752	669664
5	35340	39510	106544	122026	206064	206064	212686	216646	518696	484656	459334	473506	573924	577014	609294	627612	814180	736448	720184	777474
6	35986	35414	75642	75642	158754	158754	205122	205122	258174	259284	455502	368744	587122	584454	515190	510126	800502	668144	806528	719346
7	48704	46728	101674	102282	178348	178348	238180	245802	313790	313790	286154	286154	584566	524698	823254	686218	818210	644096	980394	940732
8	12026	13162	106150	106150	149858	149858	207390	222230	352694	352694	423884	423558	497186	490728	573062	605304	647482	627394	851142	756896
9	37476	37476	85436	88098	143762	143762	146928	146928	371072	354200	464898	414108	538534	544446	567586	522486	879640	820846	749792	714808
10	45764	39256	73562	73562	107582	107582	197738	195164	345380	343848	426814	449242	563248	604214	699462	678724	675588	687814	1037660	975554
11	32020	28582	160460	147744	245300	245300	287366	317254	424632	371968	355454	344324	658662	653834	656826	1E+06	941972	810270	799976	746296
12	37346	36860	55784	55784	157466	157466	281146	267008	326126	338084	432526	432526	618406	531714	677138	706334	653890	666754	627048	631288
13	23874	22654	56960	56960	168918	168918	354416	363826	212538	212538	414328	549848	455524	458488	718138	743334	680378	745014	838848	786742
14	32860	36172	118204	118204	145506	139794	282942	287998	336664	371222	410520	410520	343608	382022	598206	689240	736914	638414	943618	1144920
15	22012	24318	101166	101166	150590	150590	266264	231254	259772	259772	464980	534192	590984	575756	753234	805328	678166	826772	902922	841240
16	26842	25436	98630	98630	137014	137014	192776	192776	406898	364294	448976	499070	378840	411246	633072	507992	609758	638252	744106	830120
17	19216	19216	98420	98420	193226	193226	273200	252824	375632	387040	422182	401356	591346	673252	720814	740600	785716	766008	1032946	1155440
18	34860	33292	122780	122780	106788	106788	153918	153918	323112	323112	542238	482618	411758	408686	910110	800522	761484	681182	883976	838610
19	20764	26546	62030	62030	186818	186818	191516	191516	330794	385694	351594	351594	416220	477522	690150	674272	825238	834760	901362	963428
20	39862	38574	62032	62032	99420	99420	208154	194534	367154	354358	649786	544056	386030	370450	708840	702624	584904	657652	847242	946388
21	43582	50272	75110	75110	126188	128708	207668	207668	269264	256280	445250	488962	549366	432988	697378	706884	688244	637150	815934	776358
22	52806	54146	87326	87326	86076	86076	237722	240582	433004	426520	557206	627036	569088	524190	667798	562070	742332	708266	716202	710374
23	25728	25728	91910	91910	161204	161204	236680	241608	314234	276386	360716	347738	562784	511320	471356	530264	589922	547594	975550	1142624
24	28666	25800	101416	101416	177962	178922	365618	376336	327320	327320	500606	496030	455776	394260	659974	622510	988832	851342	993688	804198
25	38494	34446	79828	79828	207992	207992	248148	248148	459718	478848	268640	307616	612572	564190	530236	511530	521726	558406	692108	885458
26	33050	33050	104714	96320	139612	139612	225690	225690	477278	408514	245266	246328	381848	376068	658658	615944	673610	481314	978396	936704
27	22476	28700	92708	96164	152152	152152	289596	289596	392982	434880	383018	376768	511146	457814	561088	538818	1050126	871776	886498	755906
28	38834	39634	77006	77006	128628	128628	262102	262102	385208	346840	435228	404244	598462	484364	641380	526682	740194	601562	580488	570504
29	23696	23524	62838	62838	209498	252940	266966	266966	209272	209272	449848	526842	511894	589884	524166	537120	844990	787390	986820	930190
30	25986	24520	74212	74212	129936	129936	204362	204362	389022	379014	444572	439814	544574	660288	542526	530668	560962	553738	815596	753386
Average	31050.8	31257	89048.3	89127.933	153404.2	154777.87	239728.2	239159	346080	342330.93	430074	430221	524358	511699	644616	647550.3	747977	701897.1	842392.87	839430.67

the edges used in preprocessing (edges that formed the trees on preprocessing). Sort the edges from the most utilized (occurs in almost every tree) to less utilized (only occurs in one tree).

