

<http://dx.doi.org/10.7236/JIIBC.2013.13.1.143>

JIIBC 2013-1-21

무방향 그래프의 최대인접병합 방법을 적용한 최소절단 알고리즘

A Minimum Cut Algorithm Using Maximum Adjacency Merging Method of Undirected Graph

최명복*, 이상운**

Myeong-Bok Choi, Sang-Un Lee

요약 주어진 그래프 $G=(V,E)$, $n=|V|$, $m=|E|$ 에 대해 최소절단을 찾는 연구는 공급처 s 와 수요처 t 가 주어지지 않은 경우와 주어진 경우로 구분된다. s 와 t 가 주어지지 않은 무방향 가중 그래프에 대한 Stoer-Wagner 알고리즘은 임의의 정점을 고정시키고 최대 인접 순서로 나열하여 마지막 정점의 절단 값과 마지막 2개 정점을 병합하면서 정점을 축소시키는 방법으로 $\frac{n(n-1)}{2}$ 회를 수행한다. 또한, s 와 t 가 주어진 그래프에 대한 Ford-Fulkerson 알고리즘은 증대경로를 탐색하여 절단 간선을 결정한다. 더 이상의 증대 경로가 없으면 절단 간선들의 조합으로 최소절단을 결정해야 한다. 본 논문은 단일 s 와 t 가 주어진 무방향 가중 그래프에 대해 최대인접 병합과 절단값을 동시에 계산하는 방법으로 $n-1$ 회 수행으로 단축시켰다. 또한, Stoer-Wagner 알고리즘은 최소 절단을 기준으로 $V=S+T$ 로 양분하지 못할 수 있는데 반해 제안된 알고리즘은 정확히 양분시켰다. 제안된 알고리즘은 Ford-Fulkerson의 증대경로를 찾는 수행 횟수보다 많이 수행하지만 수행과정에서 최소절단을 결정하는 장점이 있다.

Abstract Given weighted graph $G=(V,E)$, $n=|V|$, $m=|E|$, the minimum cut problem is classified with source s and sink t or without s and t . Given undirected weighted graph without s and t , Stoer-Wagner algorithm is most popular. This algorithm fixes arbitrary vertex, and arranges maximum adjacency (MA)-ordering. In the last, the sum of weights of the incident edges for last ordered vertex is computed by cut value, and the last 2 vertices are merged. Therefore, this algorithm runs $\frac{n(n-1)}{2}$ times. Given graph with s and t , Ford-Fulkerson algorithm determines the bottleneck edges in the arbitrary augmenting path from s to t . If the augmenting path is no more exist, we determine the minimum cut value by combine the all of the bottleneck edges. This paper suggests minimum cut algorithm for undirected weighted graph with s and t . This algorithm suggests MA-merging and computes cut value simultaneously. This algorithm runs $n-1$ times and successfully divides V into disjoint S and V sets on the basis of minimum cut, but the Stoer-Wagner is fails sometimes. The proposed algorithm runs more than Ford-Fulkerson algorithm, but finds the minimum cut value within $n-1$ processing times.

Key Words : Max-flow/Min-cut, Vertices Partition, Maximum Adjacency (MA)-ordering, MA-merging

*종신회원, 강릉원주대학교, 멀티미디어공학과

**정회원, 강릉원주대학교 멀티미디어공학과

접수일자 : 2012년 9월 30일, 수정완료 : 2012년 12월 20일

게재확정일자 : 2013년 2월 8일

Received: 30 September 2012 / Revised: 20 December 2012 /

Accepted: 8 February 2013

**Corresponding Author: cmb5859@gmail.com

Dept. of Multimedia Engineering, Gangnung-Wonju National University Wonju Campus, Korea

I. 서 론

도로, 철도, 항만, 항공 등의 물류분야, 통신, 수도, 전기, 집적회로 등의 분야에서 무방향 (Undirected) 망 (network) 또는 그래프(graph)의 공급처 (source, s)에서 수요처 (sink, t)로의 최대 유동량 (maximum flow)을 파악하고, 병목 (bottleneck) 지점을 찾아 확장하면 망의 최대 유동량을 증가시킬 수 있다. 망에서 최대로 흐를 수 있는 양인 유동량은 병목현상이 발생하는 지점들의 수용량 합인 최소 절단 값 (minimum cut value)이 결정한다. 이를 Menger^[1]의 “최대 유동/최소 절단 (Max-Flow/Min-Cut) 정리라 한다. 망의 최대 유동량을 구하기 위해서는 결국, 망의 정점 (또는 노드)들 V 를 최소 절단 값을 갖는 면을 기준으로 S 와 T 의 두 집합으로 정확히 분리시켜야 한다.

최소 절단에 관한 연구는 s 와 t 가 주어지지 않은 무방향 가중 그래프 (undirected weighted graph)의 경우, 단일 (single) s 와 t 가 주어진 경우, 단일 s 와 모든 다른 정점들을 t 로 하는 경우 등이 있다. 도로, 항만, 항공 등의 물류분야와 수도, 전기, 전자회로 등에서의 최대 흐름을 결정하는 최소 절단면을 구하기 위해서는 s 와 t 가 주어진 경우가 적용된다. 따라서 s 와 t 가 주어지지 않은 경우의 알고리즘은 적용될 수 없다. 현실적으로는 다수의 공급처와 다수의 수요처가 존재하기 때문에 다중 (multiple) s 와 t 가 주어진 무방향 가중 그래프 알고리즘이 적용되어야 한다. 그러나 다중 s 와 t 가 주어진 경우 알고리즘 구현이 쉽지 않기 때문에, 본 논문에서는 단일 s 와 t 가 주어진 무방향 가중 그래프에 대한 연구로 한정하고, 다중 s 와 t 인 경우는 추후 연구과제로 남겨둔다.

