韓國經營科學會誌
第29卷 第1號
2004年 3月


# 네트워크 단절문제에 대한 상한과 하한을 구하는 해법*

김현준** · 명영수*** · 박성수**** · 오상민*****

# Lower and Upper Bounding Strategies for the Network Disconnection Problem*

Hyun-Joon Kim** · Young-Soo Myung*** · Sungsoo Park**** · Sang-Min Oh*****

■ Abstract ■

The network disconnection problem is to find a set of edges such that the total cost of removing the edges is no more than a given budget and the weight of nodes disconnected from a designated source by removing edges is maximized. Martel et al. have shown that the problem with unit capacity and unit demand is NP-hard and Myung and Kim present an integer programming formulation and develop an algorithm that includes a preprocessing procedure and lower and upper bounding strategies. In this paper, we present new findings on the properties of the optimal solution and an alternative integer programming formulation, based on which new lower and upper bounding strategies are developed. Computational results for evaluating the performance of the proposed algorithm are also presented.

Keyword : Network Disconnection Problem, Combinatorial Optimization

## 1. 서 론

네트워크 단절문제는 다음과 같이 정의된다. 무방향 네트워크(undirected network), 사전에 정해진 원천노드(source node), 각 에지(edge)마다 에지를 제거하는데 드는 비용, 에지를 제거하는데 사용할 수 있는 예산 및, 각 노드별 가중치가 주어져 있다고 가정하자. 네트워크 단절문제의 목적은 원

논문접수일 : 2004년 2월 9일 논문게재확정일 : 2004년 3월 2일
* 이 논문은 2002년도 한국학술진흥재단의 연구비에 의하여 연구되었음(KRF-2002-041-B00144).
** 울산대학교 경영학부
*** 단국대학교(천안) 경영학과
**** 한국과학기술원 산업공학과에서
***** 한국과학기술원 산업공학과에서 박사과정

천노드와 연결이 끊어지는 노드들의 가중치의 합이 최대가 되도록 에지를 제거하는 것이다. 이 때, 에지를 제거하는데 드는 비용의 합은 주어진 예산의 범위를 넘어서는 안 된다.

네트워크 단절문제는 네트워크의 노드들과 원천노드와의 연결을 단절시키려는 네트워크에 대한 공격을 모델로 하여, Martel et al.[9]에 의해서 처음 제시되었다. 저자들은 에지를 제거하는데 드는 비용과 노드별 가중치가 모두 1의 값을 갖더라도, 네트워크 단절문제는 NP-hard 에 속한 어려운 문제임을 밝히고, 선택 가능한 에지의 집합을 중복없이 일일이 탐색하여 최적해를 구하는 방법을 제시하였다. 물론 문제의 복잡성에서 알 수 있듯이 이 방법은 다항시간(polynomial time) 안에 완료되지는 않는다. Myung and Kim[11]은 Martel et al.[9]이 제시한 방법보다는 좀 더 현실적인 방법으로 네트워크 단절문제를 풀 수 있는 해법을 개발하였다. 이 들은 주어진 그래프의 특성을 이용하여 문제의 크기를 줄이는 과정과 수리계획모형을 제시하였고, 제시된 모형을 이용하여 하한 및 상한을 도출하는 절차를 개발하였다.

네트워크 단절문제를 다룬 연구는 위에서 언급한 두 연구뿐이나, Cunningham[2]과 Gusfield[7]에 의하여 유사한 문제인 네트워크 공격문제가 다루어진 적이 있다. 네트워크 공격문제는 에지의 제거비용과 에지의 제거로 인하여 서로 단절되는 부분네트워크의 개수와의 비율을 최소화하는 문제이다. Martel et al.[9]은 네트워크 단절문제가 전자상거래를 위한 전산망 및 분산처리를 위한 전산망 등에서 응용될 수 있음을 밝혔다. 네트워크 단절문제와 공격문제는 네트워크 공격자의 입장에서 정의되어 있으나, 이를 이용하면 생존도(survivability)가 높은 네트워크를 구성하는 방법으로도 활용할 수 있다. 통신 서비스에의 의존도가 늘어가고 있는 오늘날의 사회에서는 비용 면에서 효율적이면서도 일부 구성요소의 장애에도 지속적인 서비스의 제공이 가능한 안정적인 통신망의 구축이 필수적이어서, 네트워크의 생존도에 대한 연구는 중요한 과제로 인식되어 왔다[1, 6, 10, 13, 15].

본 연구의 목적은 네트워크 단절문제의 특성을 추가로 규명하고, 이를 바탕으로 네트워크 단절문제의 새로운 상한과 하한을 구하는 방법을 제시하는 것이다. 우선 상한을 구하기 위해서 Myung and Kim[11]이 제시한 수리계획모형과 다른 새로운 수리계획모형의 선형계획완화를 이용하고, 하한을 구하기 위해서 새롭게 규명된 최적해의 특성을 이용한 다양한 휴리스틱과 타부 탐색 알고리즘을 이용한 휴리스틱을 개발하기로 한다. 아울러 충분한 계산실험을 통하여 개발된 상한과 하한을 구하는 해법의 성능을 평가하기로 한다. 네트워크의 단절문제는 네트워크가 무방향(undirected)인 경우와 유방향(directed)인 두 경우로 나누어 생각할 수 있다. 두 경우 모두 NP-hard에 속하는 문제이다. 본 논문에서는 무방향의 네트워크를 가정하는데 실제로 본 논문에서 제시된 내용은 유방향의 네트워크에도 약간의 변형을 통해서 그대로 적용이 가능하다.

## 2. 용어의 정의 및 문제의 특성

주어진 네트워크에서 노드의 집합을 $V=\{1, \cdots, n\}$, 무방향 에지의 집합을 $E=\{1, \cdots, m\}$로 표시한다. 사전에 정해진 원천노드는 노드 1로 가정한다. 두 노드 $i \in V$와 $j \in V$에 걸쳐있는 에지를 $e=\{i, j\}$로 표시하기로 하고 두 노드 $i$와 $j$를 에지 $e$의 종단노드(end nodes)라고 부르기로 한다. 원천노드를 제외한 노드들의 집합을 표시하기 위하여 $V_1 = V \setminus \{1\}$을 사용한다. 주어진 그래프 $G$에는 두 노드를 동일한 종단노드로 하는 복수의 링크(multiple link)는 존재할 수 있으나 동일한 노드를 종단노드로 하는 링크(self-loop)는 존재하지 않는 것으로 가정한다. 각 에지 $e \in E$마다 에지를 제거하는데 드는 비용을 $c_e$, 에지를 제거하는데 사용할 수 있는 예산을 $b$, 각 노드 $i \in V_1$의 가중치를 $w_i$로 표시하기로 한다. $c_e, b, w_i$는 모두 양의 값

을 갖는 것으로 가정하고, 에지의 비용과 노드의 가중치에 대한 합을 표현하기 위하여 에지의 부분집합 $E' \subseteq E$에 대해서 $c(E') = \sum_{e \in E'} c_e$, 노드의 부분집합 $S \subseteq V_1$에 대해서 $w(S) = \sum_{i \in S} w_i$의 기호를 사용하기로 한다.

$V_1$의 임의의 부분집합 $S$에 대해서, $\delta(S)$는 양쪽 종단노드 중 하나의 노드는 $S$에 다른 노드는 $V \setminus S$에 속하는 에지의 집합, 즉 노드집합을 둘로 분리시키는 컷(cut)을 나타내는 것으로 정의한다. 모든 $S' \subseteq S$에 대해서 $\delta(S)$는 노드 1과 $S$의 부분집합 $S'$을 분리시키는 컷이므로 $1-S'$컷이라고 부르기로 한다. 임의의 컷 $\delta(S)$에 대해서 컷에 속한 에지를 제거하는데 드는 비용, 즉 $c(\delta(S))$를 컷의 비용으로 정의한다. 그리고 $V_1$의 임의의 부분집합 $S$에 대한 $1-S$컷 중에서 컷의 비용이 최소인 컷을 최소비용 $1-S$컷, 최소비용 $1-S$컷 중에서 컷에 의해 분리되는 그래프의 부분 중 $S$를 포함하는 부분에 속한 노드의 가중치의 합이 최대인 $1-S$컷을 최대가중치 최소비용 $1-S$컷이라고 정의한다.

$V_1$의 임의의 부분집합 $S$에 대해서 최소비용 $1-S$컷의 비용이 $b$이하이면 예산의 범위 내에서 컷에 포함된 에지를 제거하여 $S$를 원천노드에서 분리시킬 수 있게 된다. 이 때, $S$를 분리가능-노드집합이라고 부르기로 한다. 이러한 정의를 이용하면, 네트워크 단절문제는, 그래프 $G=(V,E)$와 $c_e, b, w_i$ 등이 주어져 있을 때, 가중치의 합, $w(S)$가 최대가 되는 분리가능-노드집합 $S$를 구하는 문제로 표현할 수 있다.

Myung and Kim[11]은 컷의 서브-모듈라 특성(Submodularity)과 최대가중치 최소비용 컷과 최적해의 속성을 이용하여 다음과 같은 사실이 성립함을 보였다.

