대한산업공학회/한국경영과학회 2002 춘계공동학술대회
한국과학기술원(KAIST) 2002년 5월 3일-4일

네트워크 단절문제에 대한 해법

명 영 수
단국대학교 천안캠퍼스 경상학부

김현준
울산대학교 경영정보학과

초록: 네트워크에 속한 각 에지(edge)를 제거하는데 드는 비용 및 노드(node)들의 가중치가 주어져 있다고 가정하자. 네트워크 단절문제는 주어진 네트워크에서 원천노드(source node)와 연결이 끊어지는 노드들의 가중치의 합이 최대가 되도록 에지를 제거하는 방법을 찾는 문제이다. 이 때, 제거된 에지의 비용의 합은 정해진 예산의 범위를 초과해서는 안 된다. 이 문제는 Martel 등에 의해서 응용사례와 함께 처음 소개되었고, NP-hard에 속한 문제임이 입증되었다. 본 연구에서는 주어진 그래프의 특성을 이용하여 실행가능해를 도출하는 휴리스틱과 문제의 크기를 줄이는 과정 및, 대상문제의 수리모형을 이용한 상한의 계산과정을 개발하기로 한다.

1. 서론

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 원천노드와 연결이 끊어지는 노드들의 가중치의 합이 최대가 되도록 에지를 제거하는 것이다. 이 때, 제거된 에지의 비용의 합은 주어진 예산의 범위를 넘어서는 안 된다.

네트워크 단절문제는 네트워크의 노드들과 원천노드와의 연결을 단절시키려는 네트워크에 대한 공격을 모델로 하여, Martel 등 [2]에 의해서 처음 제시되었다. 저자들은 에지를 제거하는데 드는 비용과 노드별 가중치가 모두 1의 값을 갖더라도, 네트워크 단절문제는 NP-hard 에 속한 어려운 문제임을 밝히고, 후보가 되는 에지의 집합을 일일이 헤아리는 방법을 제시하였다. 물론 문제의 복잡성에서 알 수 있듯이 이 방법은 다항시간(polynomial time)안에 완료되지는 않는다.

Martel 등 [2]은 네트워크 단절문제가 전자상거래를 위한 전산망 및 분산처리를 위한 전산망 등에서 응용될 수 있음을 밝혔다. 실제로 이 문제는 공격자의 입장에서 정의되어 있으나, 이를 이용하여 생존도(survivability)가 높은 네트워크를 구성하는 방법으로 활용할 수 있다. 실제로 통신 서비스에의 의존도가 늘어가고 있는 오늘날의 사회에서는 비용 면에서 효율적이면서도 일부 구성요소의 장애에도 지속적인 서비스의 제공이 가능한 안정적인 통신망의 구축이 필요하며, 통신망 생존도의 연구는 이를 위한 필수불가결의 요소이다 [1,3].

본 연구에서는 Martel 등 [2]이 제시한 일일이 헤아리는 방법보다는 좀 더 현실적인 방법으로 네트워크 단절문제를 풀 수 있는 해법을 개발하기로 한다. 네트워크의 단절문제는 주어진 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.

앞서 언급한대로 대상 문제는 NP-hard에 속하는 문제이므로, 다항시간 안에 이 문제를 풀 수 있는 해법을 기대하기는 어렵다. 따라서 바람직한 접근방법은 원 문제의 최적해에 가까운 실행가능해(feasible solution)를 구하는 것이며 특히 원 문제의 목적함수 값에 대한 상한(upper bound)을 구할 수 있다면 구해진 해의 품질을 평가하는데 이용할 수 있다. 또한 분지해법(branch and bound method)을 이용하여 최적해를 구하고 싶은 경우에도 하한(lower bound)을 제공하는 실행가능해와 상한은 요긴하게 사용된다. 본 연구에서는 네트워크 단절문제의 실행가능해와 상한을 구하는 방법을 제시하기로 한다. 실행가능해를 구하는 과정에서 얻은 정보를 이용하여 주어진 그래프의 크기를 줄이는 사전처리(preprocessing) 방법도 제시한다. 또한 네트워크 단절문제를 풀기 위한 수리적인 모형을 제시하고 주어진 모형을 이용하여 상한을 도출하는 방법을 개발하기로 한다.