s 와 t 가 주어지지 않은 무방향 가중 그래프에 대한 대표적인 알고리즘으로는 Stoer-Wagner^[2], 단일 s 와 t 가 주어진 경우는 Ford-Fulkerson 알고리즘^[3,4]이 있다. Ford-Fulkerson 알고리즘의 개념은 방향 그래프 (digraph)에 기반하고 있으나 무방향 그래프에도 적용이 가능하다. 그러나 이 알고리즘은 s 에서 t 로의 증대경로 (augmenting path)를 너비우선 탐색 (Breadth First Search)이나 깊이우선탐색 (Depth First Search)으로 계속적으로 찾아야 하는 어려움과 더불어 각 증대경로에서 결정된 병목지점들을 조합하여 최소 절단을 결정해야 한다. Stoer-Wagner 알고리즘은 무방향 그래프에 대한 최소 절단을 찾는 방법이지만 s 와 t 가 주어지지 않아 주어

진 그래프의 정점들 V 를 정확히 2개의 S 와 T 의 집합으로 분리시키는 최소절단을 찾지 못한다. 이 알고리즘은 최대인접순서 (maximum adjacency-ordering, MA-ordering)이라는 개념을 적용하고 있다.

본 논문에서는 단일 s 와 t 가 주어진 무방향 가중 그래프에 대한 최소절단을 찾기 위해 Stoer-Wagner가 채택한 개념들을 변형시킨 방법을 제안하고, Ford-Fulkerson 알고리즘과 비교하여 본다. 2장에서는 Stoer-Wagner와 Ford-Fulkerson 알고리즘을 고찰한다. 3장에서는 Stoer-Wagner 알고리즘에서 채택한 개념들을 변형시켜 단일 s 와 t 가 주어진 무방향 가중 그래프에 대한 알고리즘을 제안한다. 4장에서는 제안된 알고리즘을 Ford-Fulkerson 알고리즘과 비교하여 성능을 평가해 본다.

II. 관련 연구와 연구 배경

1. 최대 유동량/최소 절단 개념

정점 (vertices, $v \in V$)와 간선 (edges, $e \in E$)로 구성된 무방향 가중 그래프 (weighted undirected graph) $G=(V,E)$ 와 노드 (nodes, $n \in N$)와 호 (arcs, $a \in A$)로 구성된 방향 가중 그래프 (weighted digraph) $G=(N,A)$ 에 대해 호 $a=w(u,v)$, 간선 $e=w\{u,v\}$ 로 표기한다. 여기서 $(u,v) \neq (v,u)$ 로 u 에서 v 로만 흐를 수 있음을, $\{u,v\}=\{v,u\}$ 로 방향성 없이 양방향으로 모두 흐를 수 있음을 의미한다.

망은 근본적으로 방향 그래프 $G=(N,A)$ 의 각 호 $a=(u,v)$ 가 수용량 (capacity) $c(u,v)$ 를 갖고 있으며, 주어진 공급지 s 에서 수요지 t 로 흐를 수 있는 망의 최대유동량은 최소 절단이 결정한다는 최대 유동량/최소 절단 이론에 적용되는 개념은 다음과 같다.^[1,5]

- 유동망 (flow network): 망의 각 간선에 유동량 (flow) $f(u,v)$ 를 할당한 그래프.
- 잉여 수용량 (residual capacity): 하나의 간선에 추가로 흐릴 수 있는 여유량, $c_f(u,v) = c(u,v) - f(u,v)$
- 잉여 망 (residual network) $G_f(V,E_f)$: 간선이 $c_f(u,v)$ 를 갖는 그래프.
- 증대 경로 (augmenting path) (s, v_1, v_2, \dots, t) , $c_f(v_i, v_{i+1}) > 0$: s 에서 t 로 추가로 유동량을 보낼 수 있는 경로.

- 절단 (cut) (S, T) : 모든 $s, t \in V$ 에 대해 모든 정점 V 를 $S(s \in S)$ 와 $T(t \in T)$ 의 두 집합으로 분할하는 것.
- 절단 수용량 $c(S, T)$: $c(S, T) = \sum_{u \in S, v \in T, (u,v) \in E} c(u, v)$ 로 S 영역에서 T 영역으로 절단면 간선들 수용량 합.
- 최소 절단 (minimum cut) $\min c(S, T)$: 절단들 중 가장 작은 수용량 합을 가진 절단.

2. Stoer-Wagner 알고리즘

s 와 t 가 주어지지 않은 무방향 가중 그래프에 대한 Stoer-Wagner^[2] 알고리즘을 고찰해 보자. Stoer-Wagner는 2개 정점을 병합시키기 위해 최대인접순서 개념을 적용한 그림 1의 알고리즘을 제안하였으며, 그림 2의 G_1 그래프에 대해 설명하였다.

```

MinimumCut (G, c, a)
WHILE |V| > 1
    MinimumCutPhase (G, c, a)
    IF cut-of-the-phase < current minimum cut THEN
        current minimum cut = cut-of-the-phase
MinimumCutPhase (G, c, a)
A = {a} /* 임의의 정점을 a로 설정하고 집합 A에 저장
WHILE A ≠ V
    A에 인접한 정점 x 사이의 모든 간선들 수용량의 합 계산.
    수용량 합이 가장 큰 정점을 A에 추가. (최대 인접 순서)
    cut-of-the-phase = A에 마지막으로 추가된 정점 y와 A \ y간의
    절단 값.
    A에 최종적으로 추가된 2개 정점을 병합, 새로운 정점으로 교체.
    
```

그림 1 Stoer-Wagner 최소 절단 알고리즘
Fig. 1. Stoer-Wagner Minimum Cut Algorithm

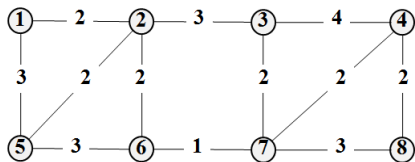


그림 2. G_1 그래프
Fig. 2. Graph G_1

G_1 그래프에 Stoer-Wagner 알고리즘을 적용한 결과를 보다 상세히 설명하면 그림 3과 같다. 여기서 $a = 2$ 로 가정한다. 또한 각 단계 (phase)에서는 실제로 정점들이 병합되지 않지만 최대 인접 순서를 결정하는 과정을 보여주기 위해 단계 1에서 가상으로 병합된 것으로 표기하였으며, 단계 2부터는 생략하였다. $n = 8$ 인 G_1 그래프에 대해 MinimumCutPhase()는 $(n-1), (n-2), \dots, 2, 1$