**정리 1.[11]**

(1) $V_1$의 임의의 부분집합 $S$에 대해서 최대가중치 최소비용 $1-S$컷은 유일하게 존재한다.

(2) $S' \subseteq S'' \subseteq S \subseteq V_1$인 $S', S'', S$에 대해서 $\delta(S)$가 최대가중치 최소비용 $1-S'$컷이면, $\delta(S)$는 동시에 최대가중치 최소비용 $1-S''$컷이다.

(3) 분리가능-노드집합 $S$가 네트워크 단절문제의 최적해라고 하자. 그러면, $\delta(S)$는 $S$의 임의의 부분집합 $S'$에 대해서 최대가중치 최소비용 $1-S'$컷을 이룬다.

최대가중치 최소비용 $1-S$컷은 노드 1에서 $S$에 속한 노드들까지의 최대흐름양을 구하는 최대흐름문제(maximum flow problem)를 풀어서 구할 수 있다. 정리 1의 (3)은 우리가 최적해를 탐색할 때 최대가중치 최소비용 컷을 이루는 노드들의 집합만을 탐색 대상으로 해도 된다는 것을 암시한다. 그리고 우리는 그러한 집합은 특정한 유형의 노드집합의 합집합으로 구성되어 있음을 발견하였다. 각 노드 $i \in V_1$에 대해서 $S_i$를 $\delta(S_i)$가 최대가중치 최소비용 $1-\{i\}$컷이 되는 $V_1$의 부분집합으로 정의한다. 그러면 다음이 성립한다.

**정리 2.** $V_1$의 임의의 부분집합 $S$와 임의의 노드 $i \in V_1 \setminus S$에 대한 최대가중치 최소비용 $1-S \cup \{i\}$컷에 의하여 분할되는 그래프의 두 부분 중에서 $S \cup \{i\}$를 포함하는 부분은 $S_i$를 포함하게 된다

**(증명)** $S$가 공집합이면 자명하게 성립되므로 $S$는 공집합이 아니라고 가정하자. 그리고 최대가중치 최소비용 $1-S \cup \{i\}$컷에 의하여 분할되는 그래프의 두 부분 중에서 $S \cup \{i\}$를 포함하는 부분을 $S'$이라고 하고 $S'$이 $S_i$를 포함하지 않는다고 가정하자. 우선 컷의 서브-모듈라 특성 때문에 $c(\delta(S')) + c(\delta(S_i)) \geq c(\delta(S' \cup S_i)) + c(\delta(S' \cap S_i))$이 성립한다. 그런데 $S'$의 정의와 가정에 의해서 $c(\delta(S')) < c(\delta(S' \cup S_i))$이므로 $c(\delta(S_i)) > c(\delta(S' \cap S_i))$이 되어서 $S_i$의 정의에 모순된다. □

**따름정리 1.** 네트워크 단절문제의 최적해는 $V_1$의 부분집합 $I$에 대하여 $\bigcup_{i\in I} S_i$의 형태를 이루고 있다.

**(증명)** 정리 2에 의해서 임의의 노드 $i\in V_1$가 최적해에 포함되면, $S_i$도 최적해에 포함된다. □

따름정리 1은 우리가 최적해를 이루는 노드 집합을 구성할 때 노드별로 포함 여부를 결정하기보다는 $S_i$별로 포함 여부를 결정하는 것이 바람직함을 암시한다. 각 노드 $i\in V_1$에 대해서 $S_i$가 노드 $i$ 외에 여러 노드를 포함하는 경우에는 서로 구별되는 $S_i$의 개수는 총 노드 수보다 적어서 최적해를 탐색할 때 도움이 된다. 이러한 사실은 다음 장에서 실행가능해를 구하는 휴리스틱을 개발할 때 유용하게 사용된다.

어떤 노드 $i$에 대해서 최소비용 $1-\{i\}$컷의 비용이 예산을 초과한다면 최적해가 되는 분리가능-노드집합은 노드 $i$를 포함할 수 없다. 따라서 원래의 그래프에 대한 최적해와 노드 1과 $i$를 축약한 그래프에서의 최적해는 동일하게 될 것이다. 여기서 두 노드의 축약은 두 노드를 종단노드로 하는 에지들을 제거하고 두 노드를 하나의 노드로 합치는 것을 의미한다. 이 때, 두 노드 중 어느 한 노드에 연결되었던 에지는 새로이 합쳐진 노드로 연결되게 되고 축약의 결과로 동일한 노드를 종단노드로 하게 되는 에지(self-loop)는 그래프에서 제거된다. 또한 축약의 결과로 두 노드 사이에는 두 노드를 동일한 종단노드로 하는 복수의 에지(multiple edge)가 새로 발생할 수도 있다.

일반적으로 망을 대상으로 하는 문제를 푸는데 걸리는 시간은 망의 규모가 커짐에 따라 기하급수적으로 늘어난다. 따라서 배제하여도 최적해에 영향을 주지 않는 노드나 링크를 알 수 있다면 망의 규모를 축소할 수 있어서 문제 해결에 소요되는 시간을 줄일 수 있다. Myung and Kim[11]은 이러한 점에 착안하여 $S_i$가 분리가능-노드집합이 아니면 노드 1과 노드 $i$를 축약시키는 사전처리과정(preprocessing procedure)을 해법에 활용하였다. 앞으로 상한과 하한을 구하는 방법을 고려할 때는 이미 사전처리가 끝나서 모든 $S_i$는 분리가능-노드집합임을 가정하기로 한다.

## 3. 하한을 구하는 해법들

네트워크 단절문제의 최적해를 구하기 위한 하나의 방법은 모든 분리가능-노드집합에 대하여 가중치의 합을 비교하는 것이다. 그러나 대상이 될 수 있는 분리가능-노드집합의 수는 매우 많기 때문에 모든 분리가능-노드집합을 분석하는 것은 현실적으로 쉽지 않으며, 이러한 사실이 이 문제가 NP-hard에 속하는 문제임을 암시하고 있다. 따라서 바람직한 접근방법은 원 문제의 최적해에 가까운 실행가능해를 구하는 것이다. 3장에서는 2장에서 소개한 최적해의 특성을 바탕으로 일부 $S_i$들의 합집합으로 구성된 분리가능-노드집합을 구하는 휴리스틱과 타부 탐색 알고리즘을 이용한 휴리스틱을 개발하기로 한다.

### 3.1 $S_i$의 합집합으로 해를 구하는 휴리스틱

정리 2에 의하여 최적해는 $V_1$의 부분집합 $I$에 대하여 $\bigcup_{i\in I} S_i$의 형태를 이루고 있다. 따라서 실행가능해 $S$를 구하는 한가지 전략은 $V_1$의 부분집합 $I$를 선택하여 $S=\bigcup_{i\in I} S_i$으로 정하는 것이다. 그러나 $I$를 선택할 때 $I$에 선택될 노드를 어떤 방법으로 정하느냐에 따라서 얻어지는 해가 달라지게 된다. 조합최적화문제의 해법들에서 이러한 집합을 선택하는 방법은 크게 두 가지로 나누어 볼 수 있다. 우선 공집합에서 출발하여 노드를 추가함으로써 $I$를 구성하는 추가형 휴리스틱(add-type heuristic)과 전체집합에서 출발하여 노드를 제거함으로써 $I$를 구성하는 탈락형 휴리스틱(drop-type heuristic)이 두 가지 방법이다. 또한 각각의 방법도 추가 또는 제거될 노드를 어떤 기준에 의해서

선택하느냐에 따라서 다양하게 구성될 수 있다.

● 추가형 휴리스틱

Input : $G=(V,E)$, $\{c_e\}$, $\{w_i\}$

Output : 분리가능-노드집합 $S$

(단계 1) $S=\varnothing$으로 초기화.

(단계 2) $J=\{i\in V_1-S \mid c(\delta(S\cup S_i)\leq b\}$가 공집합이 될 때까지 다음을 반복 :
정해진 방법에 따라 $i\in J$를 선택하여 $S=S\cup S_i$로 변경한다.

● 탈락형 휴리스틱

Input : $G=(V,E)$, $\{c_e\}$, $\{w_i\}$

Output : 분리가능-노드집합 $S$

(단계 1) $J=V_1$, $S=\bigcup_{i\in J} S_i$으로 초기화.

(단계 2) $c(\delta(S)\leq b$가 될 때까지 다음을 반복 :
정해진 방법에 따라 $k\in J$를 선택하여 $S=\bigcup_{i\in J\setminus\{k\}} S_i$로 변경한다.

본 논문에서는 추가 또는 제거될 노드를 선택하는 기준으로 두 가지 방법을 실험하여 보았다. 하나는 노드를 추가 또는 제거할 때 발생하는 에지비용의 변화에 대한 노드의 가중치의 변화의 비율을 고려하는 것이며 또 다른 방법은 4절에서 소개되는 선형계획 완화문제의 노드 변수의 값의 크기에 따라 선택하는 것이다. 이 두 가지를 추가형 휴리스틱에 적용하면 첫째 전략은 (단계 2)에서 $i\in J$를 선택할 때 $\dfrac{w(S_i-S)}{c(\delta(S\cup S_i))-c(\delta(S))}$이 최대가 되는 $S_i$를 우선 선택하는 것이고 둘째 전략은 선형계획 완화문제의 노드 $i$에 대응되는 변수의 값이 최대가 되는 $S_i$를 우선 선택하는 것이다.

