The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 14, No. 4, pp.233-241, Aug. 31, 2014. pISSN 2289-0238, eISSN 2289-0246

http://dx.doi.org/10.7236/JIIBC.2014.14.4.233
JIIBC 2014-4-33

2-간선 연결 그래프를 사용한 최소신장트리 알고리즘 제안

Proposal of Minimum Spanning Tree Algorithm using 2-Edges Connected Grap

이상운*

Sang-Un Lee*

요 약 본 논문은 원 그래프를 2-간선 연결 그래프로 단순화하고, 사이클 속성을 적용하여 최소신장트리를 빠르게 얻는 알고리즘을 제안하였다. Borůvka 알고리즘은 정점 (v) 당 최소 가중치 간선 (e)을 1개씩 선택하는 1-간선 연결 그래프에 대해 사이클 속성을 적용하여 부분신장트리를 얻는다. 추가적으로 절단속성을 적용하여 부분신장트리를 연결하는 최소 가중치 간선을 선택한다. Kruskal 알고리즘은 그래프의 모든 간선을 대상으로 오름차순으로 절단 속성을 적용한다. 역-삭제 알고리즘은 내림차순으로 사이클 속성을 적용한다. Borůvka, Kruskal과 역-삭제 알고리즘은 모든 간선들을 대상으로 하기 때문에 항상 $|e|$ 회 수행된다. 제안된 알고리즘은 첫 번째로, 정점 당 최소 가중치 간선을 2개씩 선택하는 2-간선 연결 그래프를 얻는다. 두 번째로, 2-간선 연결 그래프에 대해 사이클 속성을 적용하여 $|e|=|v|-1$ 일 때 알고리즘을 종료시켰다. 제안된 방법들을 10개의 실제 그래프들에 적용한 결과 모두 최소신장트리를 얻는데 성공하였다. 또한, Borůvka, Kruskal과 역-삭제 알고리즘에 비해 수행 횟수를 60% 단축시켰다.

Abstract This paper suggests a fast minimum spanning tree algorithm which simplify the original graph to 2-edge connected graph, and using the cycling property. Borůvka algorithm firstly gets the partial spanning tree using cycle property for one-edge connected graph that selects the only one minimum weighted edge (e) per vertex (v). Additionally, that selects minimum weighted edge between partial spanning trees using cut property. Kruskal algorithm uses cut property for ascending ordered of all edges. Reverse-delete algorithm uses cycle property for descending ordered of all edges. Borůvka and Kruskal algorithms always perform $|e|$ times for all edges. The proposed algorithm obtains 2-edge connected graph that selects 2 minimum weighted edges for each vertex firstly. Secondly, we use cycle property for 2-edges connected graph, and stop the algorithm until $|e|=|v|-1$. For actual 10 benchmark data, The proposed algorithm can be get the minimum spanning trees. Also, this algorithm reduces 60% of the trial number than Borůvka, Kruskal and Reverse-delete algorithms.

Key Words : Minimum spanning tree, Cycle property, Cut property, 2-edges connected graph, Minimum weight edge

Ⅰ. 서 론

그래프 $G=(V,E)$는 정점들 (vertices, v)이 간선들 (edges, e)로 연결되어 있고 (connected), 간선들은 무방향성 (undirected)이며, 가중치를 갖는 (weighted) 경우, 모든 정점들 $|v|$ 을 연결하는 간선들 $|e|=|v|-1$ 의 가중

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자 : 2014년 4월 15일, 수정완료 : 2014년 7월 28일
게재확정일자 : 2014년 8월 8일

Received: 15 April, 2014 / Revised: 28 July, 2014 / Accepted: 8 August, 2014
*Corresponding Author: sulee@gwnu.ac.kr
Dept. of Multimedia Eng., Gangneung-Wonju National University, Korea

치 합 ($\Sigma w(e)$)이 최소이면서 사이클 (cycle)이 없는 트리를 최소신장트리 (minimum spanning tree, MST)라 한다. MST는 전기, 전화, 가스 또는 수도 등의 분야에 활용될 수 있다.[1]

MST는 절단 속성 (cut property)과 사이클 속성 (cycle property)을 갖고 있다.[1-2]

(속성 1) 절단 속성: 최소 가중치 간선 (minimum weight edge, MWE, $e_{\min}$)의 어느 한 정점이 부분신장트리 (partial spanning tree, PST) T_i의 절단 (cut)에 속해 있으면 $e_{\min}$은 MST에 포함된다."

(속성 2) 사이클 속성: "임의의 사이클의 최대 가중치 간선 (maximum weight edge, e_{cmax})은 MST에 포함되지 않는다."

MST를 구하는 대표적인 알고리즘으로는 Borůvka[3-4], Prim[5], Kruskal[6]과 역-삭제 (Reverse-Delete)[7]가 있다. Borůvka, Kruskal과 역-삭제 알고리즘은 간선을 선택하는 방법으로 항상 $|e|$ 회 수행되며, Prim 알고리즘은 정점을 한 번에 하나씩 선택하는 방법으로 항상 $|v|$ 회 수행된다. 4개 알고리즘 모두 "그래프의 모든 간선들의 가중치는 서로 다르다 (distinct)"는 가정에 근거하고 있다. 그러나 현실적으로 그래프의 간선들은 동일한 가중치를 가질 수 있다.

Borůvka 알고리즘은 1^{st}단계 (stage)에서 정점 당 1-MWE를 선택하여 얻은 1-간선 연결 그래프 (1-edge connected graph)에 대해 절단이나 사이클 속성을 적용하여 e_{cmax}를 제거하여 PST들을 얻는다. 2^{nd} 단계에서는 절단 속성을 적용하여 PST들을 연결하는 MWE를 선택한다. Kruskal 알고리즘은 오름차순으로 정렬된 그래프의 모든 간선들 $|e|$ 에 대해 절단 속성을 만족하는 간선을 MST 간선 (e_{mst})에 포함 시킨다. 역-삭제 알고리즘은 Kruskal과 반대로 내림차순으로 정렬된 그래프의 모든 간선들 $|e|$ 에 대해 사이클 속성을 적용하여 e_{mst}에서 e_{cmax}를 배제 (제거)한다.

본 논문은 그래프의 간선을 대상으로 하는 Borůvka, Kruskal과 역-삭제 알고리즘의 수행 횟수 $|e|$ 를 단축시키는 알고리즘을 제안한다. 제안되는 알고리즘은 먼저, 원 그래프 (original graph)의 정점 당 2개의 MWE를 선택하여 2-간선 연결 그래프 (2-edges connected graph)를 얻는다. 1^{st} MWE를 $e_{\min 1}$, 2^{nd} MWE를 $e_{\min 2}$라 하자. 2-간선 연결 그래프를 대상으로 내림차순으로 사이클 속성을 만족하는 e_{cmax} 간선을 e_{mst}에서 배제 시키거나 오름차순으로 절단 속성을 만족하는 간선을 e_{mst}에 포함시킨다. 알고리즘은 $|e|=|v|-1$일 때 종료한다.