2. 용어의 정의

주어진 네트워크에서 노드의 집합을 $V=\{1,\ldots,n\}$, 무방향 에지의 집합을 $E=\{1,\ldots,m\}$ 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 i와 j를 에지 e의 종단노드(end nodes)라고 부르기로 한다. 각 에지 $e\in E$ 마다 에지를 제거하는데 드는 비용을 c_e, 에지를 제거하는데 사용할 수 있는 예산을 b, 각 노드 $v\in V_1$ 의 가중치를 w_v로 표시하기로 한다. 주어진 그래프 G에는 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.

원천노드를 제외한 노드들의 집합을 표시하기 위하여 $V_1=V\setminus\{1\}$을 사용한다. V_1의 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 S의 부분집합 S'을 분리시키는 컷이므로 $1-S'$컷이라고 부르기로 한다. 간단한 표현을 위해서, 만약에 $S'=\{i\}$이면 $\delta(S)$는 $1-\{i\}$컷 대신에 $1-i$컷으로 표기하기로 한다. 임의의 컷 $\delta(S)$에 대해서 컷을 이루는 에지에 대한 비용의 합, 즉 $\sum_{e\in\delta(S)} c_e$를 컷의 비용이라고 부르기로 한다. V_1의 임의의 부분집합 S 에 대한 $1-S$컷 중에서 컷의 비용이 최소인 컷을 최소비용 $1-S$컷이라고 정의한다.

V_1의 임의의 부분집합 S 에 대해서 $\delta(S)$가 $\sum_{e\in\delta(S)} c_e\le b$를 만족하면, S 를 분리가능-노드집합이라고 부르기로 한다. 따라서 네트워크 단절문제는, 그래프 $G=(V,E)$와 c_e, b, w_v 등이 주어져 있을 때, 가중치의 합, 즉, $\sum_{v\in S} w_v$가 최대가 되는 분리가능-노드집합 S를 구하는 문제로 정의할 수 있다.

3. 실행가능해의 도출 및 사전처리

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. 본 연구에서는 현실적인 효율성을 감안하여 적절한 시간 내에 최적해의 상한과 하한을 구하는 방법을 제시하고자 한다. 이 절에서는 실행가능해, 즉 분리가능-노드집합을 선택하는 휴리스틱을 제시하고 이 과정에서 얻어지는 정보를 통해서 주어진 그래프의 크기를 줄일 수 있는 사전처리과정을 개발하기로 한다.

우리의 휴리스틱은 다음과 같은 사실을 기반으로 구성되었다.

관찰 1. V_1의 임의의 부분집합 S 에 대해서 최소비용 $1-S$컷은 복수로 존재할 수 있다. 만약에 $\delta(S_1)$과 $\delta(S_2)$가 최소비용 $1-S$컷이라고 하면 $\delta(S_1\cup S_2)$도 최소비용 $1-S$컷이 된다. 따라서, 모든 최소비용 $1-S$컷을 $\delta(S_1),\cdots,\delta(S_t)$라고 하면, $\delta(\bigcup_{i=1}^{t} S_i)$도 최소비용 $1-S$컷이며, 이러한 컷은 유일하게 존재하는데 이를 최대규모의 최소비용 $1-S$컷이라고 부르기로 한다.

관찰 2. V_1의 임의의 부분집합 S 가 S의 임의의 부분집합 S'에 대한 최대규모의 최소비용 $1-S'$컷이라고 가정하자. 그러면, $\delta(S)$는 S'를 포함하는 어떠한 S의 부분집합 S''에 대해서도 최대규모의 최소비용 $1-S''$컷을 이룬다.

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

관찰 1과 2는 컷의 서브-모듈라 특성(Submodularity)으로부터 쉽게 알 수 있고, 관찰 3은 최적해와 최대규모의 최소비용 컷의 정의로부터 자명하게 성립한다. 또한 최대규모의 최소비용 컷은 최대흐름문제(maximum flow problem)를 이용해서 구할 수 있다. 관찰 2와 3은, 네트워크 단절문제의 최적해를 구성하는 분리가능-노드집합을 탐색할 때 고려하여야 할 후보집합의 대상을 줄일 수 있게 하여준다. 이러한 아이디어는 Martel [2]의 방법에도 간접적으로 활용되었다.