## 3.2 타부 탐색 휴리스틱

$V_1$의 부분집합 $S'$에 대해, 최소비용 $1-S'$컷은 $\delta(S)$가 되는 $V_1$의 부분집합 $S$가 존재하고, $\delta(S)$는 최대가중치 최소비용 $1-S$컷이다. 따라서, $V_1$의 부분집합 $S$를 선택하여, $c(\delta(S))$이 예산 $b$ 이하라면 이는 네트워크 단절문제의 실행가능해라는 것을 뜻하고, 이 때 $S$는 분리되는 노드의 집합을, $\delta(S)$는 제거해야 하는 에지의 집합을 의미하게 된다. 따라서 이를 이용하여 $V_1$의 부분집합 $S$를 검색하는 타부 탐색 휴리스틱[3-5]을 생각할 수 있다.

그러나, 네트워크 단절문제를 해결하기 위해 위와 같이 노드의 집합만을 고려하는 타부 탐색 휴리스틱을 생각할 경우, 가능해에서 가능해로의 이동만을 고려하는 방식은 휴리스틱의 성능을 크게 떨어뜨릴 수 있다[14]. 따라서, 비가능인 해에 대한 탐색을 허용하며 이는 비가능도를 정의하여 사용가능하게 한다. 이 때, 비가능도는 $c(\delta(S))$의 값이 $b$보다 클 경우 $c(\delta(S))-b$, 그 외의 경우는 0의 값을 가진다. 주목해야 할 점은 휴리스틱의 이러한 특성 때문에 사전처리를 통해 분리되지 않는 노드의 집합을 알고 있음에도 불구하고, 분리되지 않는 노드를 노드 1과 축약하지 않고 그대로 사용한다는 점이다.

타부 탐색 휴리스틱은 비교적 널리 알려져 있는 휴리스틱이므로, 이에 대한 설명은 생략하며 자세한 내용은 Kim *at el.*[8]을 참고하기 바란다. 사용한 타부 탐색 휴리스틱의 주요 특징만을 요약하면 다음과 같다. 단, 여기서 고려하는 해는 선택한 노드의 집합을 나타낸다.

- 이웃해 : 현재의 해에 대해 포함되어 있지 않은 하나의 노드를 추가하거나 포함되어 있는 하나의 노드를 탈락함으로써 얻어지는 모든 해로 정의한다.
- 이동 : 타부 목록에 없는 가능해를 선택한다. 만약 타부 목록에 없는 가능해가 없을 경우, 타부 목록에 없는 비가능도가 가장 작은 비가능인 해를 선택한다.
- 속성 : 타부에 저장하는 방식을 나타내는 것으로써, 추가되거나 탈락한 노드로 정의한다. 단 이

때 추가와 탈락의 구분은 하지 않고 어떤 노드인지만을 저장한다.

- 타부의 크기 : (전체 노드의 수)×0.3으로 한다.
- 열망 기준 : 두 가지의 열망 기준을 둔다. 하나는 목적식의 값이 기존에 찾은 가장 좋은 해의 목적식의 값보다 더 큰 경우이고, 다른 하나는 직전의 이동이 비가능인 해를 선택해서 이루어진 이동이었을 경우 현재의 반복시행에서 비가능도를 감소시키는 경우이다.
- 장기 메모리 : 기존에 찾은 가장 좋은 해보다 목적식의 값이 더 큰 가능해를 찾았을 경우, 이웃해 중에서 목적식의 값이 그 다음으로 큰 가능해와 비가능도가 가장 큰 비가능인 해를 저장한다.
- 종료 조건 : 반복횟수의 수가 *MAXITER*를 넘으면 알고리즘을 종료한다. 우리는 *MAXITER*를 노드 수에 15를 곱한 값으로 사용하였다.

다음으로 전반적인 알고리즘에 대하여 살펴보자. 설명을 돕기 위하여 어떤 반복시행에서 현재의 해를 $S_{current}$, 이웃해 중 가장 좋은 가능해를 $S_{best}$, 두 번째로 좋은 가능해를 $S_{second}$, 비가능도가 가장 큰 비가능인 해를 $S_{worst}$라 한다. 실제 알고리즘 수행 시 각각의 역할을 살펴보면, $S_{best}$는 현재까지의 가장 좋은 해를 저장하는 역할을 하며, $S_{second}$와 $S_{worst}$는 좋은 해를 찾을 때마다 장기 메모리를 업데이트할 때 사용된다. 그 외에, $T^*$를 사전처리를 통해 얻어진 분리되지 않는 노드의 집합이라 한다. 초기해를 구하는 알고리즘, 이웃해를 생성하고 이동하는 알고리즘, 전체적인 타부 탐색 알고리즘은 다음과 같다.

**● 초기해 발생 알고리즘**

(단계 1) $T \leftarrow T^* \cup \{1\}$, $t \leftarrow 1$로 놓는다.

(단계 2) $V_1 \backslash T$에 있고 $T$에 인접한 노드들 중에서 하나를 추가한다. 이 때 추가되는 노드를 노드 $i$라 하자.

(단계 3) $c(\delta(T \cup \{i\})) > b$이면 $T \leftarrow T \cup \{i\}$, $t \leftarrow t+1$로 바꾸고, (단계 2)로 간다. 만약 $c(\delta(T \cup \{i\})) \le b$이면 $T \cup \{i\}$를 초기해로 하고 종료하고, $t = |V_1| - |T^*|$이면 $T^*$를 초기해로 한 후 종료한다.

**● 이웃해 생성과 이동**

(단계 1) 이웃해들을 생성하고, 모든 이웃해를 선택하지 않은 해로 초기화한다.

(단계 2) 선택하지 않은 이웃해 중 목적식의 값이 가장 큰 가능해가 있으면 (단계 3)으로 가고, 없으면 (단계 4)로 간다.

(단계 3) 선택한 가능해가 타부인지 검사한다. 타부가 아니라면 선택한 해를 $S_{current}$로 바꾸고 타부에 저장한 후 (단계 6)으로 간다. 이 때 만약 선택한 해가 $S_{best}$보다 좋은 해라면 선택한 해를 $S_{best}$에 저장하고, $S_{second}$와 $S_{worst}$를 장기 메모리에 저장한다. 만약 선택한 가능해가 타부라면 첫 번째 열망 기준을 만족하는지 검사한다. 즉, $S_{best}$에 비해 목적식의 값이 증가하면 선택한 해를 $S_{current}$, $S_{best}$, 타부에 저장하고 $S_{second}$와 $S_{worst}$를 장기 메모리에 저장한 후 (단계 6)으로 간다. 그 외의 경우 검사했던 해를 선택했던 해로 바꾸고 (단계 2)로 간다.

(단계 4) 선택하지 않은 이웃해 중 비가능도가 가장 작은 비가능인 해가 있으면 (단계 5)로 가고, 없으면 (단계 7)로 간다.

(단계 5) 선택한 비가능인 해가 타부인지 검사한다. 타부가 아니라면 선택한 해를 $S_{current}$로 바꾸고 타부에 저장한 후 (단계 6)으로 간다. 만약 선택한 가능해가 타부라면 두번째 열망 기준을 만족하는지 검사한다. 즉, 지난 반복에서 이동으로 채택되었던 해가 비가능인 해이고, 현재의 비가능인 해보다 비가능도가 더 큰 경우 선택한 해를 $S_{current}$로 바꾸고 타부에 저장한 후 (단계 6)으로 간다. 그 외의 경우 검사했던 해를 선택했던 해로 바꾸고 (단계 4)로 간다.

(단계 6) 이동할 수 있는 해를 구했으므로 종료한

다. 이 해는 다음 반복시행에서 이웃해를 만들어 내기 위한 초기해로 사용된다.

(단계 7) 이동할 수 있는 해를 구하지 못한 경우이므로 (단계 6)과 마찬가지로 종료한다. 이 경우 다음 반복시행을 위해 장기 메모리의 해를 초기해로 사용한다. 만약 장기 메모리에 검색하지 않은 해가 없을 경우, 임의추출방식을 이용하여 초기해를 생성한다. 이를 위해 사전 처리를 통해 얻은 분리되지 않는 노드들은 기본적으로 선택하고, 나머지 노드들에 대해서는 0과 1 사이의 임의의 수를 생성하여 확률적으로 선택한다. 즉, 임의추출한 수가 *THRESH* 보다 크면 선택하고 작으면 선택하지 않도록 한다(단 *THRESH*는 미리 정해지는 상수이다.).

● **타부 탐색 알고리즘**

(단계 1) 초기해를 생성하여 $S_{current}$로 두고, 반복횟수 $k=1$, 해가 개선되지 않은 횟수 $k_{ub}=0$으로 둔다.

(단계 2) 이웃해 생성과 이동을 실행하고 $k \leftarrow k+1$로 바꾼다. 이 때 $S_{best}$의 변화가 없으면 $k_{ub} \leftarrow k_{ub}+1$로 변화가 있으면 $k_{ub}=0$으로 바꾼다.