2장에서는 간선을 대상으로 하는 Borůvka, Kruskal과 역-삭제 알고리즘을 실제 그래프에 적용하여 MST를 찾는 방법을 고찰해 본다. 3장에서는 간선을 대상으로 알고리즘 수행 횟수를 줄일 수 있는 2-간선 연결 그래프 방법을 제안한다. 4장에서는 실제 그래프들을 대상으로 제안된 알고리즘의 효율성을 검증해 본다.

II. 관련 연구와 연구 배경

무방향 그래프 $G=(V,E)$에서 두 정점 x와 y를 연결하는 간선은 무방향성과 가중치를 갖고 있어 $w\{x,y\}=w\{y,x\}$로 표기한다[8]. MST는 다음 3가지 조건을 만족하여야만 한다.

(조건 1) 선택된 간선들의 가중치 합이 최소가 되어야 한다.

(조건 2) $|v|$ 개의 정점들이 모두 연결되어야 한다.

(조건 3) 사이클이 존재하지 않아야 한다.

위 조건 (1)과 (3)을 만족시키기 위해 절단 속성이나 사이클 속성을 적용한다. 조건 (2)는 MST가 트리이므로 트리의 $|e|=|v|-1$이다. 따라서 $|e|=|v|-1$일 때 알고리즘을 종료시키면 된다. 그러나 Borůvka, Kruskal과 역-삭제 알고리즘은 위 조건 (1)과 (3)은 만족시키지만 조건 (2)는 적용하지 않고 그래프의 모든 간선들을 대상으로 한다.

G_1 그래프를 대상으로 Borůvka, Kruskal과 역-삭제 알고리즘으로 MST를 구하여 본다. G_1 그래프는 Wikipedia[1]에서 인용되었다.

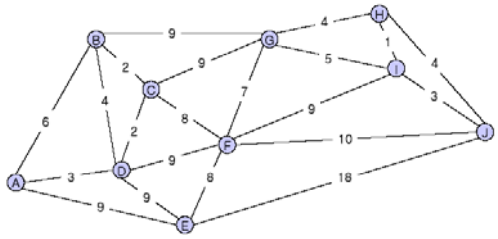


그림 1. G_1 그래프
Fig. 1. G_1 graph

Borůvka 알고리즘으로 G_1 그래프의 MST를 구하는 과정은 표 1에 제시되어 있다.

Kruskal 알고리즘은 절단속성을 만족시키는 간선을 MST에 포함시키는 방법으로, G_1 그래프의 MST를 구하는 과정은 표 2에 제시되어 있다.

표 1. G_1 그래프의 Borůvka 알고리즘
Table 1. Borůvka algorithm for G_1 graph

단계	간선 선택	T (변경 전)	사이클	T (변경 후)	e_{mst}
1^{st} Stage (S_1)	1{H,I}	∅	×	{**H,I**}	1{H,I}
	2{B,C}	{H,I}	×	{H,I}, {**B,C**}	2{B,C}
	3{C,D}	{H,I}, {B,**C**}	×	{H,I}, {B,**C,D**}	2{C,D}
	3{A,D}	{H,I}, {B,C,**D**}	×	{H,I}, {**A**,B,C,**D**}	3{A,D}
	3{I, J}	{H,I}, {A,B,C,D}	×	{H,**I,J**}, {A,B,C,D}	3{J, I}
	4{G,H}	{**H**,I,J}, {A,B,C,D}	×	{**G,H**,I,J}, {A,B,C,D}	4{G,H}
	7{F,G}	{**G**,H,I,J}, {A,B,C,D}	×	{**F,G**,H,I,J}, {A,B,C,D}	7{F,G}
	8{E,F}	{**F**,G,H,I,J}, {A,B,C,D}	×	{**E,F**,G,H,I,J}, {A,B,C,D}	8{E,F}
2^{nd} Stage (S_2)	4{B,D}	{E,F,G,H,I,J}, {A,**B,C,D**}	○	{E,F,G,H,I,J}, {A,B,C,D}	–
	4{H,J}	{E,F,G,**H,I,J**}, {A,B,C,D}	○	{E,F,G,H,I,J}, {A,B,C,D}	–
	5{G,I}	{E,F,**G,H,I,J**}, {A,B,C,D}	○	{E,F,G,H,I,J}, {A,B,C,D}	–
	6{A,B}	{E,F,G,H,I,J}, {**A,B**,C,D}	○	{E,F,G,H,I,J}, {A,B,C,D}	–
	8{C,F}	{E,**F**,G,H,I,J}, {A,B,**C**,D}	×	{A,B,**C**,D,E,**F**,G,H,I,J}	8{C,F}
	9{A,E}	{**A**,B,C,D,**E**,F,G,H,I,J}	○	{A,B,C,D,E,F,G,H,I,J}	–
	9{B,G}	{A,**B**,C,D,E,F,**G**,H,I,J}	○	{A,B,C,D,E,F,G,H,I,J}	–
	9{C,G}	{A,B,**C**,D,E,F,**G**,H,I,J}	○	{A,B,C,D,E,F,G,H,I,J}	–
	9{D,E}	{A,B,C,**D,E**,F,G,H,I,J}	○	{A,B,C,D,E,F,G,H,I,J}	–
	9{D,F}	{A,B,C,**D,E,F**,G,H,I,J}	○	{A,B,C,D,E,F,G,H,I,J}	–
	9{F, I}	{A,B,C,D,E,**F**,G,H,I,J}	○	{A,B,C,D,E,F,G,H,I,J}	–
	10{F,J}	{A,B,C,D,E,**F**,G,H,**I,J**}	○	{A,B,C,D,E,F,G,H,I,J}	–
	18{E,J}	{A,B,C,D,**E,F**,G,H,**I,J**}	○	{A,B,C,D,E,F,G,H,I,J}	–

