2016 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2016, Yugawara, Japan, November 27th-30th, 2016

A Construction Method for the Steiner Tree Problem using Betweenness Centrality

Misa Fujita†, Takayuki Kimura and Kenya Jin'no‡

†Graduate School of Electronics, Information and Media Engineering, Nippon Institute of Technology
4-1 Gakuendai, Miyashiro, Minami-Saitama, Saitama 345-8501, Japan
‡Department of Electrical and Electronic Engineering, Nippon Institute of Technology
4-1 Gakuendai, Miyashiro, Minami-Saitama, Saitama 345-8501, Japan
Email: 2158019@estu.nit.ac.jp, {tkimura, jinno}@nit.ac.jp

Abstract—Given an undirected weighted graph $G = (V, E, c)$ and a set $T \in V$, where V is the set of vertices, E is the set of edges, c is the cost function, and T is the subset of terminal vertices, the Steiner tree is a subgraph that doesn't have a loop and connects all terminals. The Steiner tree problem in graphs is to find the minimum cost Steiner tree. The Steiner tree problem is one of the $\mathcal{NP}$-complete combinatorial optimization problems. Thus, approximate methods are usually employed for constructing the Steiner tree. In this study, the KMB algorithm, which is an efficient construction method for the Steiner tree problem, is enhanced by considering betweenness centrality. Results of numerical simulations indicate that our improved KMB algorithm shows good performance for the benchmark Steiner tree problems.

1. Introduction

Given an undirected weighted graph $G = (V, E, c)$ and a set of $T \in V$, where V is the set of vertices, E is the set of edges, c is a non-negative cost function, and T is a subset of terminal vertices, a subgraph that doesn't have a loop and connects all terminals is called a Steiner tree. A cost of the Steiner tree is the sum of the edge costs included in the tree. The Steiner tree problem in graphs is to find the minimum cost Steiner tree. Figure 1 shows examples of the Steiner tree. In Fig. 1, circles express vertices and lines express edges. Black circles express terminal vertices and black lines express the edges in the Steiner tree. A network has many Steiner trees. Among them, Fig. 1(c) is the minimum cost Steiner tree for the given network.

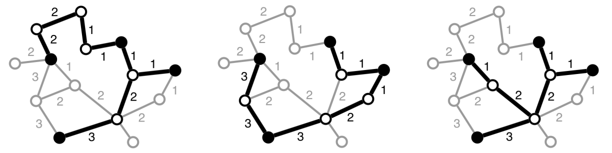

(a) Cost:13 (b) Cost:14 (c) Cost:10

Figure 1: Examples of the Steiner tree

The Steiner tree problem is applied to various real-world problems, such as VLSI routing[1], wirelength estimation[2], and network routing[3]. Because the Steiner tree problem is one of the $\mathcal{NP}$-complete combinatorial optimization problems[4], the approximation method is usually used to obtain the Steiner tree. Most popular method to obtain the near-optimum Steiner tree is the KMB algorithm[5]. The KMB algorithm constructs the Steiner tree based on the shortest path between terminals and the minimum spanning tree of all the terminals. The calculation cost of the KMB algorithm becomes $O(|V|^2)$ where $|V|$ is the number of vertices. The KMB algorithm can obtain small cost Steiner tree in short time. However, if there are some shortest paths between the terminals, an obtained cost of the Steiner tree changes depending on the selected paths.

From this view point, we propose a construction method of the Steiner tree using the betweenness centrality in this study. The cost of the Steiner tree obtained by the KMB algorithm using given edge cost is compared with the one using the betweenness centrality of nodes and edges. Results of numerical simulations indicate that our propose method can obtain the small cost Steiner tree.

2. Steiner tree problem

If the number of terminals $|T|$ becomes 2, the Steiner tree problem changes the shortest path problem. In addition, in the case by $|T| = |V|$, the Steiner tree problem becomes minimum spanning tree problem. On the other hand, in the case by $2 < |T| < |V|$, finding minimum cost tree becomes the Steiner tree problem.

To mathematically model this problem, we first define a decision variable, $x(e_i)$, that is defined as follows:

$$x(e_i) = \begin{cases} 1 & (e_i \in E_{ST}), \\ 0 & (\text{otherwise}), \end{cases} \tag{1}$$