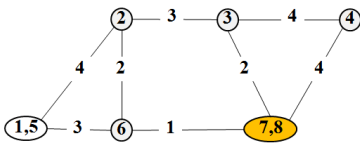
회를 수행한다. 따라서 총 수행 횟수는 $\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = 28$ 회이다. 여기서는 마지막 2개 정점이 1개 정점으로 축소되는 Phase 7은 생략하여 27회만 표현하였다.

순서	최대인접순서	인접순서 결정 방법
1	$A = \{2\}$	
2	$A = \{2, 3\}$	
3	$A = \{2, 3, 4\}$	
4	$A = \{2, 3, 4, 7\}$	
5	$A = \{2, 3, 4, 7, 8\}$	
6	$A = \{2, 3, 4, 7, 8, 6\}$	
7	$A = \{2, 3, 4, 7, 8, 6, 5\}$	

(a) Phase 1

순서	최대인접순서	인접순서 결정 방법
8	$A = \{2\}$	
9	$A = \{2, \{1,5\}\}$	
10	$A = \{2, \{1,5\}, \{6\}\}$	
11	$A = \{2, \{1,5\}, \{6,3\}\}$	
12	$A = \{2, \{1,5\}, \{6,3,4\}\}$	
13	$A = \{2, \{1,5\}, \{6,3,4,7\}\}$	

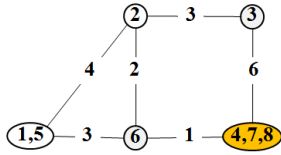
$$c\{8, \{2, \{1,5\}, 6, 3, 4, 7\}\} = 5$$



(b) Phase 2

순서	최대인접순서	인접순서 결정 방법
14	$A = \{2\}$	
15	$A = \{2, \{1,5\}\}$	
16	$A = \{2, \{1,5\}, \{6\}\}$	
17	$A = \{2, \{1,5\}, \{6,3\}\}$	
18	$A = \{2, \{1,5\}, \{6,3,4\}\}$	

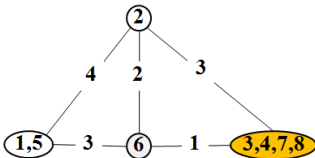
$$c\{\{7,8\}, \{2, \{1,5\}, 6, 3, 4\}\} = 7$$



(c) Phase 3

순서	최대인접순서	인접순서 결정 방법
19	$A = \{2\}$	
20	$A = \{2, \{1,5\}\}$	
21	$A = \{2, \{1,5\}, 6\}$	
22	$A = \{2, \{1,5\}, 6, 3\}$	

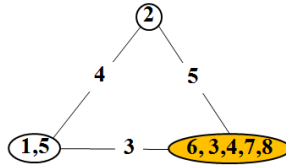
$$c\{\{4,7,8\}, \{2, \{1,5\}, 6, 3\}\} = 7$$



(d) Phase 4

순서	최대인접순서	인접순서 결정 방법
23	$A = \{2\}$	
24	$A = \{2, \{1,5\}\}$	
25	$A = \{2, \{1,5\}, 6\}$	

$$c\{\{3,4,7,8\}, \{2, \{1,5\}, 6\}\} = 4$$

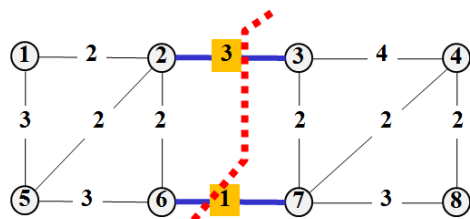


(e) Phase 5

순서	최대인접순서	인접순서 결정 방법
26	$A = \{2\}$	
27	$A = \{2, 3, 4, 6, 7, 8\}$	

$$c\{\{1,5\}, \{2, 3, 4, 6, 7, 8\}\} = 7$$

최소 절단 값 : $\{2, 1, 5, 6\}$ — 4 — $\{3, 4, 7, 8\}$



Min-cut

(f) Phase 6

그림 3. G_1 그래프의 Stoer-Wagner 알고리즘 적용

Fig. 3. Stoer-Wagner Algorithm Application of Graph G_1

만약, $n = 12$ 인 그림 4의 G_2 그래프 도시에서 인접도시와 연결 도로 s 와 t 가 주어지지 않았다면 Stoer-Wagner 알고리즘은 66회를 수행하여 i 정점의 절단 값 4를 최소 절단값으로 결정한다. 즉, 만약, i 가 s 또는 t 가 아니라면 변두리의 하나의 마을로 s 에서 t 로의 주요 교통 중심로의 병목지점이 되지 못한다. 결국, s 와 t 가 주

어진 무방향 그래프에 대해 Stoer-Wagner 알고리즘을 적용할 수 없는 경우가 발생한다. G_2 그래프는 Peiper^[6]에서 인용되었다.