표 2. G_1 그래프의 Kruskal 알고리즘
Table 2. Kruskal algorithm for G_1 graph

S	T (변경 전)	사이클	T (변경 후)	e_{mst}
1{H, I}	{ ∅ }	×	{**H,I**}	1{H, I}
2{B,C}	{H,I}	×	{H,I}, {**B,C**}	2{B,C}
2{C,D}	{H,I}, {B,**C**}	×	{H,I}, {B,**C,D**}	2{C,D}
3{A,D}	{H,I}, {B,C,**D**}	×	{H,I}, {**A**,B,C,**D**}	3{A,D}
3{I, J}	{H,**I**}, {A,B,C,D}	×	{H,**I,J**}, {A,B,C,D}	3{I, J}
4{B,D}	{H,I,J}, {A,**B**,C,**D**}	○	{H,I,J}, {A,B,C,D}	–
4{G,H}	{**H**,I,J}, {A,B,C,D}	×	{**G,H**,I,J}, {A,B,C,D}	4{G,H}
4{H, J}	{G,**H,I,J**}, {A,B,C,D}	○	{G,H,I,J}, {A,B,C,D}	–
5{G, I}	{**G**,H,I,J}, {A,B,C,D}	○	{G,H,I,J}, {A,B,C,D}	–
6{A,B}	{G,H,I,J}, {**A,B**,C,D}	○	{G,H,I,J}, {A,B,C,D}	–
7{F,G}	{**G**,H,I,J}, {A,B,C,D}	×	{**F,G**,H,I,J}, {A,B,C,D}	7{F,G}
8{C,F}	{**F**,G,H,I,J}, {A,B,**C**,D}	×	{A,B,**C**,D,**F**,G,H,I,J}	8{C,F}
8{E,F}	{A,B,C,D,**F**,G,H,I,J}	×	{A,B,C,D,**E,F**,G,H,I,J}	8{E,F}
9{A,E}	{**A**,B,C,D,**E**,F,G,H,I,J}	○	{A,B,C,D,E,F,G,H,I,J}	–
9{B,G}	{A,**B**,C,D,E,F,**G**,H,I,J}	○	{A,B,C,D,E,F,G,H,I,J}	–
9{C,G}	{A,B,**C**,D,E,F,**G**,H,I,J}	○	{A,B,C,D,E,F,G,H,I,J}	–
9{D,E}	{A,B,C,**D,E**,F,G,H,I,J}	○	{A,B,C,D,E,F,G,H,I,J}	–
9{D,F}	{A,B,C,**D,E,F**,G,H,I,J}	○	{A,B,C,D,E,F,G,H,I,J}	–
9{F, I}	{A,B,C,D,E,**F**,G,H,**I**,J}	○	{A,B,C,D,E,F,G,H,I,J}	–
10{F,J}	{A,B,C,D,E,**F**,G,H,I,**J**}	○	{A,B,C,D,E,F,G,H,I,J}	–
18{E,J}	{A,B,C,D,**E**,F,G,H,I,**J**}	○	{A,B,C,D,E,F,G,H,I,J}	–

역-삭제 알고리즘은 사이클 속성을 만족하는 e_{cmax}를 e_{mst}에서 배제 (삭제)하는 방법으로, G_1 그래프의 MST를 구하는 과정은 표 3에 제시되어 있다.

표 3. G_1 그래프의 역-삭제 알고리즘
Table 3. Reverse-delete algorithm for G_1 graph

S	사이클	사이클 속성	e_{mst}
18{E,J}	○	**18{E,J}**, 10{J,F}, 8{F,E}	–
10{F,J}	○	**10{F,J}**, 3{J,I}, 9{I,F}	–
9{A,E}	○	**9{A,E}**, 9{E,D}, 3{D,A}	–
9{B,G}	○	**3{B,G}**, 9{G,C}, 2{C,B}	–
9{C,G}	○	**9{C,G}**, 7{G,F}, 8{F,C}	–
9{D,E}	○	**9{D,E}**, 8{E,F}, 9{F,D}	–
9{D,F}	○	**9{D,F}**, 8{F,C}, 2{C,D}	–
9{F, I}	○	**9{F,I}**, 5{I,G}, 7{G,F}	–
8{E,F}	×	–	8{E,F}
8{C,F}	×	–	8{C,F}
7{F,G}	×	–	7{F,G}
6{A,B}	○	**6{A,B}**, 4{B,D}, 3{D,A}	–
5{G, I}	○	**5{G,I}**, 1{I,H}, 4{H,G}	–
4{H, J}	○	**4{H,J}**, 3{J,I}, 1{I,H}	–
4{G,H}	×	–	4{G,H}
4{B,D}	○	**4{B,D}**, 2{D,C}, 2{C,B}	–
3{I, J}	×	–	3{I, J}
3{A,D}	×	–	3{A,D}
2{C,D}	×	–	2{C,D}
2{B,C}	×	–	2{B,C}
1{H, I}	×	–	1{H, I}

표 4. G_1 그래프의 절단과 사이클 속성 간선
Table 4. Cut and cycle property edges for G_1 graph

가중치		1	2	3	4	5	6	7	8	9	10	18
간선		{H,I}	{B,C} {C,D}	{A,D} {I, J}	{B,D} {G,H} {H,J}	{G,I}	{A,B}	{F,G}	{C,F} {E,F}	{A,E} {B,G} {C,G} {D,E} {D,F} {F,I}	{F,J}	{E,J}
MST	절단속성	{H,I}	{B,C} {C,D}	{A,D} {I, J}	{G,H}			{F,G}	{C,F} {E,F}			
	사이클속성				{B,D} {H,J}	{G,I}	{A,B}			{A,E} {B,G} {C,G} {D,E} {D,F} {F,I}	{F,J}	{E,J}
Borůvka 1-간선 연결 그래프	선택	{H,I}	{B,C} {C,D}	{A,D} {I, J}	{B,D}	{G,I}	{A,B}	{F,G}	{C,F} {E,F}	{A,E} {D,E}		
	배제				{G,H} {H,J}					{B,G} {C,G} {D,F} {F,I}	{F,J}	{E,J}

Borůvka, Kruskal과 역-삭제 알고리즘은 다음과 같은 문제점을 갖고 있다.

(1) 그래프의 모든 간선들을 대상으로 절단 속성이나 사이클 속성을 적용한다. 이로 인해 알고리즘 수행 중 $|e|=|v|-1$일 때 최소신장트리를 얻었음에도 불구하고 나머지 간선들에 대해 불필요한 알고리즘을 수행한다.

(2) 표 4에 따르면 MST에 기여하지 못하는 사이클 최대 가중치 간선은 그래프의 최대 가중치 영역의 일정 부분 (가중치가 9 이상)을 차지하고 있다. 따라서 사전에 그래프의 최대 가중치 영역의 일정 부분을 제거하여 단순화된 그래프를 얻는 방법이 필요하다.

Borůvka의 1-간선 연결 그래프는 가중치가 9 이상인 간선들은 선택하지 않아 (2)의 문제점은 해결하였으나

사이클 속성 간선인 5{G,I}와 6{A,H}를 배제시키기 위해 사이클 속성을 적용하여야 하며, MST 간선인 4{G,H}를 선택하지 못해 다시 절단 속성을 적용해야 한다. 즉, Borůvka의 1-간선 연결 그래프 방법은 MST 간선들을 충분히 선택하지 못한다.