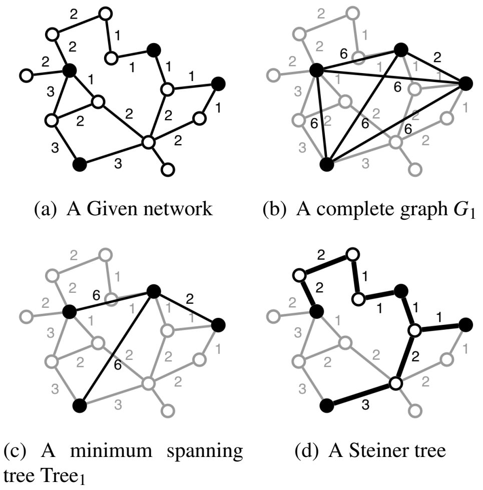
where e_i is the ith edge in the network, E_{ST} is the set of edges included in a Steiner tree. By using the decision variables, an objective function for the Steiner tree problem in graphs is then defined as follows:

$$\min \sum_{i=1}^{|E|} c(e_i)x(e_i), \tag{2}$$

where $|E|$ is the total number of edges, $c(e_i)$ is a cost of the ith edge. If the ith edge, e_i, is included in the Steiner tree, a corresponding decision variable takes 1, and the cost of the edge e_i is added to the total cost of the Steiner tree. In other words, the Steiner tree problem is to find the best combination of $x(e_i)$.

3. KMB algorithm

In this study, we use the KMB algorithm[5] to construct the Steiner tree. The KMB algorithm is one of the efficient approximation algorithms for the Steiner tree problems. To obtain the Steiner tree using the KMB algorithm, we first construct a complete graph $G_1 = (T_1, E_1, c_1)$ from G and T (Fig. 2(b)). In the complete graph G_1, all the vertices are terminals. The cost of the edges in G_1 corresponds to the cost of the shortest path between these terminals. Next, we construct a minimum spanning tree Tree$_1$ from the complete graph G_1 (Fig. 2(c)). Finally, edges used in the Tree$_1$ are replaced by the shortest paths connecting to the corresponding vertices in original graph (Fig. 2(d)). An example Steiner tree using the KMB algorithm is shown in Fig. 2.


(a) A Given network (b) A complete graph G_1

(c) A minimum spanning tree Tree$_1$ (d) A Steiner tree

Figure 2: An example Steiner tree using the KMB algorithm

4. KMB algorithm using betweenness centrality

In the KMB algorithm, if there are some shortest paths that have the same cost connecting to two vertices, the cost of the obtained Steiner tree changes depending on the selected paths (Fig. 3). The cost of the obtained Steiner tree might be longer if an undesirable path is selected. Thus, it is important to select better shortest paths which various shortest paths go through as much as possible.

From this view point, we use the network centrality for constructing the shorter Steiner tree. Although various types of network centrality have already been proposed, we employ the betweenness centrality[6, 7, 8] in this study.

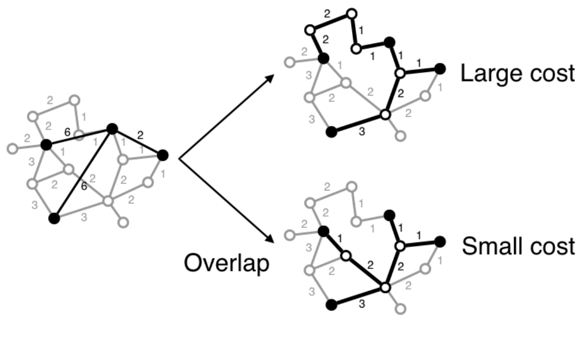


Figure 3: A poor point of the KMB algorithm

The betweenness centrality expresses how many times a vertex appears on every shortest path in the graph. We modify the cost of link to combine the original cost and the betweenness cost.

The betweenness centrality of a vertex v is defined as follows:

$$\mathrm{bc}(v) = \frac{\displaystyle\sum_{s=1}^{|V|}\sum_{g=1,g\neq s}^{|V|-1} \frac{P_{v(s,g)}}{P_{(s,g)}}}{(|V|-1)(|V|-2)/2}, \tag{3}$$

where s is a start vertex and g is a goal vertex of the shortest path, $|V|$ is the number of vertices, $P_{(s,g)}$ is the number of shortest paths between s and g, $P_{v(s,g)}$ is the number of shortest paths between s and g that goes through the vertex v.

If a vertex has high betweenness centrality, the vertex frequently lies on the shortest path. By selecting the path includes the high betweenness vertices, the cost of the obtained Steiner tree may become small.

To use the betweenness centrality of vertices to edges, the betweenness cost of the edge(v_i, v_j), eb(v_i, v_j), is defined as follows:

$$\mathrm{eb}(v_i, v_j) = \frac{\mathrm{bc}(v_i) + \mathrm{bc}(v_j)}{2}, \tag{4}$$

where v_i is the ith vertex, v_j is the jth vertex, bc(v) is the betweenness centrality of the vertex v.