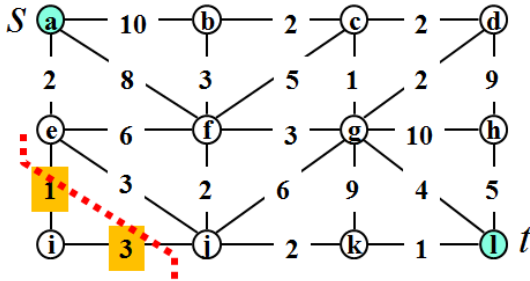


그림 4. G_2 그래프
Fig. 4. Graph G_2

단일 s 와 t 가 주어진 경우 모든 흐름은 s 에서 공급되어 t 로 모여진다는 방향 그래프 개념이 적용된다. 유동-기반의 대표적인 최소 절단 알고리즘인 Ford-Fulkerson 알고리즘^[3,4]은 $s-t$ 의 임의의 증대경로를 너비우선 탐색이나 깊이우선 탐색 방법으로 찾아 유동량 f 을 계산하고 $c-f=0$ 인 병목지점을 삭제하고 다시 $s-t$ 의 가능한 증대경로를 찾아 유동량을 계산한다. 이 방법을 모든 가능한 $s-t$ 증대경로가 존재하지 않을 때까지 수행한다. 이때 증대 경로상의 임의의 간선이 $s \rightarrow t$ 의 순방향이면 $f = f + c_f(p)$, $t \rightarrow s$ 의 역방향이면 $f = f - c_f(p)$ 로 계산한다. Ford-Fulkerson 알고리즘은 그림 5에 제시하였다. Edmonds-Karp 알고리즘^[7]은 Ford-Fulkerson 알고리즘과 동일하나 증대경로를 임의로 찾지 않고 최단경로로 찾는다. 그림 6의 G_3 그래프에 대해 Ford-Fulkerson 알고리즘으로 최소절단을 찾은 결과는 그림 7과 같다.^[7]

MinimumCut (G, c, s, t)

모든 간선 방향 (u, v) 에 $f(u, v) = 0$ 로 초기화.

WHILE 잉여 수용량 $c_f(u, v) > 0$ 을 가진 s 에서 t 로의 경로 p 존재

병목 수용량 $c_f(p) = \min \{c_f(u, v) | (u, v) \in p\}$ 결정.

각 간선 $(u, v) \in p$ 에 대해

if $f(u, v)$ then $f(u, v) = f(u, v) + c_f(p) /* s \rightarrow t$ 방향

else if $f(v, u)$ then $f(v, u) = f(v, u) - c_f(p) /* s \leftarrow t$ 방향

$c-f=0$ 인 간선들을 대상으로 최소절단을 구함.

그림 5. Ford-Fulkerson 최소 절단 알고리즘
Fig. 5. Ford-Fulkerson Minimum Cut Algorithm

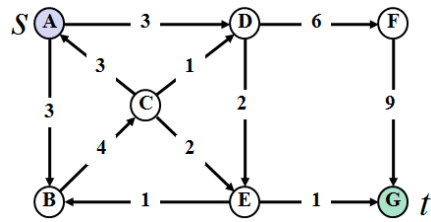


그림 6. G_3 그래프
Fig. 6. Graph G_3

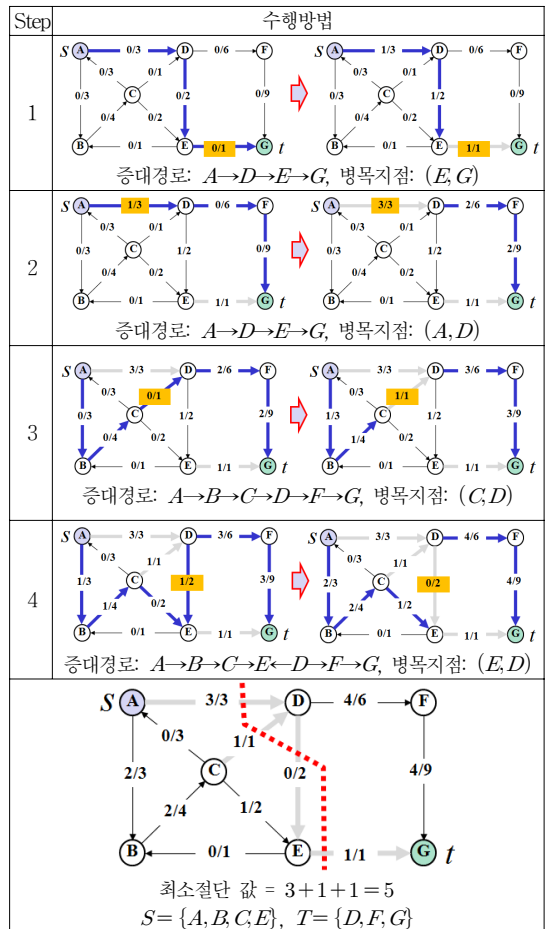


그림 7. G_3 그래프의 Ford-Fulkerson 최소 절단 알고리즘
Fig. 7. Ford-Fulkerson Algorithm Application of Graph G_3

Ford-Fulkerson 알고리즘을 적용할 경우 증대경로를 탐색하는 과정이 쉽지 않다. 왜냐하면 경로상의 모든 간선의 방향이 $s \rightarrow t$ 의 순방향이면 쉽게 찾지만 Step 4에서 처럼 $A \rightarrow B \rightarrow C \rightarrow E \leftarrow D \rightarrow F \rightarrow G$ 경로에서 $s \leftarrow t$ 역방

향인 $E \leftarrow D$ 를 고려한 증대경로를 찾기가 쉽지 않다. 또한, 증대경로에서 병목지점은 찾을 수 있지만 더 이상의 증대경로가 없어 알고리즘이 종료되면 그 동안 찾은 병목지점들의 조합으로 최소절단을 결정해야만 한다.

III. 최대인접병합 최소 절단 알고리즘

단일 s 와 t 가 주어진 무방향 가중 그래프에 대한 최소절단을 찾는 알고리즘은 다음과 같이 가정한다.

- (1) 모든 정점의 간선은 양방향으로 방향성이 없으며 가중치는 주어진다 고 간주한다. 또한, s 와 t 는 사전에 결정된다.
- (2) 그래프의 모든 정점에 대한 부속 간선 가중치 합을 구하여 최소 가중치 합을 가진 2개 정점에 대해서는 s 와 t 로 결정하지 않는다.

제안된 알고리즘의 수행 방법은 다음과 같으며, 그림 8에 제시되어 있다.