MST와 관련하여 기존의 알고리즘들을 개선한 연구로는 최명복과 이상운[9-11]와 이상운과 최명복[12]가 있다.

반면에, 본 논문에서는 원 그래프에서 MST 간선들을 모두 포함하면서, 최대 가중치 영역의 일정 부분을 사전에 배제시키는 단순화된 그래프를 유도하고, 사이클 속성만을 적용하여 MST를 빠르게 얻는 알고리즘을 제안한다.

Ⅲ. 2-간선 연결 그래프 MST 알고리즘

본 장에서는 Borůvka의 1-간선 연결 그래프가 MST 간선들을 충분히 선택하지 못하는 단점을 보완하는 알고리즘을 제안한다. Borůvka는 e_{mwe}는 대부분의 MST 간선들을 포함하고 있기 때문에 정점 당 1-간선 (e_{mwe}) 을 한번에 선택하는 방법을 적용하고 있다. 그러나 1-간선으로는 e_{mst}를 모두 충족시킬 수 없어 2^{nd} 단계에서 PST들을 하나의 MST로 연결하는 e_{mwe}들을 추가로 선택해야 하는 단점이 있다. 본 논문에서는 e_{mst}를 모두 포함하고 있는 충분한 간선들을 한 번에 선택하는 방법으로 정점 당 2-간선 ($e_{\min 1}$과 $e_{\min 2}$)을 선택한다. 이를 2-간선 연결 그래프 (2-Edge Connected Graph)라 하며, MST에 대해 (속성 3)을 제안한다.

(속성 3) "2-간선 연결 그래프는 모든 MST 간선들을 포함한다. 즉, 임의의 정점에 연결된 여러 간선들 중 MST 간선은 $e_{\min 1}$과 $e_{\min 2}$중 어느 하나가 결정한다.

알고리즘은 첫 번째로, (속성 3)을 적용하여 원 그래프에 대해 TwoEdgeSelection을 수행하여 2-간선 연결 그래프로 단순화 시킨다.

두 번째로 2-간선 연결 그래프의 간선을 대상으로 사이클 속성 (속성 2)을 적용하여 MST 간선을 얻는다. 이는 Exclude-CyclePropertyEdge를 수행한다. 단, 이 과정은 (속성 4)를 만족할 때까지 수행한다.

[TwoEdgeSelection]

if $(|e_{min1}|=1)\cap(|e_{min2}|=1)$ then
e_{min1}과 e_{min2} 선택
else if $(|e_{min1}|=1)\cap(|e_{min2}|\geq 2)$ then
e_{min1}과 e_{min2} 모두 선택 (동일 가중치 모두 선택)
else if $(|e_{min1}|\geq 2)$ then
e_{min1} 모두 선택, e_{min2} 선택 않음.

[IncludeCutPropertyEdge] - 오름차순 정렬

/* 사이클 속성 간선 배제
if $(i \in T_i)\cap(j \in T_i)$ then Skip
/* 절단 속성 간선을 MST에 포함
else if $(i \in T_i)\cap(j \in T_{i+1})$ then T_i와 T_{i+1} 병합, $e_{mst}\leftarrow\{i,j\}$
else if $(i \in T_i)\cap(j \notin T)$ then $T_i\leftarrow j$, $e_{mst}\leftarrow\{i,j\}$
else if $(i \notin T)\cap(j \notin T)$ then $T_i\leftarrow\{i,j\}$, $e_{mst}\leftarrow\{i,j\}$.

[ExcludeCyclePropertyEdge] - 내림차순 정렬

if $\{i,j\}\neq e_{cmax}$ then Skip /* 절단 속성 간선
else if $\{i,j\}=e_{cmax}$ then e_{mst}에서 삭제. /* 사이클 속성 간선

(속성 4) 트리 (Tree)의 간선 수는 $|e|=|v|-1$이다.[13] MST도 트리이기 때문에 이 명제가 성립한다. 따라서 $|e_{mst}|=|v|-1$ 일 때 알고리즘을 종료한다.

두 번째 단계는 IncludeCutPropertyEdge의 절단속성을 적용할 수도 있다. 그러나 본 논문에서는 사이클 속성을 우선적으로 적용한다.

Borůvka, Kruskal, 역-삭제 알고리즘과 제안된 알고리즘의 차이점은 그림 2와 같다.

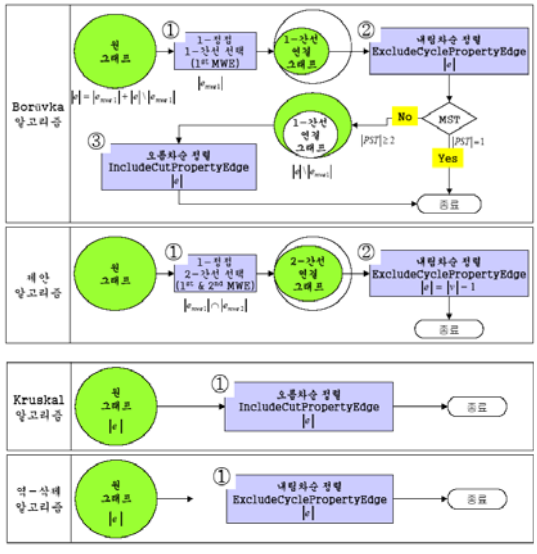


그림 2. 최소신장트리 알고리즘 차이점
Fig. 2. Difference of MST algorithms

제안된 2-ECG MST 알고리즘은 원 그래프의 모든 간선들 $|e|$ 에서 정점 당 2-간선을 선택하여 2-간선 연결 그래프로 단순화 시킨다. 2-간선 연결 그래프에 대해 사이클 속성을 적용하여 $|e_{mst}|=|v|-1$일 때 알고리즘을 종료한다. 제안된 방법은 역-Borůvka 알고리즘으로 생각할 수 있다. 왜냐하면, Borůvka 방법은 1-간선 연결 그래프 (PST)에 절단 속성을 이용하여 $e_{\min}$을 추가적으로 포함하는데 반해, 제안된 알고리즘은 2-간선 연결 그래프에서 사이클 속성을 적용하여 $e_{\max}$를 배제시키는 방법이기 때문이다. 또한 제안된 방법은 Kruskal과 역-삭제 알고리즘이라 할 수 있다. 단지, 그래프의 모든 간선 $|e|$ 대신 단순화된 2-간선 연결 그래프를 모집단으로 하는 차이점이 있다.