We then define a new cost of edge by considering the given edge cost and the betweenness cost as follows:

$$c_{\mathrm{new}}(e_i) = \alpha c(e_i) + (1-\alpha)\frac{1}{\mathrm{bc}(e_i)}, \tag{5}$$

where $c(e_i)$ is the normalized given edge cost of the ith edge, bc(e_i) is the normalized betweenness cost of the ith edge, α is a controlling parameter which determines the priority between the given edge cost and the betweenness cost. Although the betweenness centrality is usually calculated by using the all the shortest paths in the graph, vertices placed at the outside from every terminal are not necessary to include the Steiner tree. We then limit the area of the graph for calculating the betweenness centrality in this study. Figure 4 illustrates the given network and the vertices for calculating the betweenness centrality(inside the

Table 1: Result of numerical simulations

		KMB algorithm			KMB algorithm using Betweenness centrality							
					Betweenness centrality				Limited betweenness centrality			
Name	Opt	Min	Max	Mid	α	Min	Max	Mid	α	Min	Max	Mid
dmxa0296	344	374	387	374	0.6	**364**	**364**	**364**	0.1	**352**	**352**	**352**
dmxa0368	1017	1036	1046	1046	0.1	**1031**	**1031**	**1031**	0.1	**1031**	**1031**	**1031**
dmxa0454	914	991	1009	1004	0.1	**958**	**958**	**958**	0.1	**953**	**953**	**953**
dmxa0628	275	297	297	297	0.2	**277**	**277**	**277**	0.1	**297**	**297**	**297**
dmxa0734	506	537	541	537	0.2	**537**	**537**	**537**	0.1	**537**	**537**	**537**
dmxa0848	594	637	657	644	0.1	**632**	**632**	**632**	0.1	**632**	**632**	**632**
dmxa0903	580	639	660	660	0.3	645	**645**	**645**	0.1	**632**	**632**	**632**
dmxa1010	1488	1538	1561	1551	0.1	1546	**1546**	**1546**	0.1	1546	**1546**	**1546**
dmxa1109	454	473	498	483	0.2	**473**	**473**	**473**	0.1	**468**	**468**	**468**
dmxa1200	750	821	834	821	0.1	**769**	**769**	**769**	0.1	**773**	**773**	**773**
dmxa1304	311	329	334	334	0.1	**316**	**316**	**316**	0.1	**316**	**316**	**316**
dmxa1516	508	513	533	523	0.3	528	**528**	528	0.1	528	**528**	528
dmxa1721	780	825	840	830	0.2	**794**	**794**	**794**	0.1	**794**	**794**	**794**
dmxa1801	1365	1484	1540	1530	0.4	**1479**	**1479**	**1479**	0.1	**1466**	**1466**	**1466**

black square). In Fig. 4, black circles express the terminal vertices. Here, V_{lim} is the set of vertices in the limited area, and E_{lim} is the set of edges connected to two vertices in V_{lim}. We call the betweenness centrality calculated using the limited graph $G_{\text{lim}} = (V_{\text{lim}}, E_{\text{lim}}, c)$ as a limited betweenness centrality.

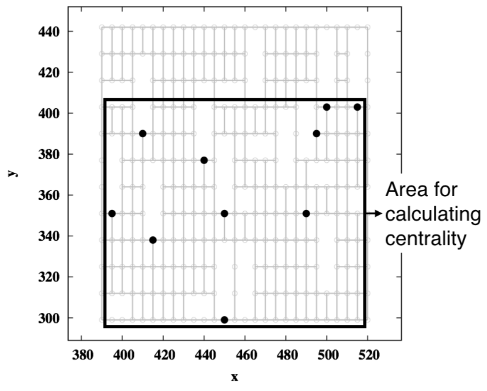


Figure 4: A limited graph

5. Numerical experiments

To compare the performance of the conventional KMB algorithm with our proposed method, we used the benchmark problems provided in SteinLib[9]. The value of the controlling parameter α in our proposed method is set to the optimal value by the preliminary experiments. A source vertex of the Steiner tree is randomly selected among the terminals.

Table 1 shows the cost of the obtained Steiner trees by the conventional KMB algorithm and the proposed method for the test sets DMXA in SteinLib[9]. From the results, our proposed method obtained the shorter Steiner tree than the conventional KMB algorithm. In addition, the method using betweenness centrality and that by limited betweenness centrality show similar performance. However, the limited betweenness centrality has less calculation cost than the original betweenness centrality. Thus, we can say that the method using the limited betweenness centrality is better constructing method than that by the original betweenness centrality.

Figure 5 shows the performance of our propose method for various α cases. In Fig. 5, the constructing method by the limited betweenness centrality shows good performance almost the α cases. This is because the edges in the outside of limited area are removed.