The Modified Dijkstra Algorithm starts by selecting the two

© 2023 The Authors.

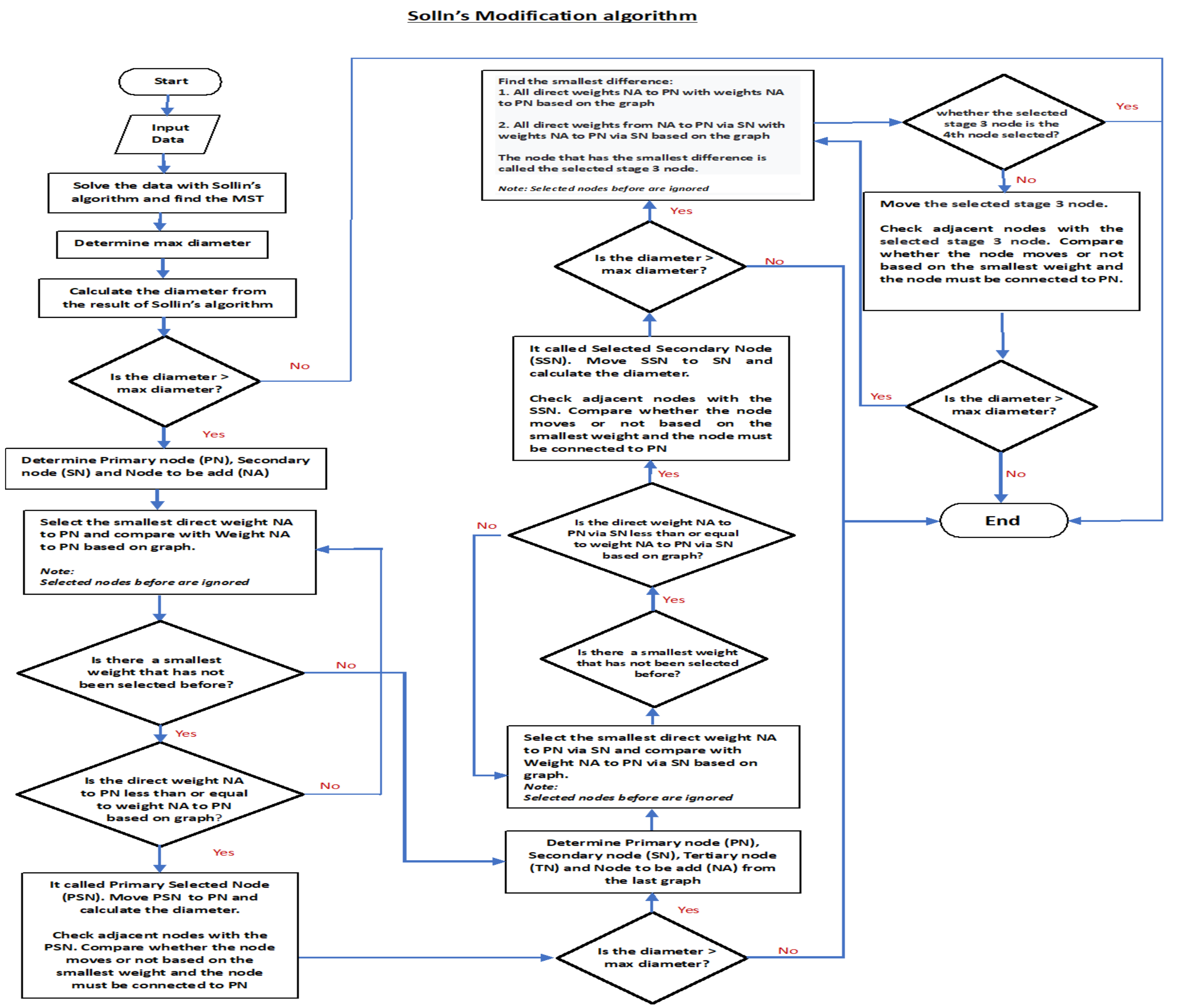

Figure 3. Flowchart Modified Sollin Algorithm

smallest costs in the table list and putting them in set T (T can be a tree or forest), and the vertices adjacent to those edges in the set V. If the number of components in T is more than one, then choose the next utilized edge in the table, and put in T, and corresponding adjacent vertices in V. Check if $|T| = n$-1. If yes, stop, otherwise continue the step. The next step is giving labels to components in T (there are two labels, one is 0 and the other is 1), then searching for the smallest edge. Check if both adjacent vertices in the smallest edge are in V. If yes, connect component 0 and component 1 by adding the smallest edge (i, j) so that the components 0 and 1 are connected. Relabeled the component as 0. If only one vertex is adjacent to (i, j) in V (suppose i is already in V), then put j in V, edge (i, j) in T, and label edge (i, j) with the same label as its connected component. Do those steps until $|T| = n$-1 and the component has only one label. Figure 2 illustrates the procedure of the Modified Dijkstra Algorithm.