(단계 3) $k_{ub} > MAXUB$이거나 (단계 2)의 결과가 이동가능하지 않은 종료였을 경우, (단계 4)로 간다. $k > MAXITER$이면 알고리즘을 종료하고 $S_{best}$를 최적해로 한다. 그 외의 경우 (단계 2)로 간다.

(단계 4) 장기 메모리의 해를 $S_{current}$로 둔다. 만약 장기 메모리에 검색하지 않은 해가 없을 경우, 임의 추출한 해를 $S_{current}$로 둔다. $k_{ub}=0$으로 바꾸고 (단계 2)로 간다

## 4. 상한의 도출

조합최적화문제의 상한(최대화문제의 경우)을 구하는 전형적인 방법의 하나는 해당 문제의 정수계획모형을 수립하고 선형계획 완화문제를 풀어서 상한을 구하는 것이다. 특정의 에지 $e$의 제거 여부를 나타내는 변수 $x_e$와 노드 $i$가 1에서 분리되는지 여부를 나타내는 변수 $y_i$를 다음과 같이 정의하자.

$$x_e=\begin{cases}1, & \text{에지 } e\text{가 제거되었을 때}\\ 0, & \text{위와 다른 경우}\end{cases}$$

$$y_i=\begin{cases}1, & \text{노드 } i\text{와 1이 단절된 경우}\\ 0, & \text{위와 다른 경우}\end{cases}$$

Myung and Kim[11]은 네트워크 단절문제가 다음과 같은 정수계획모형 ($P1$)으로 나타낼 수 있음을 밝혔다.

$$(P1)\quad \max\ z=\sum_{i\in V_1} w_i y_i \tag{1}$$

$$s.t.\ \sum_{e\in E} c_e x_e \le b, \tag{2}$$

$$\sum_{e\in E(p)} x_e \ge y_i,\ \forall p\in P_i,\ \forall i\in V_1, \tag{3}$$

$$x_e\in\{0,1\},\quad \forall e\in E, \tag{4}$$

$$y_i\in\{0,1\},\quad \forall i\in V_1, \tag{5}$$

여기서 $P_i$는 그래프 $G$에 존재하는 노드 1과 노드 $i$ 사이의 모든 경로(path)의 집합을 나타내고, $E(p)$는 경로 $p$에 속한 에지의 집합을 나타낸다. 목적식 (1)은 노드 1에서 단절되는 노드의 가중치의 합을 최대화시키는 식이고, 제약식 (2)는 예산제약을 반영하는 식이다. 그리고 제약식 (3)은 노드 $i$가 노드 1로부터 단절되기 위해서는 두 노드 사이의 경로마다 적어도 한 에지는 제거되어야 함을 의미하는 것이다. 제약식 (4)와 식 (5)는 변수들의 정수조건을 나타낸다.

이제 새로운 정수계획모형을 소개하기로 한다. 새로운 정수계획모형 ($P2$)는 노드 변수와 에지 변수간의 관계를 표시하는 제약식 (3)을 다음과 같은 제약식으로 대체한 모형이다.

$$x_e \geq y_i - y_j,\ x_e \geq y_j - y_i,\ \forall e=\{i,j\} \in E, \quad (6)$$

$$y_1 = 0, \quad (7)$$

제약식 (6)은 노드 1에 연결된 노드( $y_i$가 0인 노드)와 단절된 노드( $y_i$가 1인 노드) 사이에 걸친 에지는 제거되어야 함을 의미한다.

두 정수계획모형 ($P1$)과 ($P2$)의 최적해는 동일하고 네트워크 단절문제의 최적해가 된다. 그러나 두 모형의 선형계획 완화문제 중 어느 쪽이 더 효율적인 상한을 제공하는지, 또한 계산시간이 적게 소요되는지는 관심거리이다. 우선 첫째 질문에 대해 분석해 보기로 하자. 정수계획모형 ($P1$)과 ($P2$)에서 제약식 (4)와 식 (5)를 각각 $0 \leq x_e \leq 1, \forall e \in E$, $0 \leq y_i \leq 1, \forall i \in V_1$로 대체함으로써 얻어지는 선형계획 완화문제를 각각 ($LP1$)과 ($LP2$)라고 하고 ($LP1$)과 ($LP2$)의 최적목적함수 값을 각각 $V(LP1)$과 $V(LP2)$로 표기하자. 그러면 다음이 성립함을 보일 수 있다.

**정리 3.** $V(LP1) = V(LP2)$.

**(증명)** (i) 우선 $V(LP1) \geq V(LP2)$이 성립함을 증명하기로 한다. 이를 위해서 ($LP2$)의 실행가능영역이 ($LP1$)의 실행가능영역에 포함됨을 보이기로 한다. ($LP2$)의 임의의 실행가능해를 $(x, y)$라 하자. 그러면 $(x, y)$가 노드 1에서 임의의 노드 $i \in V_1$까지의 경로 $p$에 대한 제약식 (3)을 만족시킴을 보이기로 한다. 표기의 편의를 위해서 경로 $p$는 노드 $1, 2, 3, \cdots, i$로 구성되었다고 가정하자. 그러면 식 (6)에서 $x_{\{k,k+1\}} \geq y_{k+1} - y_k$, $k = 1, \cdots, i-1$이므로 $\sum_{e \in E(p)} x_e = x_{\{1,2\}} + , \cdots, + x_{\{i-1,i\}} \geq y_i - y_1 = y_i$이 성립된다.

(ii) $V(LP1) \leq V(LP2)$가 성립되는 것은 ($LP1$)의 최적해 중 적어도 하나가 ($LP2$)의 실행가능해임을 보임으로써 증명하기로 한다. ($LP1$)의 최적해 $(x, y)$가 ($LP2$)의 실행가능해가 아닌 경우를 고려해 보자. 그러면 $(x, y)$가 어떤 $e = \{i, j\} \in E$에 대해서 제약식 (6)을 만족시키지 못하는 경우이다. 일반성을 손상함이 없이 $y_i > y_j$라고 가정할 수 있다. 그러면 $(x, y)$가 제약식 (6)을 만족시키지 못하므로 $x_{\{i,j\}} < y_i - y_j$가 된다. 이 경우에 주어진 그래프에서 노드 1에서 노드 $j$까지의 모든 경로는 항상 노드 $i$를 지나게 된다. 왜냐하면 노드 $i$를 거치지 않는 노드 1에서 노드 $j$까지의 경로 $p$가 존재한다면 $(x, y)$는 $p$와 에지 $\{i, j\}$가 결합된 노드 1에서 노드 $i$까지의 경로에 대한 제약식 (3)을 만족하지 못하기 때문이다. 주어진 그래프에서 노드 1에서 노드 $j$까지의 모든 경로는 항상 노드 $i$를 지나게 되면 $y_j$를 $y_i$와 같도록 증가시켜도 노드 1에서 노드 $j$까지의 모든 경로에 대한 제약식 (3)을 만족시킬 수 있다. 그리고 이렇게 변동된 해는 목적함수 값을 감소시키지 않으며 $e = \{i, j\} \in E$에 대해서 제약식 (6)을 만족하게 된다. □

이처럼 두 정수계획모형은 동일한 상한을 제공하지만 모형이 상이하므로 선형계획 완화문제를 푸는 시간은 동일하지 않을 것이다. 일단 문제의 크기만을 비교한다면 두 모형은 $y_1$을 제외하면 동일한 변수를 갖고 있으나 ($LP1$)의 제약식 (3)은 그래프의 크기가 커지면 기하급수적으로 늘어나는데 반해서 ($LP2$)의 제약식 (6)은 에지의 개수에 2배만 존재하므로 ($LP2$)의 모형이 훨씬 간단하다. 하지만, Myung and Kim[11]처럼 ($LP1$)을 절단면해법(cutting plane algorithm)을 이용하여 푸는 경우에는 제약식 (3)의 개수가 많다고 해서 계산시간이 많이 필요하다고 결론을 내리기는 어렵다.

Myung and Kim[11]의 절단면해법은 우선 목적식과 제약식 (2), $0 \leq x_e \leq 1$, $\forall e \in E$, $0 \leq y_i \leq 1$, $\forall i \in V_1$만을 포함하는 선형계획문제를 풀면서 시작한다. 매 단계에서 현재 구해진 선형계획문제의 해가 모든 제약식 (3)을 만족하는지를 판정하고, 아닌 경우에 현재의 해가 만족시키지 못하는 제약식이 어떤 것인지를 결정하기 위하여 제약식 (3)에 대한 분리문제(separation problem)를 풀게

된다. 제약식 (3)에 해당하는 분리문제는 최단경로 문제(shortest path problem)를 풀어서 해결할 수 있다. 최적해 $\{x_e, y_i\}$를 구하여, $x_e$를 에지 $e$의 길이로 하는 그래프에서 노드 1로부터 각 노드 $i \in V_1$까지의 최단경로를 구하고 그 길이 $l_i$를 구하여, 만일 $l_i < y_i$이면 이 최단경로 $p$에 대해서 제약식 (3)을 추가한다. 왜냐하면, $x_e$를 에지 $e$의 길이로 하는 그래프에서, 두 노드 1과 $i$사이의 최단경로의 길이가 $y_i$ 이상이면 1과 $i$사이의 모든 경로의 길이가 $y_i$ 이상이므로 현재의 선형계획문제의 해 $\{x_e, y_i\}$는 노드 $i$에 대한 모든 제약식 (3)을 만족하게 되고, 만약 어떤 경로 $p$에 대한 경로의 길이가 $y_i$ 미만이면 $\{x_e, y_i\}$는 해당 경로에 대해서 식 (3)을 만족시키지 못하기 때문이다.