- (1) 알고리즘은 S, T, U 와 V 의 4개 집합을 사용하며, 초기치는 $S = \{s\}, T = \{t\}, V = V \setminus \{s, t\}, U = \{\emptyset\}$ 이다. 여기서 S 는 공급지 집합, T 는 목적지 집합, U 는 S 와 T 로의 병합이 미확정된 집합이다. V 는 그래프의 모든 정점들 중에서 병합되지 않고 남아 있는 정점들의 집합이다.
- (2) 망의 최대 수용량 간선 $w(u, v)$ 를 선택하여 이 간선의 2개 정점 중 어느 하나가 S 와 T 중 어느 집합의 원소인지 결정하여 나머지 정점을 해당 집합에 병합시키고 인접정점들과의 간선 값을 변경한다. 또한, 병합된 집합의 부속 간선 가중치 합을 절단값으로 계산한다. 만약, 최대 수용량이 동일한 다수의 간선이 존재하면 S, T, U 순서로 선택한다.
- (3) $V = \{\emptyset\}$ 이면 망의 모든 정점들이 정확히 S 와 T 의 2개 집합으로 분리된 상태임으로 알고리즘을 종료한다.

“최대 수용량 인접 정점 병합” 알고리즘과 Stoer-Wagner와 Ford-Fulkerson 알고리즘의 차이점은 다음과 같다.

- (1) Stoer-Wagner 알고리즘은 임의의 정점을 a 로 고정시키고 최대인접순서로 정점들을 나열시켜 마

지막 2개 정점을 병합시킨다. 반면에 제안된 알고리즘은 $S = \{s\}, T = \{t\}, U = \{\emptyset\}$ 로 고정시키고 그래프의 최대 가중치 간선 (최대 인접)의 상대 정점을 인접한 집합인 S 또는 T 로 병합한다. 따라서 최대 인접의 주체는 A 집합 단독이 아닌 S 와 T 중 하나가 된다. 제안된 알고리즘은 MA-Merging과 동시에 절단값을 계산한다.

- (2) Ford-Fulkerson 알고리즘은 s 에서 t 로의 증대경로를 찾아 병목 지점을 결정한다. 더 이상의 증대경로가 없으면 병목 지점들을 대상으로 최소 절단을 결정한다. 반면에, 제안된 알고리즘은 그래프의 최대 가중치 간선 $w(u, v)$ 을 S 또는 T 로 병합시키고 절단값을 계산한다.

MinCutStep (G, c, s, t)

$S = \{s\}, T = \{t\}, V = V \setminus \{s, t\}, U = \{\emptyset\}$. /* 초기치 결정

ΣS 와 ΣT 계산. /* ΣS : S 의 총 유출량 (부속 간선 가중치 합), ΣT : T 의 총 유입량

$C = \min\{\Sigma S, \Sigma T\}$. /* 초기 최소 절단 값

While $V = \{\emptyset\}$ do

최대 수용량 간선 $w(u, v)$ 선택 (단, 동일 수용량 간선이 다수 존재시 S, T, U 순서로 선택).

if $u \in S, v \in T$ then skip

else if $u \in V, v \in V$ then $U = \{u, v\}, V = V \setminus \{u, v\}$

else if $u \in S, v \in V$ then $S = S + v, V = V \setminus v$

else if $u \in S, v \in U$ then $S = S + U, U = \{\emptyset\}$

else if $u \in T, v \in V$ then $T = T + v, V = V \setminus v$

else if $u \in T, v \in U$ then $T = T + U, U = \{\emptyset\}$

else if $u \in U, v \in V$ then $U = U + v, V = V \setminus v$.

병합된 집합과 인접한 정점들 간의 간선들 수용량 재 계산.

if 병합된 집합 = S 또는 T then 절단 값 계산.

$C = \min\{C, C_i\}$.

최소 절단 값 : $\min c(S, T) = C$.

그림 8. 최대인접병합 최소 절단 탐색 알고리즘
Fig. 8. A Minimum Cut Algorithm using Maximum Adjacency Merging Method

IV. 알고리즘 적용 및 분석

그림 2의 G_1 그래프에서 $s = 5, t = 4$ 인 경우 제안된 알고리즘을 적용한 결과는 그림 9와 같다. 제안된 알고리즘은 Stoer-Wagner 알고리즘의 $\frac{n(n-1)}{2}$ 회를 $n-1$

회로 단축시킴과 동시에 최소절단을 기준으로 $V = S + T$ 로 양분시킬 수 있다.