3.2 The Modified Sollin Algorithm

Before starting the Modified Sollin Algorithm, we run the Sollin Algorithm to determine the MST as preprocessing. Thus, the preprocessing in the Modified Sollin is to find the MST. Using the MST gained in the preprocessing, the algorithm starts by checking the path length for every pair of vertices. Note that the path length in this case is the number of edges connecting every two vertices, not the total cost of the path. The value of

© 2023 The Authors.

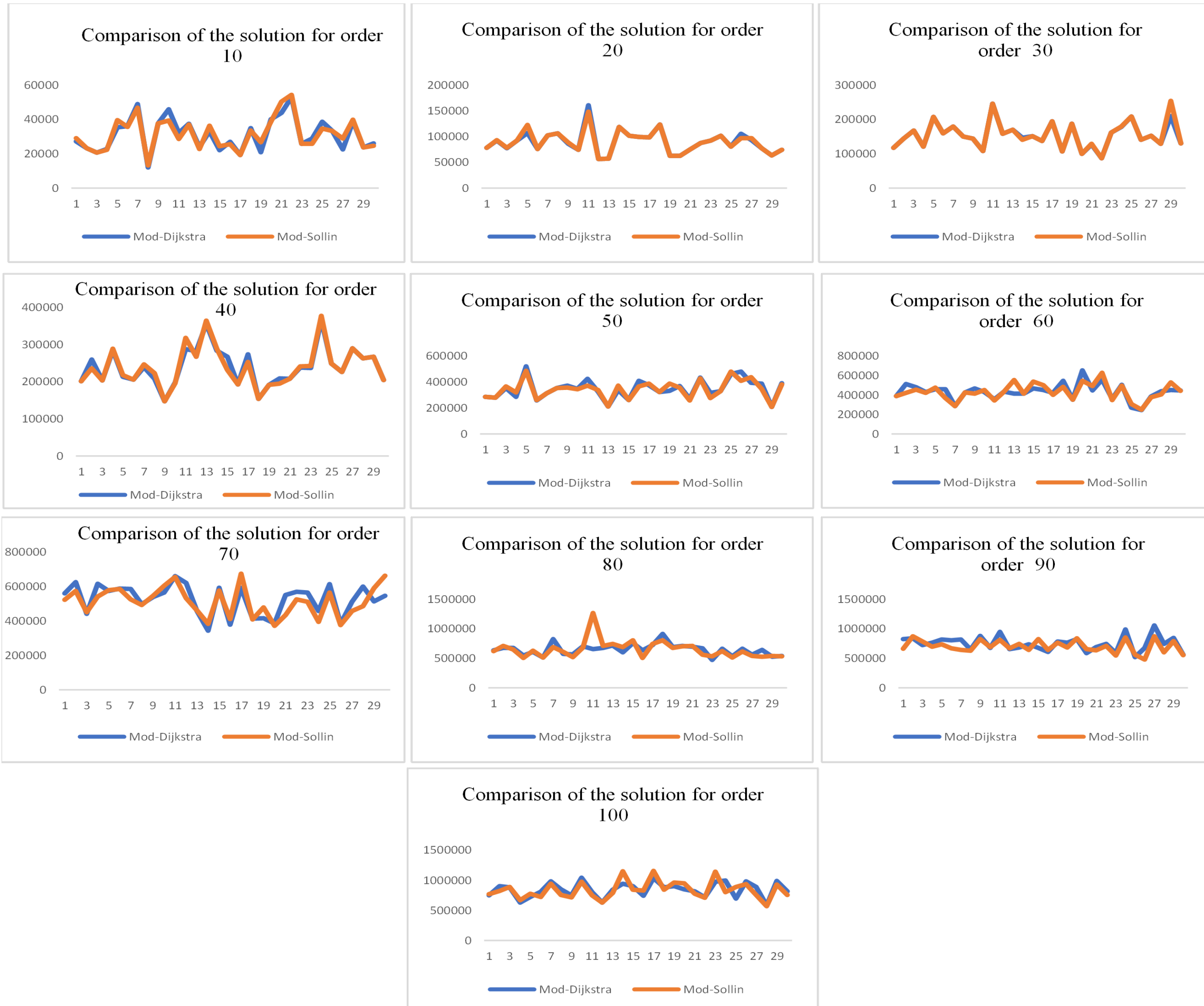

Figure 4. The Comparison of the Solution of Modified Dijkstra and Modified Sollin Algorithms for Vertex Order 10 to 100