2-ECG MST 알고리즘을 그림 1의 G_1 그래프에 적용한 결과는 표 5에 제시되어 있다. Borůvka, Kruskal과 역-삭제 알고리즘은 $|e|=21$ 회를 수행하여 MST를 얻는다. 반면에, 제안 알고리즘은 원 그래프의 $|e|=21$ 을 2-간선 연결 그래프로 15개로 축소시키고, ExcludeCyclePropertyEdge를 10회 수행하여 MST를 얻었다. 즉, "2-간선 연결 그래프는 모든 MST 간선들을 포함하고 있고, 임의의 정점에 연결된 여러 간선들 중 MST 간선은 $e_{\min 1}$과 $e_{\min 2}$중 어느 하나가 결정한다."는 (속성 3)을 만족시킨다. 또한, $|e_{mst}|=|v|-1$일 때 알고리즘을 종료시키는 (속성 4)를 만족시킴으로서 알고리즘 수해 횟수를 크게 단축시키는 효과를 얻었다.

표 5. G_1 그래프의 2-간선 연결 그래프 MST 알고리즘
Table 5. 2-ECG MST algorithm for G_1 graph

(a) 2-간선 연결 그래프

	A	B	C	D	E	F	G	H	I	J
A		6		3	9					
B	6		2	4			9			
C		2		2		8	9			
D	3	4	2		9	9				
E	9			9		8				18
F			8	9	8		7		9	10
G		9	9			7		4	5	
H							4		1	4
I						9	5	1		3
J					18	10		4	3	

	A	B	C	D	E	F	G	H	I	J
A		6		3	9					
B			2	4						
C				2		8				
D					9					
E						8				
F							7			
G								4	5	
H									1	4
I										3
J										

(b) ExcludeCyclePropertyEdge

S	사이클	사이클 속성	e_{mst}	$\|e_{mst}\|$
9{A,E}	○	**9{A,E}**, 9{D,E}, 3{A,D}	-	14
9{D,E}	○	**9{D,E}**, 8{E,F}, 8{C,F}, 2{C,D}	-	13
8{C,F}	×	-	8{C,F}	13
8{E,F}	×	-	8{E,F}	13
7{F,G}	×	-	7{F,G}	13
6{A,B}	○	**6{A,B}**, 4{B,D}, 3{A,D}	-	12
5{G, I}	○	**5{G, I}**, 1{H, I}, 4{G,H}	-	11
4{B,D}	○	**4{B,D}**, 2{B,C}, 2{C,D}	-	10
4{G,H}	×	-	4{G,H}	10
4{H,J}	○	**4{H,J}**, 3{I, J}, 1{H, I}	-	9
3{A,D}		(종료)	3{A,D}	
3{I, J}			3{I, J}	
2{B,C}			2{B,C}	
2{C,D}			2{C,D}	
1{H, I}			1{H, I}	

Ⅳ. 알고리즘 적용성 평가

본 장에서는 그림 3의 9개 그래프에 대해 알고리즘 적용성을 평가해 본다. $G_2, G_3, G_5, G_9, G_{10}$ 그래프는 Peiper[14], G_4는 Wikipedia[15], G_6, G_7, G_8은 Chen[13]에서 인용되었다. G_2와 G_3 그래프는 모든 간선들의 가중치가 상이한 경우이다. 반면에 $[G_4, G_{10}]$ 그래프는 동일한 가중치 간선들이 다수 존재하는 경우이다.

그림 3의 9개 그래프에 대해 제안된 알고리즘을 적용한 결과는 그림 4 ~ 그림 12에 제시하였다. 9개 그래프 모두에서 제안된 알고리즘 모두 최소신장트리를 빠르게 얻을 수 있었다.

본 논문에 적용된 10개 그래프에 대한 알고리즘 수행 횟수를 종합하여 표 6에 제시하였다.

제안된 알고리즘은 2-간선 연결 그래프를 이용함으로서 원 그래프의 간선 수를 28% 줄인 단순화된 그래프를 얻을 수 있었다. 2-간선 연결 그래프에 대해 사이클 속성을 적용한 경우 약 60% 까지 단축시키는 효과를 얻었다. 결국, 제안된 알고리즘이 MST를 가장 빨리 찾을 수 있는 알고리즘으로 검증되었다.

Ⅴ. 결론 및 향후 연구과제

본 논문은 그래프의 간선을 대상으로 최소신장트리를 구하는 Borůvka, Kruskal과 역-삭제 알고리즘을 실제 그래프에 적용하여 문제점을 고찰하였다. 이들 알고리즘들은 그래프의 모든 간선들을 대상으로 절단 속성과 사이클 속성을 적용한다. 이로 인해 알고리즘 수행 과정에서 이미 최소신장트리를 얻었음에도 불구하고 나머지 간선들에 대해 알고리즘을 추가로 불필요하게 수행하는 문제점을 갖고 있다.

본 논문은 먼저, 원 그래프에서 정점 당 최소 가중치 간선 2개 씩을 선택하여 2-간선 연결 그래프를 얻었다. 2-간선 연결 그래프를 대상으로 사이클 속성을 적용하여 빠르게 MST를 얻을 수 있었다. 제안된 방법들을 10개의 실제 그래프들에 적용한 결과 기존의 Borůvka, Kruskal과 역-삭제 알고리즘보다 수행 횟수를 60%를 단축시키는 효과를 얻었다.

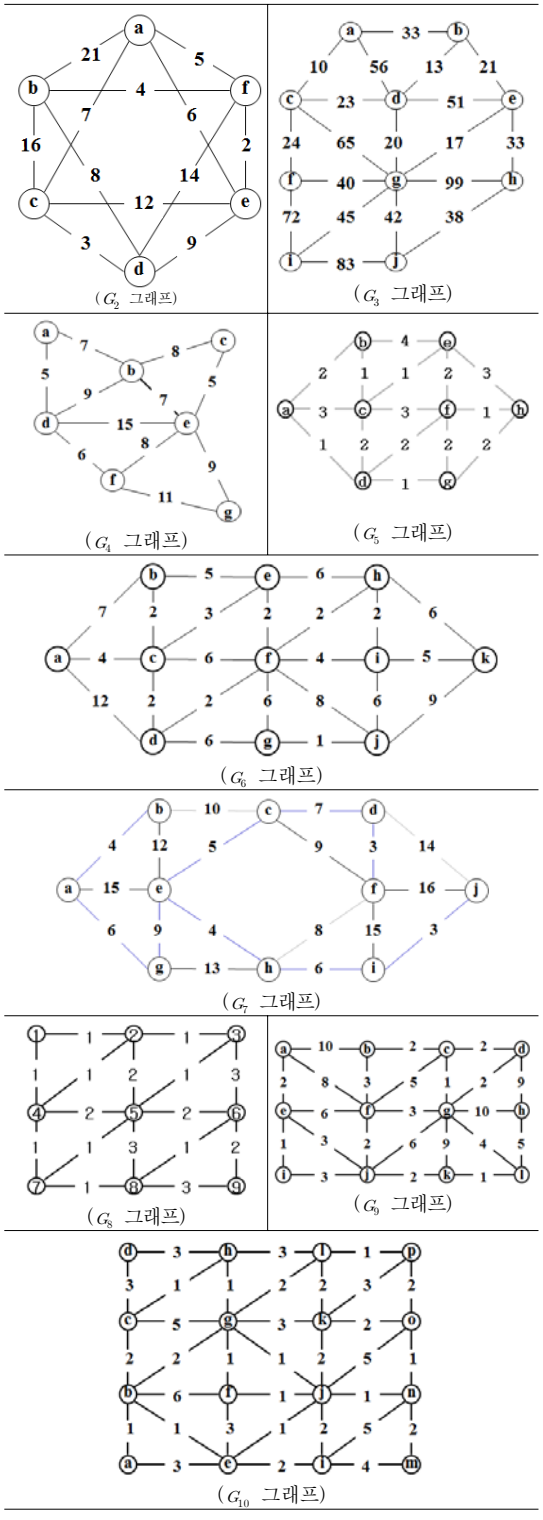
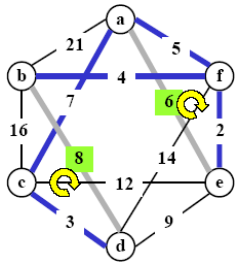