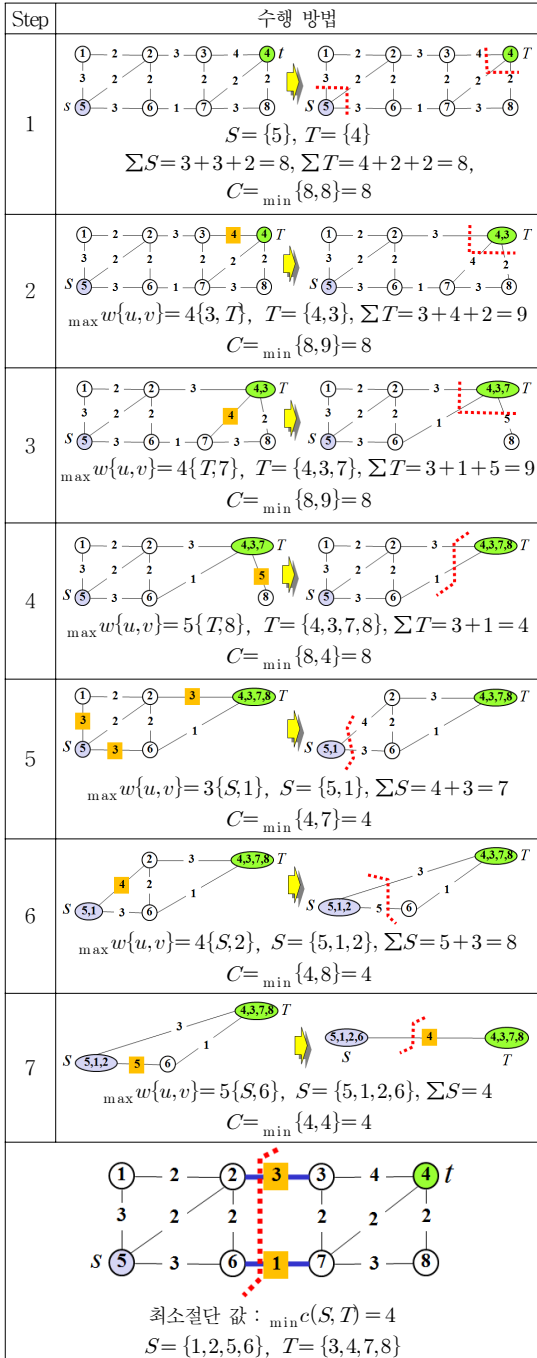
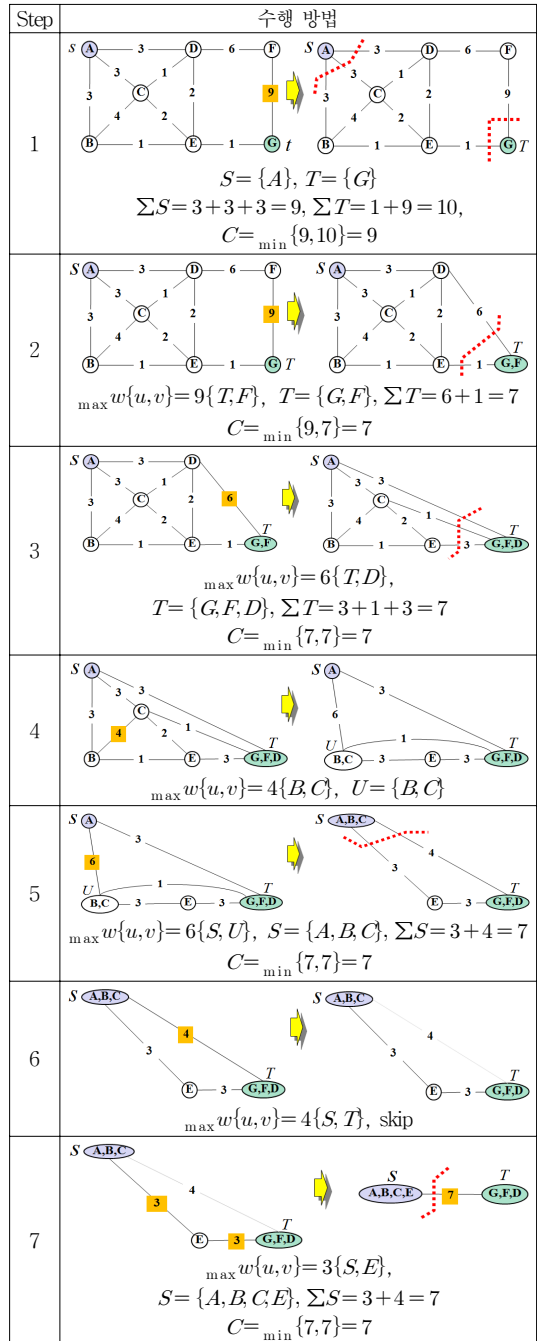
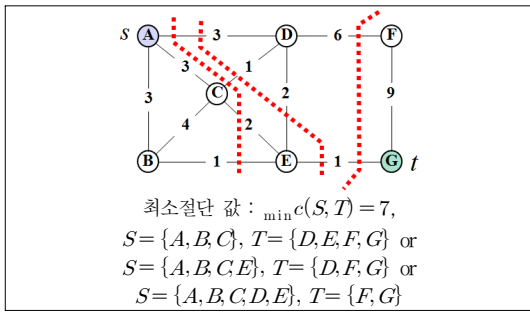


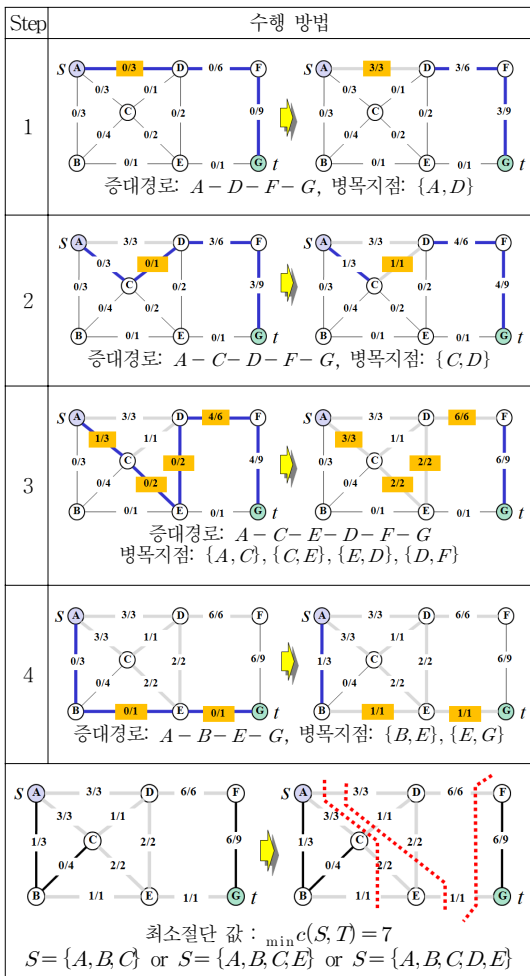
그림 9. G_1 그래프의 최대인접병합 최소 절단
 Fig. 9. A Minimum Cut Algorithm using Maximum Adjacency Merging Method with Graph G_1

그림 6의 G_3 그래프가 무방향 그래프라 가정하고 제안된 알고리즘과 Ford-Fulkerson 알고리즘을 적용한 결과는 그림 10에 제시되어 있다. Ford-Fulkerson 알고리즘에서 구한 병목 간선들이 제안된 알고리즘에서 구한 최소 절단의 모든 간선들과 동일함을 알 수 있다.