d_{max} for graphs of orders 10, 20, and 30 is $\frac{n}{2}$, while for orders 40 to 100 is 15. If the path length is $>d_{max}$, then the path revision must be done to reduce the path length. The idea of reducing path length is due to the longer the path the higher the cost. If the path length is $>d_{max}$, then the algorithm will check the highest degree vertices, denote it as the primary vertex, and put in set V. If not, then there are no modifications made. To do the path revision, remove the highest cost edge that is farthest from the primary vertex, and add the new edge that connects the adjacent leaf on that farthest edge to the smallest edge connecting to the secondary vertex. If there is more than one primary vertex, then denote also as primary and put it in V. Next, check the secondary vertices which are the vertices that are adjacent to the primary vertex/vertices, and choose the smallest edges connecting every primary or secondary vertex, and put the edges that connect them in T. If all vertices already in primary or secondary, then check if $T = |n-1|$. If yes, then stop, otherwise determine the unconnected vertices. Calculate the smallest cost $d(u, v)$ from the primary vertex u to every unconnected vertices v, $d(u, v) = \min\{\frac{c_{uv}}{w(P_{uv})}\}$, where c_{uv} is the cost of edge (u, v), and $w(P_{uv})$ is the cost of the shortest path that connects primary vertex u to vertex v. Choose the smallest edge in the calculation and connect. Do that step until every unconnected vertex is connected and $T = |n-1|$. Figure 3 shows the flowchart of the Modified Sollin algorithm.

We implement both algorithms on complete graphs of orders 10 to 100. There are thirty problems for each vertex order.

Table 1 shows the result of implementing 300 problems. It shows that the average solutions gained from the Modified Dijsktra Algorithm perform better than the Modified Sollin on orders 10, 20, 30, 60, and 80, while the Modified Sollin performs better than the Modified Dijkstra on orders 40, 50, 70, 90, and 100. Figure 4 shows the comparative solutions of both algorithms for vertex order 10 to 100. From the comparative solutions for orders 20 and 30, it can be seen that the perfor-

© 2023 The Authors.

mance of those two algorithms is quite similar where the line showing solutions gained by the Modified Dijkstra Algorithm (blue line), and solutions gained by the Modified Sollin (orange line) almost collide in every problem.

4. CONCLUSION

Based on the discussion above, we conclude that implemented on the data problems, the performance of the Modified Dijkstra and the Modified Sollin Algorithms are slightly similar. On orders 10, 20, 30, 60, and 80, the Modified Dijkstra Algorithm performs better than the Modified Sollin, however on orders 40, 50, 70, 90, and 100, the Modified Sollin performs better.

5. ACKNOWLEDGMENT

This study is supported by the Directorate of Research and Community Service through a postgraduate research grant no: 2192/UN26.21/PN/2023. The authors wish to thank you for that fund.

REFERENCES

Borůvka, O. (1926). O Jistém Problému Minimálním. *Práce Moravské Přírodovědecké Společnosti*, **3**(3); 37–58

Brandes, U. and S. Cornelsen (2009). Phylogenetic Graph Models beyond Trees. *Discrete Applied Mathematics*, **157**(10); 2361–2369

Campos, R. and M. Ricardo (2008). A Fast Algorithm for Computing Minimum Routing Cost Spanning Trees. *Computer Networks*, **52**(17); 3229–3247

de Nooy, W. (2009). Social Network Analysis, Graph Theoretical Approaches To. *Encyclopedia of Complexity and System Science*; 8231–8245

Elumalai, A. (2020). Graph Theory Applications in Computer Science and Engineering. *Malaya Journal of Matematik*, (2); 4025–4027

Fischetti, M., G. Lancia, and P. Serafini (2002). Exact Algorithms for Minimum Routing Cost Trees. *Networks: An International Journal*, **39**(3); 161–173