그림 3. 실험에 적용된 그래프
Fig. 3. Experimental graphs

S	사이클	사이클 속성	e_{mst}	$\|e_{mst}\|$
8{b,d}	○	**8{b,d}**, 3{d,c}, 7{c,a}, 5{a,f}, 4{f,b}	-	6
7{a,c}	×	-	7{a,c}	6
6{a,e}	○	**6{a,e}**, 2{e,f}, 5{f,a}	-	5
5{a,f}		(알고리즘 종료)	5{a,f}	
4{b,f}			4{b,f}	
3{c,d}			3{c,d}	
2{e,f}			2{e,f}	

(a) ExcludeCyclePropertyEdge



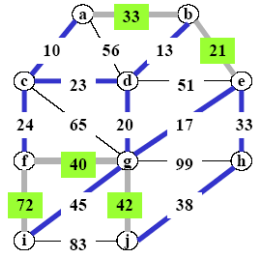
$|v|= 6, |e|= 12 \rightarrow 7$

(b) MST

그림 4. G_2 그래프의 2-간선 연결 그래프 MST
Fig. 4. 2-ECG MST for G_2 graph

S	사이클	사이클 속성	e_{mst}	$\|e_{mst}\|$
72{f,i}	○	**72{f,i}**, 45{g,i}, 40{f,g}	-	13
45{g,i}		-	45{g,i}	13
42{g,j}		**42{g,j}**, 38{h,j}, 33{e,h}, 17{e,g}	-	12
40{f,g}		**40{f,g}**, 20{d,g}, 23{c,d}, 24{c,f}	-	11
38{h,j}		-	38{h,j}	11
35{e,h}		-	35{e,h}	11
33{a,b}		**33{a,b}**, 13{b,d}, 23{c,d}, 10{a,c}	-	10
24{c,f}		-	24{c,f}	10
23{c,d}		-	23{c,d}	10
21{b,e}		**21{b,e}**, 17{e,g}, 20{d,g}, 13{b,d}	-	9
20{d,g}		(알고리즘 종료)	20{d,g}	
17{e,g}			17{e,g}	
13{b,d}			13{b,d}	
10{a,c}			10{a,c}	

(a) ExcludeCyclePropertyEdge



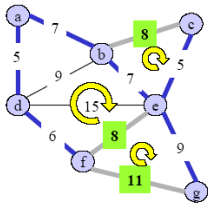
$|v|= 10, |e|= 19 \rightarrow 14$

(b) MST

그림 5. G_3 그래프의 2-간선 연결 그래프 MST
Fig. 5. 2-ECG MST for G_3 graph

S	사이클	사이클 속성	e_{mst}	$\|e_{mst}\|$
11{f,g}	○	**11{f,g}**, 9{g,e}, 8{e,f}	-	8
9{e,g}	×	-	9{e,g}	8
8{b,c}	○	**8{b,c}**, 5{c,e}, 7{e,b}	-	7
8{e,f}	○	**8{e,f}**, 6{f,d}, 5{d,a}, 7{a,b}, 7{b,e}	-	6
7{a,b}		(알고리즘 종료)	7{a,b}	
7{b,e}			7{b,e}	
6{d,f}			6{d,f}	
5{a,d}			5{a,d}	
5{c,e}			5{c,e}	

(a) ExcludeCyclePropertyEdge



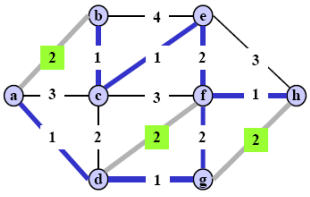
$|v|= 7, |e|= 12 \rightarrow 9$

(b) MST

그림 6. G_4 그래프의 2-간선 연결 그래프 MST
Fig. 6. 2-ECG MST for G_4 graph

S	사이클	사이클 속성	e_{mst}	$\|e_{mst}\|$
2{a,b}	○	**2{a,b}**, 1{b,c}, 1{c,e}, 2{e,f}, 2{d,f}, 1{a,d}	-	9
2{d,f}		**2{d,f}**, 2{f,g}, 1{d,g}	-	8
2{e,f}		-	2{e,f}	8
2{g,h}		**2{g,h}**, 1{f,h}, 2{f,g}	-	7
1{a,d}		(알고리즘 종료)	1{a,d}	
1{b,c}			1{b,c}	
1{c,e}			1{c,e}	
1{d,g}			1{d,g}	
1{f,g}			1{f,g}	
1{f,h}			1{f,h}	

(a) ExcludeCyclePropertyEdge



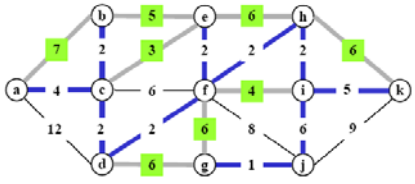
$|v|= 8, |e|= 15 \rightarrow 10$

(b) MST

그림 7. G_5 그래프의 2-간선 연결 그래프 MST
Fig. 7. 2-ECG MST for G_5 graph

S	사이클	사이클 속성	e_{mst}	$\|e_{mst}\|$
7{a,b}	○	**7{a,b}**, 2{b,c}, 4{a,c}	-	17
6{g,f}	○	6{g,f}, 2{d,f}, 6{d,g}	-	16
6{d,g}	○	**6{d,g}**, {g,j}=1, 6{i,j}, 4{f,i}, 2{d,f}	-	15
6{e,h}	○	**6{e,h}**, 2{f,h}, 2{e,f}	-	14
6{h,k}	○	**6{h,k}**, 2{h,i}, 5{i,k}	-	13
6{i, j}	×	-	{i, j}=6	13
5{b,e}	○	**5{b,e}**, 3{c,e}, 2{b,c}	-	12
5{i,k}	×	-	{i,k}=5	12
4{a,c}	×	-	{a,c}=4	12
4{f,i}	○	**4{f,i}**, 2{h,i}, 2{f,h}	-	11
3{c,e}	○	**3{c,e}**, 2{e,f}, 2{d,f}, 2{c,d}	-	10
2{b,c}		(알고리즘 종료)	{b,c}=2	
2{c,d}			{c,d}=2	
2{d,f}			{d,f}=2	
2{e,f}			{e,f}=2	
2{f,h}			{f,h}=2	
2{h,i}			{h,i}=2	
1{g,j}			{g,j}=1	