이처럼 제약식 (3)에 해당하는 분리문제가 다항시간 내에 풀리기 때문에 ($LP1$)도 다항시간 내에 풀 수 있으므로 두 선형계획문제의 이론적 계산시간은 같다고 할 수 있다. 따라서 실질적인 두 모형의 계산시간을 비교하는 것도 흥미로운 과제이며 이는 5장에서 다루기로 한다.

## 5. 계산실험 및 결과 분석

본 연구에서 제시된 네트워크 단절문제의 상한과 하한을 구하기 위한 절차는 C 언어를 통하여 프로그램으로 구현되었다. 물론 계산실험에 적용할 대상문제를 만드는 프로그램도 같이 구현되었으며, 상한을 구하는 선형계획 완화문제를 반복적으로 풀어 가기 위해서 CPLEX 라이브러리를 이용하였다. 계산실험을 위하여 30개에서 80개까지의 노드로 구성되는 가상의 문제를 만들었다. 가상의 문제를 만들기 위해서는 먼저 (100×100) 사각형 모양의 격자에 필요한 수만큼의 노드를 임의로 위치시키고, 다음에 노드간에 정해진 수의 에지를 위치시켰다. 에지의 선택은 우선 노드들이 상호 연결되도록 하나의 걸침나무(spanning tree)를 이루도록 에지를 우선 선택하고, 지정된 수만큼의 추가적인 링크를 설치하였다. 노드의 가중치는 1에서 20사이에서 난수(random number)를 이용하여 임의로 추출하였고 에지의 제거비용도 1에서 20사이에서 난수를 이용하여 임의로 생성하였으며, 에지 제거를 위한 예산은 원천노드에 연결된 에지비용의 합의 일정한 비율로 설정하였다. 이 비율은 1 이하의 값에서 선택되었다. 왜냐하면 원천노드에 연결된 에지를 모두 제거하면 모든 노드를 원천노드에서 단절시킬 수 있으므로 예산이 이 비용을 초과하는 경우에는 자명한 문제가 되기 때문이다.

작성된 코드는 2.0Ghz CPU를 가진 펜티엄급 PC에서 실행되어 졌다. 다양한 크기의 네트워크에서 서로 다른 비용 및 가중치를 갖는 많은 문제에 대한 시험계산을 수행하였다. <표 1>~<표 3>에 나타난 것처럼 문제별로 주어진 그래프의 크기를 노드의 수와 에지의 수로 표시하였고, 예산의 경우는 원천노드에 연결된 에지제거 비용의 합계에 대한 비율로 표시하였다. 앞서 언급한대로 상한과 하한을 구하기 전에 사전처리과정을 거치게 되는데 이를 위해서 사전처리가 이루어진 이후 그래프의 크기를 표시하였다.

우선 3장에서 제시한 하한을 구하는 해법인 추가형 휴리스틱과 탈락형 휴리스틱 및 타부 탐색 휴리스틱의 성능을 분석하였다. 추가형 휴리스틱과 탈락형 휴리스틱에 대해서는 추가 또는 제거될 노드를 선택하는 기준 두 가지에 대해서 각각 실험하여 보았다. Table에서 Add1과 Drop1은 추가형 휴리스틱과 탈락형 휴리스틱에서 노드를 추가 또는 제거할 때 발생하는 에지비용의 변화에 대한 노드의 가중치의 변화의 비율을 고려한 해법의 결과를 각각 나타내고 Add2와 Drop2는 두 휴리스틱에서 선형계획 완화문제의 노드 변수의 값의 크기에 따라 선택하는 해법을 각각 나타낸다. Tabu는 타부 탐색 휴리스틱의 결과를 의미한다. 타부 탐색 휴리스틱에서 MAXITER는 노드 수의 15배로 하였고 THRESH는 0.5, MAXUB는 30으로 선택하였다. 추가형 휴리스틱은 일단 알고리즘을 이용하여서

해가 구해지면 해에 포함되지 않은 $S_i$를 제일 먼저 포함시키고 다시 추가형 휴리스틱을 반복하는 방법으로 모든 $S_i$가 한 번씩은 포함되도록 해를 탐색하였다.

한편으로 상한을 구하기 위에서 사용된 선형계획 완화모형 ($LP1$)과 ($LP2$)의 계산시간을 비교하였다. 두 모형을 푸는데 걸린 시간을 ($LP1$)과 ($LP2$)로 구별하여 표시하였다. 상한의 값을 표현하는 열에는 ($LP1$)과 ($LP2$)의 목적함수는 같으므로 별도로 구분하지 않고 (UB)로 표시하였다. Myung and Kim[11]은 ($LP1$)과 ($LP2$)을 이용한 상한의 효율을 높이기 위하여 유효부등식을 이용하였는데, ($LP1$)이나 ($LP2$)에 유효부등식을 추가하여 구해지는 상한을 (UBV)로 표시하여 제

〈표 1〉 시험계산결과(노드 30개인 경우)