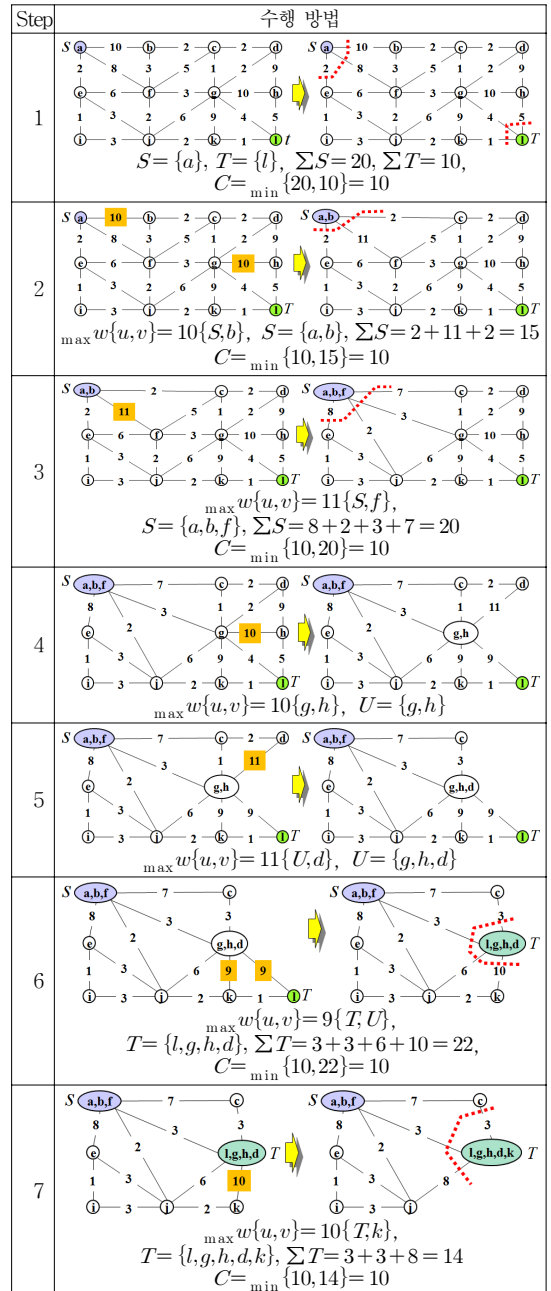
(a) 최대인접병합 알고리즘

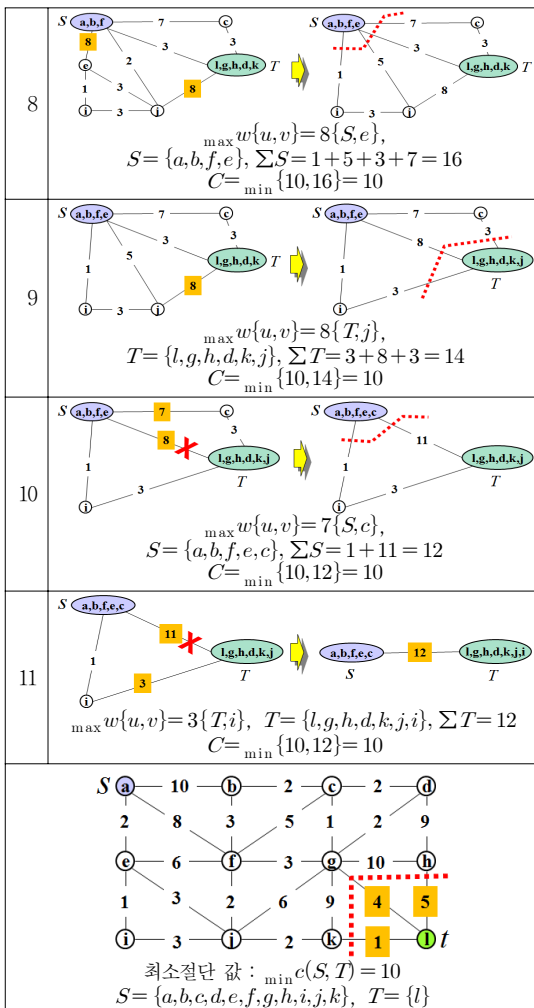


(b) Ford-Fulkerson 알고리즘

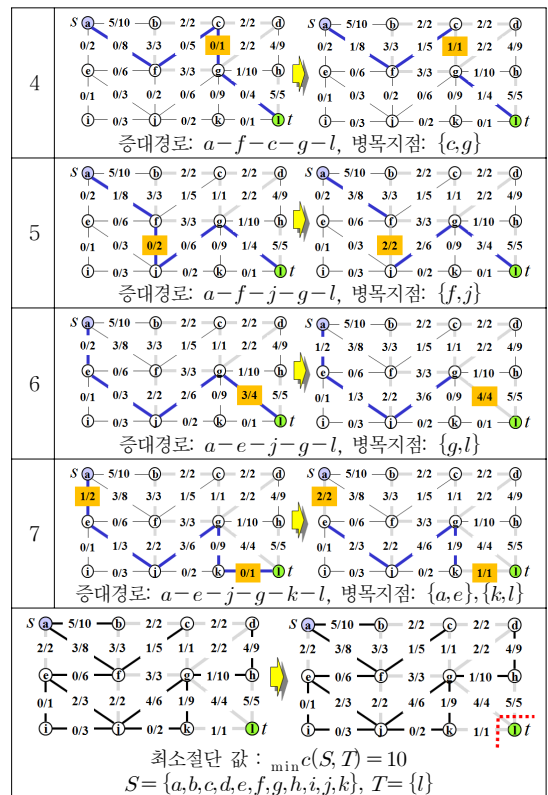
그림 10. G_3 무방향 그래프의 최소 절단
 Fig. 10. A Minimum Cut Algorithm using Maximum Adjacency Merging Method with Graph G_3

그림 4의 G_2 그래프에 대해 제안된 알고리즘과 Ford-Fulkerson 알고리즘을 적용한 결과는 그림 11에 제시하였다. Ford-Fulkerson 알고리즘은 증대경로를 찾는 수행횟수는 제안된 알고리즘보다 적은 반면에 병목지점들의 조합으로 최소절단을 구해야 한다. 그러나 제안된 알고리즘은 $n-1$ 회 수행 과정에서 최소 절단을 결정한다.