(a) ExcludeCyclePropertyEdge



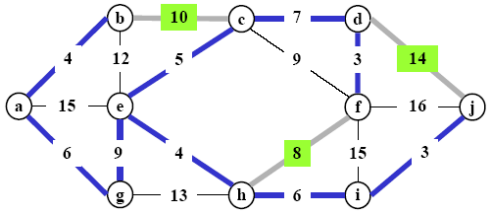
$|v|= 11, |e|= 22 \rightarrow 18$

(b) MST

그림 8. G_6 그래프의 2-간선 연결 그래프 MST
Fig. 8. 2-ECG MST for G_6 graph

S	사이클	사이클 속성	e_{mst}	$\|e_{mst}\|$
14{d,j}	○	**14{d,j}**, 3{i,j}, 6{h,i}, 8{f,h}, 3{d,f}	-	11
10{b,c}	○	**10{b,c}**, 5{c,e}, 9{e,g}, 6{a,g}, 4{a,b}	-	10
9{e,g}	×	-	9{e,g}	10
8{f,h}	○	**8{f,h}**, 4{e,h}, 5{c,e}, 7{c,d}, 3{d,f}	-	9
7{c,d}		(알고리즘 종료)	7{c,d}	
6{a,g}			6{a,g}	
6{h,i}			6{h,i}	
5{c,e}			5{c,e}	
4{a,b}			4{a,b}	
4{e,h}			4{e,h}	
3{d,f}			3{d,f}	
3{i,j}			3{i,j}	

(a) ExcludeCyclePropertyEdge



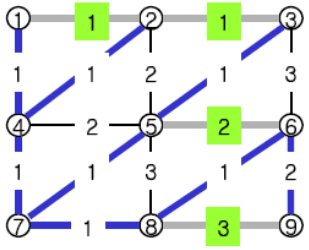
$|v|= 10, |e|= 18 \rightarrow 12$

(b) MST

그림 9. G_7 그래프의 2-간선 연결 그래프 MST
Fig. 9. 2-ECG MST for G_7 graph

S	사이클	사이클 속성	e_{mst}	$\|e_{mst}\|$
3{8,9}	○	**3{8,9}**, 2{6,9}, 1{6,8}	-	11
2{5,6}		**2{5,6}**, 1{6,8}, 1{7,8}, 1{5,7}	-	10
2{6,9}		-	2{6,9}	10
1{1,2}		**1{1,2}**, 1{2,4}, 1{1,4}	-	9
1{1,4}		-	1{1,4}	9
1{2,3}		**1{2,3}**, 1{3,5}, 1{5,7}, 1{4,7}, 1{2,4}	-	8
1{2,4}		(알고리즘 종료)	1{2,4}	
1{3,5}			1{3,5}	
1{4,7}			1{4,7}	
1{5,7}			1{5,7}	
1{6,8}			1{6,8}	
1{7,8}			1{7,8}	

(a) ExcludeCyclePropertyEdge



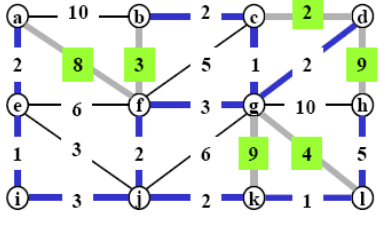
$|v|= 9, |e|= 16 \rightarrow 12$

(b) MST

그림 10. G_8 그래프의 2-간선 연결 그래프 MST
Fig. 10. 2-ECG MST for G_8 graph

S	사이클	사이클 속성	e_{mst}	$\|e_{mst}\|$
9{d,h}	○	**9{d,h}**, 5{h,l}, 4{g,l}, 2{d,g}	-	16
9{g,k}	×	**9{g,k}**, 1{k,l}, 4{g,l}	-	15
8{a,f}	○	**8{a,f}**, 2{f,j}, 3{i,j}, 1{e,i}, 2{a,e}	-	14
5{h,l}	○	-	5{h,l}	14
4{g,l}		**4{g,l}**, 1{k,l}, 2{j,k}, 2{f,j}, 3{f,g}	-	13
3{b,f}		**3{b,f}**, 3{f,g}, 1{c,g}, 2{b,c}	-	12
3{f,g}		-	3{f,g}	12
3{i,j}		-	3{i,j}	12
2{a,e}		-	2{a,e}	12
2{b,c}		-	2{b,c}	12
2{c,d}		**2{c,d}**, 2{d,g}, 1{c,g}	-	11
2{d,g}		(알고리즘 종료)	2{d,g}	
2{f,j}			2{f,j}	
2{j,k}			2{j,k}	
1{k,l}			1{k,l}	
1{c,g}			1{c,g}	
1{e,i}			1{e,i}	