| 원네트워크 | | | 사전처리후 | | 하 한 값 | | | | | 상 한 값 | | 실 행 시 간 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| \|V\| | \|E\| | b/c(E) | \|V\| | \|E\| | Add1 | Add2 | Drop1 | Drop2 | Tabu | UB | UBV | LP1 | LP2 | Add1 | Drop1 | Tabu |
| 30 | 100 | 0.40 | 3.6 | 3.1 | 4.56 | 4.56 | 4.56 | 4.56 | 4.56 | 5.28 | 4.84 | 0.003 | 0.002 | 0.000 | 0.000 | 0.006 |
| 30 | 100 | 0.45 | 4.5 | 4.3 | 5.33 | 5.33 | 5.20 | 5.33 | 5.33 | 5.95 | 5.33 | 0.001 | 0.003 | 0.000 | 0.000 | 0.005 |
| 30 | 100 | 0.50 | 5.4 | 5.6 | 6.26 | 6.22 | 5.60 | 6.22 | 6.26 | 7.15 | 6.39 | 0.003 | 0.005 | 0.000 | 0.000 | 0.011 |
| 30 | 100 | 0.55 | 6.0 | 6.4 | 6.68 | 6.49 | 6.28 | 6.49 | 6.65 | 7.83 | 6.96 | 0.005 | 0.002 | 0.000 | 0.000 | 0.008 |
| 30 | 100 | 0.60 | 7.0 | 8.2 | 7.08 | 7.08 | 7.08 | 7.08 | 7.08 | 8.29 | 7.37 | 0.005 | 0.002 | 0.000 | 0.000 | 0.006 |
| 30 | 100 | 0.65 | 8.3 | 11.8 | 8.44 | 8.44 | 8.44 | 8.44 | 8.44 | 9.06 | 8.90 | 0.000 | 0.008 | 0.000 | 0.000 | 0.010 |
| 30 | 100 | 0.70 | 9.6 | 15.0 | 8.75 | 8.44 | 7.92 | 8.08 | 8.75 | 9.83 | 9.23 | 0.006 | 0.005 | 0.000 | 0.000 | 0.014 |
| 30 | 100 | 0.75 | 11.3 | 21.9 | 8.87 | 8.56 | 8.18 | 8.59 | 8.87 | 10.71 | 9.86 | 0.008 | 0.003 | 0.000 | 0.000 | 0.011 |
| 30 | 100 | 0.80 | 12.4 | 26.7 | 9.65 | 9.20 | 7.73 | 9.14 | 9.65 | 12.23 | 10.86 | 0.006 | 0.005 | 0.000 | 0.000 | 0.016 |
| 30 | 100 | 0.85 | 14.8 | 36.4 | 18.94 | 18.58 | 16.53 | 17.88 | 18.94 | 21.21 | 19.73 | 0.009 | 0.005 | 0.000 | 0.000 | 0.014 |
| 30 | 100 | 0.90 | 16.3 | 43.3 | 19.59 | 18.90 | 18.07 | 18.43 | 19.59 | 22.53 | 20.78 | 0.011 | 0.005 | 0.002 | 0.000 | 0.017 |
| 30 | 100 | 0.95 | 16.8 | 45.1 | 19.59 | 18.57 | 17.55 | 17.91 | 19.74 | 23.38 | 21.30 | 0.019 | 0.005 | 0.000 | 0.001 | 0.016 |
| 30 | 200 | 0.40 | 1.1 | 0.1 | 0.54 | 0.54 | 0.54 | 0.54 | 0.54 | 0.54 | 0.54 | 0.002 | 0.003 | 0.000 | 0.000 | 0.006 |
| 30 | 200 | 0.45 | 1.3 | 0.3 | 1.19 | 1.19 | 1.19 | 1.19 | 1.19 | 1.19 | 1.19 | 0.003 | 0.002 | 0.000 | 0.000 | 0.006 |
| 30 | 200 | 0.50 | 1.4 | 0.4 | 1.41 | 1.41 | 1.41 | 1.41 | 1.41 | 1.41 | 1.41 | 0.003 | 0.006 | 0.000 | 0.000 | 0.005 |
| 30 | 200 | 0.55 | 1.7 | 0.7 | 1.62 | 1.62 | 1.62 | 1.62 | 1.62 | 1.63 | 1.62 | 0.002 | 0.002 | 0.000 | 0.000 | 0.011 |
| 30 | 200 | 0.60 | 2.4 | 1.7 | 2.35 | 2.35 | 2.35 | 2.35 | 2.35 | 2.41 | 2.35 | 0.005 | 0.000 | 0.000 | 0.000 | 0.008 |
| 30 | 200 | 0.65 | 3.2 | 2.9 | 3.54 | 3.54 | 3.54 | 3.54 | 3.54 | 3.88 | 3.54 | 0.003 | 0.006 | 0.000 | 0.000 | 0.008 |
| 30 | 200 | 0.70 | 3.6 | 3.6 | 4.17 | 4.17 | 4.17 | 4.17 | 4.17 | 4.66 | 4.17 | 0.000 | 0.003 | 0.000 | 0.000 | 0.011 |
| 30 | 200 | 0.75 | 4.5 | 4.9 | 4.65 | 4.65 | 4.59 | 4.65 | 4.59 | 5.27 | 4.65 | 0.006 | 0.000 | 0.000 | 0.000 | 0.012 |
| 30 | 200 | 0.80 | 5.6 | 8.4 | 5.04 | 5.04 | 4.93 | 5.04 | 5.04 | 5.90 | 5.04 | 0.003 | 0.003 | 0.000 | 0.000 | 0.011 |
| 30 | 200 | 0.85 | 6.7 | 12.6 | 5.07 | 5.07 | 4.85 | 5.07 | 5.07 | 6.50 | 5.07 | 0.003 | 0.002 | 0.000 | 0.000 | 0.016 |
| 30 | 200 | 0.90 | 8.4 | 19.6 | 5.24 | 5.24 | 4.99 | 5.24 | 5.13 | 6.91 | 5.24 | 0.010 | 0.001 | 0.000 | 0.000 | 0.014 |
| 30 | 200 | 0.95 | 10.3 | 30.8 | 5.47 | 5.47 | 5.16 | 5.47 | 5.44 | 7.93 | 5.47 | 0.008 | 0.003 | 0.000 | 0.000 | 0.019 |
| 30 | 300 | 0.40 | 1.0 | 0.0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.002 | 0.000 | 0.000 | 0.000 | 0.009 |
| 30 | 300 | 0.45 | 1.0 | 0.0 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.002 | 0.005 | 0.000 | 0.000 | 0.006 |
| 30 | 300 | 0.50 | 1.2 | 0.2 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.005 | 0.003 | 0.000 | 0.000 | 0.013 |
| 30 | 300 | 0.55 | 1.8 | 0.9 | 1.47 | 1.47 | 1.47 | 1.47 | 1.47 | 1.48 | 1.47 | 0.004 | 0.004 | 0.000 | 0.000 | 0.007 |
| 30 | 300 | 0.60 | 2.3 | 1.8 | 1.75 | 1.75 | 1.71 | 1.75 | 1.75 | 1.90 | 1.75 | 0.006 | 0.003 | 0.000 | 0.000 | 0.008 |
| 30 | 300 | 0.65 | 3.5 | 5.2 | 2.94 | 2.94 | 2.87 | 2.94 | 2.94 | 3.17 | 2.94 | 0.003 | 0.003 | 0.000 | 0.000 | 0.011 |
| 30 | 300 | 0.70 | 4.9 | 9.5 | 4.03 | 4.03 | 3.96 | 4.03 | 4.03 | 4.41 | 4.03 | 0.003 | 0.008 | 0.000 | 0.000 | 0.017 |
| 30 | 300 | 0.75 | 6.4 | 16.3 | 4.74 | 4.74 | 4.67 | 4.74 | 4.59 | 5.36 | 4.74 | 0.006 | 0.001 | 0.000 | 0.000 | 0.016 |
| 30 | 300 | 0.80 | 9.1 | 33.8 | 5.55 | 5.55 | 5.31 | 5.55 | 5.32 | 6.40 | 5.55 | 0.008 | 0.005 | 0.000 | 0.000 | 0.020 |
| 30 | 300 | 0.85 | 12.5 | 68.4 | 5.58 | 5.58 | 5.31 | 5.58 | 5.52 | 7.65 | 5.58 | 0.019 | 0.008 | 0.000 | 0.000 | 0.031 |
| 30 | 300 | 0.90 | 14.7 | 88.1 | 6.08 | 6.08 | 5.54 | 6.08 | 5.91 | 9.47 | 6.08 | 0.023 | 0.009 | 0.002 | 0.000 | 0.031 |
| 30 | 300 | 0.95 | 17.3 | 122.0 | 6.11 | 6.11 | 5.58 | 6.11 | 5.88 | 13.71 | 6.11 | 0.031 | 0.009 | 0.000 | 0.001 | 0.033 |

시하였다. 분석이 용이하도록 하한과 상한을 총 가중치의 합에 대한 비율로 표현하였다. 각 수치는 10개 문제의 평균치이다.

표에 나타난 결과를 살펴보면 하한을 구하는 방법은 노드를 추가할 때 발생하는 에지비용의 변화에 대한 노드의 가중치의 변화의 비율을 고려한 추가형 휴리스틱과 타부 탐색 휴리스틱이 효과가 있음을 알 수 있다. 다만 타부 탐색 휴리스틱은 해를 구하기 위해서 수행하는 반복 시행이 늘어나면 해의 품질이 개선된다. 즉 실행시간과 연관하여 평가하여야 할 것이다. 상한을 구하는 선형계획 완화모형 ($LP1$)과 ($LP2$)의 계산시간은 문제에 따라 다르게 나타났으나 ($LP2$)의 계산시간이 더 안정적임을 확인할 수 있다. 문제의 규모가 커질수록 상한과 하한의 차이가 커지는 경향이 있으나 실험결과로 도출된 하한 및 상한의 성능, 실행시간 등

〈표 2〉 시험계산결과(노드 50개인 경우)