(a) 최대 인접병합 알고리즘



(b) Ford-Fulkerson 알고리즘

그림 11. G_2 그래프의 최소 절단
 Fig. 11. A Minimum Cut of Graph G_2

V. 결론

본 논문에서는 기존의 관련 연구들^[1-8]에 비해 단일 s 와 t 가 주어진 무방향 가중 그래프에 대한 최소절단을 매우 간단히 구하는 알고리즘을 제안하였다. 무방향 가중 그래프에 대해 Stoer-Wagner 알고리즘은 최대인접순서 (MA-ordering) 개념을 적용한데 반해 제안된 알고리즘은 최대인접 병합 (MA-merging) 개념을 적용하였다. Stoer-Wagner 알고리즘을 $\frac{n(n-1)}{2}$ 회의 MA-ordering을 수행하며, s 와 t 가 주어진 경우 얻은 최소절단은 $V = S + T$ 로 양분시키는 절단면이 되지 않을 수도 있다. 반면에 제안된 알고리즘은 최대인접병합을 $n-1$ 회로 단축시킴과 동시에 최소절단면을 기준으로 $V = S + T$ 로 양분시킨다.

Ford-Fulkerson 알고리즘은 임의의 증대경로를 계속 탐색하면서 하나의 증대경로 상의 병목지점을 결정하고 나머지 지점간의 잉여 수용량을 계산해야한다. 더 이상의 증대경로가 없으면, 결정된 병목지점들의 조합으로 최소절단을 다시 계산해야 한다. 반면에, 제안된 알고리즘은 항상 $n-1$ 회를 수행하지만 이 과정에서 $V=S+T$ 로 양분시키는 최소절단 값이 계산되는 특징이 있다.

제안된 알고리즘은 s 와 t 가 주어진 무방향 가중 그래프에 대한 최소절단을 구하는 알고리즘에 한정시켰다. 그러나 최소절단 문제에 대한 원조인 구 소련의 다중 공급처에서 다수의 동유럽 소련연방 국가로의 철도 물자 수송에 대한 최소절단 문제^[9,10]를 해결하기 위해서는 다중 s 와 t 와 무방향 그래프 (양방향)와 방향 그래프 (단방향)이 혼합된 문제로 발전시켜야 한다. 따라서 이 분야에 대한 알고리즘은 추후 연구할 계획이다.

참 고 문 헌

- [1] Wikipedia, "Max-flow Min-cut Theorem," http://en.wikipedia.org/wiki/Max_flow_Min_cut_Theorem, Wikimedia Foundation Inc., 2010.
- [2] M. Stoer and F. Wagner, "A Simple Min-Cut Algorithm," Journal of the ACM, Vol. 44, Issue. 4, pp. 585-591, 1997.
- [3] Wikipedia, "Ford-Fulkerson Algorithm," http://en.wikipedia.org/wiki/Ford_Fulkerson_Algorithm, Wikimedia Foundation Inc., 2010.
- [4] L. R. Ford and D. R. Fulkerson, "Maximal Flow Through a Network," Canadian Journal of Mathematics, Vol. 8, pp. 399-404, 1956.
- [5] Wikipedia, "Maximum Flow Problem" http://en.wikipedia.org/wiki/Maximum_flow_Problem, Wikimedia Foundation Inc., 2010.
- [6] C. Peiper, CS 400 - Data Structures for Non CS-Majors," http://www.cs.uiuc.edu/class/fa05/cs400_Labs/Lab12/suuri/, 2005.
- [7] J. Edmonds and R. M. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," Journal of the ACM, Vol. 19, Issue. 2, pp. 248-264, 1972.
- [8] Yong-Jin Lee, Dong-Woo Lee, "Minimum Cost Spanning Tree Problem in Wireless Sensor Network and Heuristic Algorithm," Journal of Advanced Information Technology and Convergence, pp. 275~282, vol. 7, no. 4, 2009. 8.
- [9] A. W. Boldyreff, "Determination of the Maximal Steady State Flow of Traffic through a Railroad Network, Research Memorandum RM-1532, The RAND Corporation, Santa Monica, California, Journal of the Operations Research Society of America, Vol. 3, pp. 443-465, 1955.
- [10] A. Schrijver, "On the History of the Transportation and Maximum Flow Problems," <http://homepages.cwi.nl/~lex/files/histtrpclean.pdf>, Department of Mathematics, University of Amsterdam, Netherlands, 2004.

저자 소개

최 명 복(중신회원)



- 2001년 : 아주대학교 컴퓨터공학과(박사)
- 1997 ~ 현재 : 강릉원주대학교 멀티미디어공학과 교수
- 2004년 1월 ~ 현재 : 한국인터넷방송통신학회 이사

<주관심분야 : 지능형 정보검색 등>

이 상 운(정회원)



- 1998년 ~ 2001년 : 경상대학교 컴퓨터과학과 (박사)
- 2003년 : 강원도립대학 컴퓨터응용과 전임강사
- 2004년 ~ 2007년 2월 : 국립 원주대학 여성교양과 조교수
- 2007년 3월 ~ 현재 : 강릉원주대학교 과학기술대학 멀티미디어공학과 부교수

<주관심분야 : 그래프 알고리즘 등>