(a) ExcludeCyclePropertyEdge



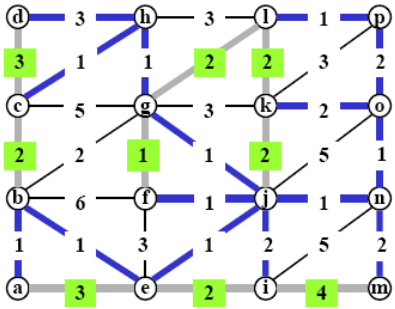
$|v|= 12, |e|= 23 \rightarrow 17$

(b) MST

그림 11. G_9 그래프의 2-간선 연결 그래프 MST
Fig. 11. 2-ECG MST for G_9 graph

S	사이클	사이클 속성	e_{mst}	$\|e_{mst}\|$
4{i,m}	○	**4{i,m}**, 2{m,n}, 1{j,n}, 2{i,j}	-	23
3{a,e}		**3{a,e}**, 1{b,e}, 1{a,b}	-	22
3{c,d}		**3{c,d}**, 3{d,h}, 1{c,h}	-	21
3{d,h}		-	3{d,h}	21
2{b,c}		**2{b,c}**, 1{c,h}, 1{h,g}, 1{g,f}, 1{f,j}, 1{e,j}, 1{b,e}	-	20
2{e,i}		**2{e,i}**, 2{i,j}, 1{e,j}	-	19
2{g,l}		**2{g,l}**, 2{l,k}, 2{k,j}, 1{g,j}	-	18
2{i,j}		-	2{i,j}	18
2{j,k}		**2{j,k}**, 2{k,o}, 1{n,o}, 1{j,n}	-	17
2{k,l}		**2{k,l}**, 1{l,p}, 2{o,p}, 2{k,o}	-	16
2{k,o}		-	2{k,o}	16
2{m,n}		-	2{m,n}	16
2{o,p}		-	2{o,p}	16
1{a,b}		-	1{a,b}	16
1{b,e}		-	1{b,e}	16
1{c,h}		-	1{c,h}	16
1{e,j}		-	1{e,j}	16
1{f,g}		**1{f,g}**, 1{g,j}, 1{f,j}	-	15
1{f,j}		(알고리즘 종료)	1{f,j}	
1{g,h}			1{g,h}	
1{g,j}			1{g,j}	
1{j,n}			1{j,n}	
1{l,p}			1{l,p}	
1{n,o}			1{n,o}	

(a) ExcludeCyclePropertyEdge



$|v|= 16, |e|= 33 \rightarrow 24$

(b) MST

그림 12. G_{10} 그래프의 2-간선 연결 그래프 MST
Fig. 12. 2-ECG MST for G_{10} graph

표 6. MST 알고리즘의 수행 횟수 비교
Table 6. Compare of Execution Number for algorithms

그래프	$\|v\|$	Borůvka, Kruskal과 역-삭제 알고리즘		2-간선 연결 그래프 방법			
				$\|e\|$	수행 횟수	단축율 (%)	
		$\|e\|$	수행 횟수			$\|e\|$	수행 횟수
G_1	10	**21**	**21**	**15**	**10**	28.6	52.4
G_2	6	**12**	**12**	**7**	**3**	41.7	75.0
G_3	10	**19**	**19**	**14**	**10**	26.3	47.4
G_4	7	**11**	**11**	**9**	**4**	18.2	63.6
G_5	8	**15**	**15**	**10**	**4**	33.3	73.3
G_6	11	**22**	**22**	**18**	**11**	18.2	50.0
G_7	10	**18**	**18**	**12**	**4**	33.3	77.8
G_8	9	**16**	**16**	**12**	**6**	25.0	62.5
G_9	12	**23**	**23**	**17**	**11**	26.1	52.2
G_{10}	16	**33**	**33**	**24**	**18**	27.3	45.5
평균						**27.8**	**60.0**

본 논문에서 제안한 "임의의 정점에 연결된 여러 간선들 중 MST 간선은 e_{min1}과 e_{min2}중 어느 하나가 결정한다."는 속성은 10개 그래프에 적용한 결과 모두 만족시켰으나 보다 다양하고 많은 그래프를 대상으로 검증하여 일반화된 명제로 유도할 필요가 있다. 따라서 추후 이 분야를 연구할 계획이다.

References

[1] Wikipedia, "Minimum Spanning Tree," http://en.wikipedia.org/wiki/Minimum_spanning_tree, Wikimedia Foundation, Inc., 2014.

[2] J. Kleinberg and E. Tardos, "Algorithm Design," Addison-Wesley, 2004.

[3] O. Borůvka, "O Jistem Problemu Minimalnim," Prace Mor. Prrodved. Spol. V Brne (Acta Societ. Natur. Moravicae), Vol. III, No. 3, pp. 37-58, 1926.

[4] J. Nešetřil, E. Milková, and H. Nešetřilová, "Otakar Borůvka on Minimum Spanning Tree Problem (Translation of the both 1926 Papers, Comments, History)," DMATH: Discrete Mathematics, Vol. 233, No. 1-3, pp. 3-36, Apr. 2001.

[5] R. C. Prim, "Shortest Connection Networks and Some Generalisations," Bell System Technical Journal, Vol. 36, No. 6, pp. 1389-1401, Nov. 1957.

[6] J. B. Kruskal, "On the Shortest Spanning Subtree and The Traveling Salesman Problem," Proceedings of the American Mathematical Society, Vol. 7, No. 1, pp. 48-50, Feb. 1956.

[7] Wikipedia, "Reverse-Delete Algorithm," http://en.wikipedia.org/wiki/Reverse_delete_algorithm, Wikimedia Foundation, Inc., 2014.

[8] Wikipedia, "Graph (mathematics)," http://en.wikipedia.org/wiki/Graph_(mathematics), Wikimedia Foundation, Inc., 2014.

[9] M. B. Choi and S. U. Lee, "Generalized Borůvka's Minimum Spanning Tree Algorithm," Journal of Institute of Internet, Broadcasting and Communication, Vol. 12, No. 6, pp. 165-173, Dec. 2012.

[10] M. B. Choi and S. U. Lee, "An Efficient Implementation of Kruskal's and Reverse- Delete Minimum Spanning Tree Algorithm," Journal of Institute of Internet, Broadcasting and Communication, Vol. 13, No. 2, pp. 103-114, Apr. 2013.

[11] M. B. Choi and S. U. Lee, "Select and Delete Minimum Spanning Tree Algorithm," Journal of Institute of Internet, Broadcasting and Communication, Vol. 13, No. 4, pp. 107-116, Aug. 2013.

[12] S. U. Lee and M. B. Choi, "Fast Determination of Minimum Spanning Tree Based on Down-sizing the Population of Edges," Journal of Institute of Internet, Broadcasting and Communication, Vol. 14, No. 1, pp. 51-59, Feb. 2014.

[13] WWL. Chen, "Discrete Mathematics," Department of Mathematics, Division of ICS, Macquarie University, Australia, 2003.

[14] C. Peiper, CS 400 - Data Structures for Non CS-Majors," http://www.cs.uiuc.edu/class/fa05/s400/_labs/Lab12/suuri/, 2005.

[15] Wikipedia, "Prim's Algorithm," http://en.ikipedia.org/wiki/Prims_algorithm, Wikimedia Foundation, Inc., 2014.

저자 소개

이 상 운(정회원)

- 1987년 : 한국항공대학교 항공전자공학과 (학사)
- 1997년 : 경상대학교 컴퓨터과학과 (석사)
- 2001년 : 경상대학교 컴퓨터과학과 (박사)
- 2003년 : 강원도립대학 컴퓨터응용과 전임강사
- 2004년 ~ 2007.2 : 국립 원주대학 여성교양과 조교수
- 2007년 3월 ~ 현재 : 강릉원주대학교 멀티미디어공학과 부교수
- 관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 그래프 알고리즘
- e-mail : sulee@gwnu.ac.kr