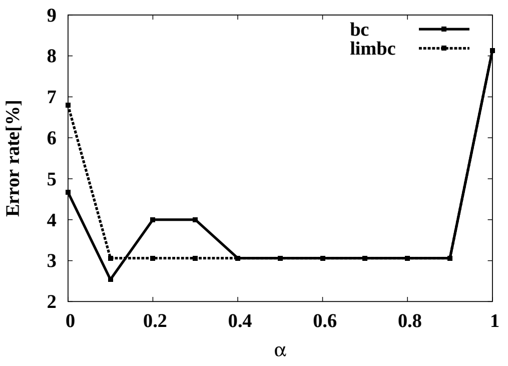
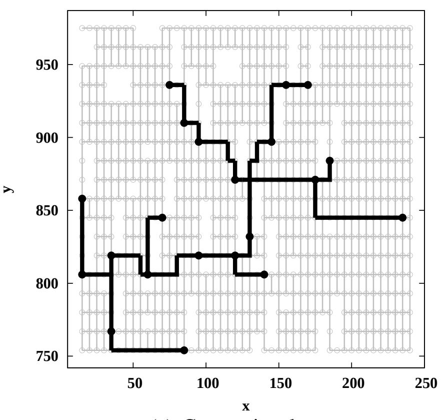
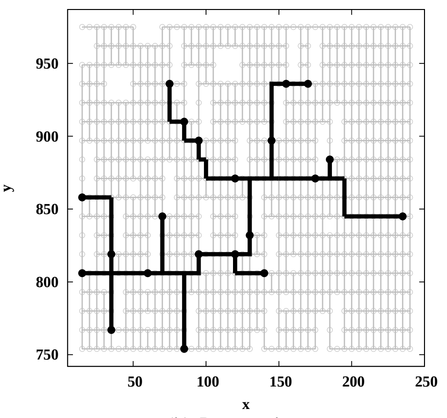


Figure 5: An error rate of our propose method for various α cases (using dmxa1200)

Figure 6 shows the topologies of Steiner tree obtained by the conventional method and our propose method. Black circles denote the terminals and black lines denote edges in the Steiner tree. In Fig. 6, these Steiner trees have different topologies.



(a) Conventional



(b) Proposed

Figure 6: Steiner trees of dmxa1200 obtained by the conventional and propose methods

6. Conclusions

In this study, we try to change the edge cost by using the betweenness centrality for constructing the small cost Steiner tree. We evaluated the performance of our propose method by the testsets in the SteinLib. From the results of numerical simulations, the propose method obtains the small cost Steiner tree stably in comparison to the conventional KMB algorithm. The conventional KMB algorithm often finds larger cost Steiner tree if there are some shortest paths between the nodes. By selecting the shortest path which has high betweenness centrality, the cost of the Steiner tree becomes small. In addition, although the calculation cost of the betweenness centrality becomes large as the size of network increases, we can solve this undesirable problem by using the limited betweenness centrality.

The performance of the KMB algorithm can be enhanced by selecting the shortest paths between the terminals which various shortest paths commonly use. Because our propose method changed the costs of edges to improve the performance of the KMB algorithm, obtaining the smallest cost path by using the original edge cost is impossible. If we employ the edge which has original edge cost and high betweenness centrality, the cost of the obtained Steiner tree might become small. In the future work, we try to evaluate our method by the network whose links have original cost and high betweenness centrality to construct the small cost Steiner tree.

The research of T.K. was partially supported by a Grant-in-Aid for Young Scientists(B) from JSPS (No. 16K21327).

References

[1] A. B. Kahng and G. Robins, "On Optimal interconnections for VLSI," *Kluwer*, Dordrecht, 1995.

[2] A. Caldwell, A. B. Kahng, S. Mantik, I. Markov and A. Zelikocsky, "On wirelength estimations for row-based placement," *Proceedings of the International Symposium on Physical Design*, pp. 4–11, 1981.

[3] B. Korte, H. J. Promel and A. Steger, "Steiner trees in VLSI-layout," *in Paths, Flows, and VLSI-Layout*, *Springer*, pp. 185–214, 1990.

[4] R. M. Karp, "Reducibility among Combinatorial Problems," *Complexity of Computer Computations*, pp. 85–103, 1972.

[5] L. Kou, G. Markowsky and L. Berman, "A Fast Algorithm for Steiner Trees," *Acta Informatica*, Vol. 15, pp. 141–145, 1981.

[6] A. Shimbel, "Structural Parameters of Communication Networks," *The bulletin of mathmatical biophysics*, Vol. 15, No. 4, pp. 501–507, 1953.

[7] L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, pp. 35–41, 1977.

[8] U. Brandes, "A Faster Algorithm for Betweenness Centrality," *Journal of Mathmatical Sociology*, Vol. 25, No. 2, pp. 163–177, 2001.

[9] T. Koch, A. Martin and S. Vob, "SteinLib: An Updated Library on Steiner Tree Problems in Graphs," Available online at: "http://elib.zib.de/steinlib," 2001. (31st, Jul. 2016, access)