| 원네트워크 | | | 사전처리후 | | 하 한 값 | | | | | 상 한 값 | | 실 행 시 간 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| \|V\| | \|E\| | b/c(E) | \|V\| | \|E\| | Add1 | Add2 | Drop1 | Drop2 | Tabu | UB | UBV | LP1 | LP2 | Add1 | Drop1 | Tabu |
| 50 | 200 | 0.40 | 5.3 | 4.9 | 4.62 | 4.60 | 4.42 | 4.62 | 4.68 | 5.33 | 4.89 | 0.001 | 0.001 | 0.000 | 0.000 | 0.035 |
| 50 | 200 | 0.45 | 6.4 | 6.5 | 4.93 | 4.93 | 4.71 | 4.93 | 4.93 | 5.97 | 5.06 | 0.006 | 0.003 | 0.001 | 0.000 | 0.036 |
| 50 | 200 | 0.50 | 7.7 | 8.6 | 5.25 | 5.18 | 4.84 | 5.18 | 5.25 | 6.48 | 5.36 | 0.005 | 0.005 | 0.000 | 0.000 | 0.042 |
| 50 | 200 | 0.55 | 9.4 | 12.7 | 5.33 | 5.24 | 5.16 | 5.28 | 5.33 | 6.87 | 5.88 | 0.003 | 0.006 | 0.000 | 0.000 | 0.040 |
| 50 | 200 | 0.60 | 11.1 | 16.7 | 6.08 | 5.98 | 5.90 | 6.08 | 6.08 | 7.31 | 6.56 | 0.005 | 0.003 | 0.000 | 0.000 | 0.048 |
| 50 | 200 | 0.65 | 12.9 | 22.2 | 6.25 | 5.95 | 5.95 | 6.21 | 6.31 | 7.78 | 7.14 | 0.006 | 0.005 | 0.000 | 0.000 | 0.050 |
| 50 | 200 | 0.70 | 14.7 | 29.1 | 7.20 | 6.74 | 6.95 | 6.86 | 7.34 | 8.21 | 7.85 | 0.009 | 0.003 | 0.001 | 0.001 | 0.050 |
| 50 | 200 | 0.75 | 16.2 | 34.5 | 7.79 | 7.70 | 7.79 | 7.63 | 7.79 | 8.72 | 8.43 | 0.014 | 0.006 | 0.002 | 0.000 | 0.058 |
| 50 | 200 | 0.80 | 18.2 | 41.4 | 8.04 | 7.44 | 7.73 | 7.88 | 8.04 | 9.10 | 8.71 | 0.027 | 0.002 | 0.003 | 0.000 | 0.061 |
| 50 | 200 | 0.85 | 20.5 | 48.9 | 8.61 | 7.93 | 7.99 | 8.35 | 8.61 | 9.52 | 9.24 | 0.041 | 0.001 | 0.003 | 0.000 | 0.064 |
| 50 | 200 | 0.90 | 23.0 | 57.6 | 8.74 | 8.39 | 8.04 | 8.50 | 8.74 | 10.33 | 9.61 | 0.055 | 0.002 | 0.000 | 0.006 | 0.070 |
| 50 | 200 | 0.95 | 24.9 | 64.5 | 9.04 | 8.66 | 7.59 | 8.05 | 9.04 | 11.92 | 10.06 | 0.048 | 0.006 | 0.003 | 0.001 | 0.072 |
| 50 | 350 | 0.40 | 1.7 | 0.7 | 1.13 | 1.13 | 1.03 | 1.13 | 1.13 | 1.15 | 1.13 | 0.003 | 0.003 | 0.000 | 0.000 | 0.034 |
| 50 | 350 | 0.45 | 2.2 | 1.3 | 1.48 | 1.48 | 1.38 | 1.48 | 1.48 | 1.71 | 1.48 | 0.005 | 0.000 | 0.000 | 0.000 | 0.037 |
| 50 | 350 | 0.50 | 2.7 | 2.2 | 1.52 | 1.52 | 1.42 | 1.52 | 1.52 | 1.95 | 1.52 | 0.005 | 0.002 | 0.000 | 0.000 | 0.038 |
| 50 | 350 | 0.55 | 4.1 | 4.3 | 2.17 | 2.17 | 1.80 | 2.03 | 2.17 | 2.63 | 2.17 | 0.005 | 0.002 | 0.000 | 0.000 | 0.044 |
| 50 | 350 | 0.60 | 4.9 | 5.4 | 2.30 | 2.30 | 2.09 | 2.30 | 2.30 | 2.97 | 2.30 | 0.005 | 0.003 | 0.000 | 0.000 | 0.044 |
| 50 | 350 | 0.65 | 6.2 | 8.3 | 2.42 | 2.42 | 2.29 | 2.42 | 2.42 | 3.19 | 2.42 | 0.006 | 0.003 | 0.000 | 0.000 | 0.047 |
| 50 | 350 | 0.70 | 8.1 | 13.7 | 3.05 | 3.05 | 3.03 | 3.05 | 3.05 | 3.76 | 3.06 | 0.003 | 0.006 | 0.002 | 0.001 | 0.052 |
| 50 | 350 | 0.75 | 10.2 | 22.0 | 3.23 | 3.23 | 3.21 | 3.23 | 3.23 | 4.16 | 3.25 | 0.005 | 0.008 | 0.000 | 0.000 | 0.055 |
| 50 | 350 | 0.80 | 12.8 | 36.5 | 3.66 | 3.64 | 3.58 | 3.64 | 3.64 | 4.38 | 3.72 | 0.022 | 0.008 | 0.000 | 0.000 | 0.070 |
| 50 | 350 | 0.85 | 15.4 | 50.1 | 3.86 | 3.84 | 3.75 | 3.84 | 3.86 | 4.63 | 3.95 | 0.028 | 0.009 | 0.000 | 0.000 | 0.078 |
| 50 | 350 | 0.90 | 18.0 | 67.8 | 4.66 | 4.54 | 4.50 | 4.62 | 4.66 | 5.37 | 4.85 | 0.048 | 0.009 | 0.000 | 0.000 | 0.086 |
| 50 | 350 | 0.95 | 20.4 | 80.5 | 4.85 | 4.73 | 4.30 | 4.73 | 4.74 | 5.95 | 5.17 | 0.066 | 0.006 | 0.002 | 0.001 | 0.089 |
| 50 | 500 | 0.40 | 1.4 | 0.4 | 0.91 | 0.91 | 0.91 | 0.91 | 0.91 | 0.94 | 0.91 | 0.002 | 0.003 | 0.000 | 0.000 | 0.040 |
| 50 | 500 | 0.45 | 1.8 | 0.9 | 0.91 | 0.91 | 0.91 | 0.91 | 0.91 | 0.98 | 0.91 | 0.003 | 0.003 | 0.000 | 0.000 | 0.044 |
| 50 | 500 | 0.50 | 2.3 | 2.2 | 1.10 | 1.10 | 1.06 | 1.10 | 1.10 | 1.23 | 1.10 | 0.005 | 0.002 | 0.000 | 0.000 | 0.044 |
| 50 | 500 | 0.55 | 3.0 | 3.9 | 1.10 | 1.10 | 1.06 | 1.10 | 1.10 | 1.32 | 1.10 | 0.006 | 0.005 | 0.000 | 0.000 | 0.045 |
| 50 | 500 | 0.60 | 3.4 | 5.5 | 1.40 | 1.40 | 1.36 | 1.40 | 1.40 | 1.70 | 1.40 | 0.003 | 0.003 | 0.000 | 0.000 | 0.055 |
| 50 | 500 | 0.65 | 5.1 | 11.7 | 1.90 | 1.90 | 1.86 | 1.90 | 1.90 | 2.29 | 1.90 | 0.005 | 0.000 | 0.000 | 0.000 | 0.053 |
| 50 | 500 | 0.70 | 7.3 | 22.2 | 2.29 | 2.29 | 2.21 | 2.29 | 2.29 | 2.82 | 2.29 | 0.016 | 0.003 | 0.000 | 0.000 | 0.065 |
| 50 | 500 | 0.75 | 9.9 | 37.6 | 3.29 | 3.29 | 3.23 | 3.29 | 3.29 | 3.76 | 3.29 | 0.039 | 0.005 | 0.000 | 0.000 | 0.077 |
| 50 | 500 | 0.80 | 12.9 | 55.6 | 3.57 | 3.53 | 3.40 | 3.57 | 3.55 | 4.41 | 3.67 | 0.034 | 0.008 | 0.002 | 0.001 | 0.087 |
| 50 | 500 | 0.85 | 15.8 | 77.9 | 3.79 | 3.71 | 3.50 | 3.71 | 3.75 | 5.34 | 3.94 | 0.053 | 0.008 | 0.000 | 0.000 | 0.105 |
| 50 | 500 | 0.90 | 19.6 | 113.2 | 3.92 | 3.84 | 3.26 | 3.55 | 3.88 | 9.13 | 4.17 | 0.072 | 0.011 | 0.000 | 0.002 | 0.119 |
| 50 | 500 | 0.95 | 22.3 | 131.4 | 4.43 | 4.35 | 3.42 | 4.10 | 4.29 | 13.96 | 4.72 | 0.089 | 0.016 | 0.005 | 0.000 | 0.128 |

〈표 3〉 시험계산결과(노드 80개인 경우)