Hieu, N. M., P. Quoc, and N. D. Nghia (2011). An Approach of Ant Algorithm for Solving Minimum Routing Cost Spanning Tree Problem. In *Proceedings of the 2nd Symposium on Information and Communication Technology*; 5–10

Huson, D. H. and D. Bryant (2006). Application of Phylogenetic Networks in Evolutionary Studies. *Molecular Biology and Evolution*, **23**(2); 254–267

Julstrom, B. A. (2001). The Blob Code: A Better String Coding of Spanning Trees for Evolutionary Search. In *Genetic and Evolutionary Computation Conference Workshop Program*. Morgan Kaufmann San Francisco; 256–261

Julstrom, B. A. (2005). The Blob Code is Competitive with Edge-Sets in Genetic Algorithms for the Minimum Routing Cost Spanning Tree Problem. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*; 585–590

Kannimuthu, S., D. Bhanu, and K. Bhuvaneshwari (2020). A Novel Approach for Agricultural Decision Making Using Graph Coloring. *SN Applied Sciences*, **2**; 1–6

Kawakura, S. and R. Shibasaki (2018). Grouping Method Using Graph Theory for Agricultural Workers Engaging in Manual Tasks. *Journal of Advanced Agricultural Technologies Vol*, **5**(3); 173–181

Kawatra, R. (2002). A Multiperiod Degree Constrained Minimal Spanning Tree Problem. *European Journal of Operational Research*, **143**(1); 53–63

Kruskal, J. B. (1956). On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, **7**(1); 48–50

Masone, A., M. E. Nenni, A. Sforza, and C. Sterle (2019). The Minimum Routing Cost Tree Problem: State of the Art and a Core-Node Based Heuristic Algorithm. *Soft Computing*, **23**; 2947–2957

Nøjgaard, N. (2020). *Graph Theoretical Problems in Life Sciences*. Ph.D. thesis, University of Greifswald, Germany

Prim, R. C. (1957). Shortest Connection Networks and Some Generalizations. *The Bell System Technical Journal*, **36**(6); 1389–1401

Raidl, G. R. and B. A. Julstrom (2003). Edge Sets: An Effective Evolutionary Coding of Spanning Trees. *IEEE Transactions on Evolutionary Computation*, **7**(3); 225–239

Sari, R. P., Wamiliana, A. Junaidi, and W. Susanty (2022). The Diameter and Maximum Link of the Minimum Routing Cost Spanning Tree Problem. *Science and Technology Indonesia*, **7**(4); 481–485

Segal, M. and O. Tzfaty (2022). Finding Bounded Diameter Minimum Spanning Tree in General Graphs. *Computers & Operations Research*, **144**; 105822

Singh, A. (2008). A New Heuristic for the Minimum Routing Cost Spanning Tree Problem. In *2008 International Conference on Information Technology*. IEEE; 9–13

Singh, A. and S. Sundar (2011). An Artificial Bee Colony Algorithm for the Minimum Routing Cost Spanning Tree Problem. *Soft Computing*, **15**; 2489–2499

Singh, R. P. (2014). Application of Graph Theory in Computer Science and Engineering. *International Journal of Computer Applications*, **104**(1); 10–13

Tan, Q. P. (2012a). A Genetic Approach for Solving Minimum Routing Cost Spanning Tree Problem. *International Journal of Machine Learning and Computing*, **2**(4); 410

Tan, Q. P. (2012b). A Heuristic Approach for Solving Minimum Routing Cost Spanning Tree Problem. *International Journal of Machine Learning and Computing*, **2**(4); 406

Wamiliana, W., M. Usman, F. Elfaki, and M. Azram (2015a). Some Greedy Based Algorithms for Multi Periods Degree Constrained Minimum Spanning Tree Problem. *ARPN Journal of Engineering and Applied Sciences*, **10**(21); 10147–10152

Wamiliana, W., M. Usman, D. Sakethi, R. Yuniarti, and A. Cucus (2015b). The Hybrid of Depth First Search Technique and Kruskal's Algorithm for Solving the Multiperiod Degree Constrained Minimum Spanning Tree Problem. In

© 2023 The Authors.

2015 4th International Conference on Interactive Digital Media (ICIDM). IEEE; 1–4

Wamiliana, W., M. Usman, W. Warsito, W. Warsono, and J. I. Daoud (2020). Using Modification of Prim's Algorithm and Gnu Octave and to Solve the Multiperiods Installation Problem. *IIUM Engineering Journal*, **21**(1); 100–112

© 2023 The Authors.