| 원네트워크 | | | 사전처리후 | | 하 한 값 | | | | | 상 한 값 | | 실 행 시 간 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| \|V\| | \|E\| | b/c(E) | \|V\| | \|E\| | Add1 | Add2 | Drop1 | Drop2 | Tabu | UB | UBV | LP1 | LP2 | Add1 | Drop1 | Tabu |
| 80 | 300 | 0.40 | 8.0 | 8.6 | 2.79 | 2.77 | 2.75 | 2.77 | 2.79 | 3.30 | 2.84 | 0.001 | 0.003 | 0.000 | 0.000 | 0.139 |
| 80 | 300 | 0.45 | 10.4 | 12.5 | 3.11 | 3.10 | 3.05 | 3.11 | 3.11 | 3.61 | 3.25 | 0.008 | 0.003 | 0.002 | 0.000 | 0.150 |
| 80 | 300 | 0.50 | 13.8 | 18.0 | 3.46 | 3.33 | 3.17 | 3.42 | 3.50 | 3.92 | 3.76 | 0.003 | 0.003 | 0.002 | 0.000 | 0.158 |
| 80 | 300 | 0.55 | 15.2 | 20.1 | 3.85 | 3.73 | 3.52 | 3.73 | 3.88 | 4.31 | 4.07 | 0.003 | 0.003 | 0.002 | 0.000 | 0.172 |
| 80 | 300 | 0.60 | 18.0 | 25.9 | 4.11 | 3.88 | 3.33 | 3.76 | 4.11 | 4.60 | 4.38 | 0.006 | 0.006 | 0.000 | 0.000 | 0.184 |
| 80 | 300 | 0.65 | 20.7 | 32.5 | 4.43 | 4.34 | 4.04 | 4.25 | 4.43 | 4.86 | 4.66 | 0.011 | 0.005 | 0.000 | 0.000 | 0.192 |
| 80 | 300 | 0.70 | 23.6 | 41.4 | 4.73 | 4.48 | 4.06 | 4.54 | 4.73 | 5.17 | 4.99 | 0.017 | 0.005 | 0.000 | 0.000 | 0.210 |
| 80 | 300 | 0.75 | 26.3 | 48.3 | 4.98 | 4.53 | 4.52 | 4.84 | 4.98 | 5.48 | 5.44 | 0.019 | 0.005 | 0.005 | 0.000 | 0.220 |
| 80 | 300 | 0.80 | 29.0 | 57.5 | 5.22 | 4.87 | 4.43 | 5.08 | 5.23 | 5.75 | 5.68 | 0.032 | 0.002 | 0.005 | 0.000 | 0.234 |
| 80 | 300 | 0.85 | 32.3 | 68.8 | 5.53 | 5.05 | 4.86 | 5.27 | 5.49 | 6.00 | 5.96 | 0.045 | 0.002 | 0.005 | 0.002 | 0.250 |
| 80 | 300 | 0.90 | 35.8 | 82.4 | 5.71 | 5.20 | 4.76 | 5.40 | 5.78 | 6.33 | 6.33 | 0.078 | 0.006 | 0.005 | 0.002 | 0.270 |
| 80 | 300 | 0.95 | 41.3 | 110.0 | 15.17 | 14.60 | 14.02 | 14.78 | 15.20 | 15.84 | 15.80 | 0.120 | 0.009 | 0.006 | 0.002 | 0.305 |
| 80 | 500 | 0.40 | 3.0 | 2.3 | 1.88 | 1.88 | 1.88 | 1.88 | 1.88 | 2.03 | 1.88 | 0.008 | 0.003 | 0.000 | 0.000 | 0.161 |
| 80 | 500 | 0.45 | 4.8 | 4.5 | 1.96 | 1.96 | 1.96 | 1.96 | 1.96 | 2.37 | 1.96 | 0.006 | 0.001 | 0.000 | 0.000 | 0.178 |
| 80 | 500 | 0.50 | 6.8 | 7.6 | 2.13 | 2.13 | 2.05 | 2.13 | 2.13 | 2.72 | 2.19 | 0.006 | 0.000 | 0.000 | 0.000 | 0.188 |
| 80 | 500 | 0.55 | 8.7 | 11.1 | 2.35 | 2.35 | 2.11 | 2.35 | 2.35 | 2.98 | 2.42 | 0.006 | 0.003 | 0.000 | 0.000 | 0.197 |
| 80 | 500 | 0.60 | 11.6 | 17.2 | 2.47 | 2.41 | 2.15 | 2.41 | 2.47 | 3.22 | 2.52 | 0.003 | 0.005 | 0.000 | 0.000 | 0.213 |
| 80 | 500 | 0.65 | 14.5 | 26.0 | 2.65 | 2.47 | 2.27 | 2.58 | 2.65 | 3.51 | 2.71 | 0.005 | 0.006 | 0.002 | 0.000 | 0.231 |
| 80 | 500 | 0.70 | 19.0 | 38.2 | 3.11 | 3.11 | 2.97 | 3.11 | 3.08 | 3.74 | 3.15 | 0.016 | 0.003 | 0.000 | 0.000 | 0.262 |
| 80 | 500 | 0.75 | 23.7 | 57.0 | 3.17 | 3.01 | 3.06 | 3.08 | 3.17 | 3.97 | 3.30 | 0.023 | 0.006 | 0.002 | 0.000 | 0.302 |
| 80 | 500 | 0.80 | 28.2 | 74.9 | 3.59 | 3.37 | 3.30 | 3.38 | 3.59 | 4.19 | 3.81 | 0.041 | 0.010 | 0.000 | 0.002 | 0.328 |
| 80 | 500 | 0.85 | 32.4 | 92.5 | 3.91 | 3.69 | 3.63 | 3.69 | 3.91 | 4.39 | 4.17 | 0.069 | 0.010 | 0.006 | 0.000 | 0.358 |
| 80 | 500 | 0.90 | 37.1 | 121.1 | 4.19 | 4.10 | 3.90 | 4.10 | 4.19 | 4.63 | 4.45 | 0.180 | 0.014 | 0.003 | 0.003 | 0.406 |
| 80 | 500 | 0.95 | 42.5 | 157.0 | 4.35 | 4.21 | 3.86 | 4.26 | 4.35 | 4.92 | 4.70 | 0.370 | 0.017 | 0.009 | 0.001 | 0.450 |
| 80 | 700 | 0.40 | 2.4 | 1.6 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.89 | 0.70 | 0.002 | 0.002 | 0.000 | 0.000 | 0.171 |
| 80 | 700 | 0.45 | 3.8 | 3.8 | 1.38 | 1.38 | 1.38 | 1.38 | 1.38 | 1.47 | 1.38 | 0.002 | 0.003 | 0.000 | 0.000 | 0.178 |
| 80 | 700 | 0.50 | 5.6 | 8.6 | 1.53 | 1.53 | 1.53 | 1.53 | 1.53 | 1.69 | 1.58 | 0.006 | 0.002 | 0.000 | 0.000 | 0.195 |
| 80 | 700 | 0.55 | 7.1 | 13.5 | 1.74 | 1.74 | 1.71 | 1.74 | 1.71 | 1.97 | 1.82 | 0.006 | 0.005 | 0.000 | 0.000 | 0.205 |
| 80 | 700 | 0.60 | 10.8 | 27.1 | 1.99 | 1.97 | 1.91 | 1.97 | 1.96 | 2.39 | 2.08 | 0.016 | 0.004 | 0.000 | 0.002 | 0.237 |
| 80 | 700 | 0.65 | 14.1 | 42.3 | 2.19 | 2.17 | 2.11 | 2.17 | 2.16 | 2.75 | 2.32 | 0.034 | 0.008 | 0.000 | 0.002 | 0.272 |
| 80 | 700 | 0.70 | 17.9 | 64.3 | 2.58 | 2.58 | 2.52 | 2.58 | 2.58 | 3.13 | 2.60 | 0.422 | 0.013 | 0.003 | 0.000 | 0.312 |
| 80 | 700 | 0.75 | 22.2 | 91.3 | 2.78 | 2.78 | 2.59 | 2.78 | 2.78 | 3.45 | 2.82 | 0.344 | 0.014 | 0.002 | 0.000 | 0.356 |
| 80 | 700 | 0.80 | 25.8 | 121.7 | 2.90 | 2.85 | 2.70 | 2.85 | 2.86 | 3.89 | 2.94 | 0.334 | 0.011 | 0.003 | 0.000 | 0.399 |
| 80 | 700 | 0.85 | 29.4 | 145.9 | 2.95 | 2.90 | 2.58 | 2.73 | 2.94 | 4.45 | 3.10 | 0.403 | 0.019 | 0.002 | 0.003 | 0.439 |
| 80 | 700 | 0.90 | 35.3 | 189.2 | 3.03 | 2.89 | 2.52 | 2.66 | 3.00 | 5.87 | 3.41 | 0.381 | 0.022 | 0.006 | 0.001 | 0.500 |
| 80 | 700 | 0.95 | 39.5 | 223.7 | 3.51 | 3.40 | 3.09 | 3.17 | 3.51 | 7.29 | 3.88 | 0.533 | 0.027 | 0.006 | 0.005 | 0.548 |

을 볼 때, 제시된 상한과 하한을 구하는 해법이 큰 규모의 문제를 효과적으로 풀 수 있다고 판단된다.

## 참 고 문 헌

[1] Cosares, S., Deutch, N.D., Saniee, I. and Wasem, O.J., "SONET toolkit : A decision support system for designing robust and cost-effective fiber-optic networks," *Interfaces*, Vol.25(1995), pp.20-40.

[2] Cunningham, W.H., "Optimal attack and reinforcement of a network," *Journal of the ACM*, Vol.32(1985), pp.549-561.

[3] Glover, F., "Future Paths for Integer Programming and Links to Artificial Intelligence," *Comput. Operations Res*, Vol.5(1986),

pp.533-540.

[4] Glover, F., "Tabu Search-Part I," *ORSA J. Computing*, Vol.1(1989), pp.190-206.

[5] Glover, F., "Tabu Search-Part II," *ORSA J. Computing*, Vol.2(1990), pp.4-32.

[6] Grötschel, M., Monma, C.L. and Stoer, M., "Design of survivable networks," *Network Models*, M.O. Ball et al. (eds.), North-Holland, Amsterdam, 1995.

[7] Gusfield, D., "Computing the strength of a graph," *SIAM Journal on Computing*, Vol. 20(1991), pp.639-654.

[8] Kim, Y-K., Yun, B-S. and Lee, S-B., *Meta Heuristic*, Yeongji Moonhwasa, Seoul, Korea, 1999.

[9] Martel, C., Nuckolls, G. and Sniegowski, D., "Computing the disconnectivity of a graph," Working paper, UC Davis, 2001.

[10] Myung, Y.-S. and Kim, H.-J., "A Cutting Plane Algorithm for Computing $k$-edge Survivability of a Network," To appear in *European Journal of Operational Research*, 2001.

[11] Myung, Y.-S. and Kim, H.-J., "An algorithm for the graph disconnection problem," Working paper, Dankook University, 2003a.

[12] Myung, Y.-S. and Kim, H.-J., "An improved algorithm for the graph disconnection problem," Proceedings of KIIE/KORMS Spring Conference, 2003b.

[13] Myung, Y.-S., Kim, H.-J. and Tcha, D.-W., "Design of communication networks with survivability constraints," *Management Science*, Vol.45(1999), pp.238-252.

[14] Oh, S-M., "Mathematical Models and a Heuristic Algorithm for the Graph Disconnection Problem," *Master Dissertation, Korea Advanced Institute of Science and Technology*, 2003.

[15] Wu, T., *Fiber network survivability*, Artech House, Boston, 1992.