우리의 휴리스틱은 최적해의 보장은 없으나 가능한 가중치의 합이 큰 분리가능-노드집합을 구하는 것이다. 우리의 해법은 V_1에 속

한 노드 중에서 분리가능-노드집합에 포함될 노드를 추가해 가는 방식의 휴리스틱(add heuristic)이다. 특징적인 것은 관찰 2의 사실을 이용하여 분리가능-노드집합에 포함될 노드를 하나씩 선택하는 대신에, 현재까지 선택된 집합에 포함되어 있지 않은 노드 i를 추가할 때에는 최대규모의 최소비용 $1-i$컷의 i쪽에 포함된 모든 노드를 선택한다는 것이다. 노드를 어떤 순서로 선택하느냐에 따라서 얻어지는 해가 달라지게 된다. 본 연구에서는 우선 V_1에 속한 각각의 노드 i에 대해서 최대규모의 최소비용 $1-i$컷, $\delta(S_i)$를 구한 다음에, S_i를 추가할 때 발생하는 에지비용의 증가와 가중치의 증가를 비교하여 선택의 순서를 결정하였다. 그리고 이 과정에서 얻어진 $1-i$컷의 비용을 이용하여 주어진 그래프의 크기를 줄일 수 있는지 검토한다. 우리의 휴리스틱은 다음과 같이 진행된다.

실행가능해의 도출과 사전처리를 위한 알고리즘

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

Output: **분리가능-노드집합** S

(단계 1) 모든 $i\in V_1$**에 대해서 최대규모의 최소비용** $1-i$**컷,** $\delta(S_i)$**를 구한다.**

(단계 2) (사전처리과정) S_i**가 분리가능-노드집합이 아니면 노드 1과 노드** i**를 축약시킨다.**

(단계 3) $S=\emptyset$**으로 초기화하고,**
$I=\{i\in V_1-S \mid S\cup S_i$**는 분리가능-노드집합**$\}$
가 공집합이 아닐 때까지 다음을 반복:

$\max_{i\in I}\left(\frac{\sum_{v\in S_i-S} w_v}{\sum_{e\in\delta(S_i\cup S)-\delta(S)} c_e}\right)$**가 되는** i**를 선택하여** $S=S\cup S_i$**로 놓는다.**

단계 2에서의 사전처리는 다음과 같은 사실에 근거를 두고 있다. 어떤 노드 i에 대해서 최소비용 $1-i$컷의 비용이 예산을 초과한다면 최적해가 되는 분리가능-노드집합은 노드 i를 포함할 수 없다. 따라서 원래의 그래프에 대한 최적해와 노드 1과 i를 축약한 그래프에서의 최적해는 동일하게 될 것이다. 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 문제해결에 소요되는 시간을 줄일 수 있다.

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.

4. 상한의 도출

본 절에서는 네트워크 단절문제의 상한을 구하는 방법을 고려해 보기로 한다. 이를 위해서, 대상문제의 정수계획모형을 수립하고 선형계획 완화문제를 푸는 해법을 개발하기로 한다.

특정의 에지 e의 제거 여부를 나타내는 변수 x_e와 노드 i가 1에서 분리되는지 여부를 나타내는 변수 y_i를 다음과 같이 정의하자.

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

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

그러면 네트워크 단절문제는 다음과 같은 정수계획모형 (IP)로 나타낼 수 있다.

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

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

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

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

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

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

모형 (IP)의 최적해는 네트워크 단절문제의 최적해가 된다. 따라서, 모형 (IP)에서 제약식 (4)와 (5)를 각각 $0 \le x_e \le 1, \forall e \in E$, $0 \le y_i \le 1, \forall i \in V_1$로 대체함으로써 얻어지는 ($IP$)의 선형계획 완화문제(linear programming relaxation problem) - 이를 (LP)라 하자 - 는 원문제의 최적해의 목적함수값에 대한 상한(upper bound)을 제공한다. 그러나 제약식 (3)의 수가 너무 많기 때문에, 모든 제약식을 전부 포함한 선형계획문제를 풀어서 상한을 구하기에는 현실적인 무리가 따른다.

본 연구에서는 제약식 (3) 중 필요한 식만 선택적으로 포함하는 선형계획문제의 해결방법을 사용한다. 이 방법은 우선 제약식 (3) 중 일부의 식만을 포함한 선형계획문제를 풀고, 얻어진 해가 (LP)의 최적해인지를 확인한 다음, 필요하면 다시 제약식을 추가하여 반복하는 단계적 절차로 구성되어 있다. 반복의 단계마다 제약식의 추가에 의해서 실행가능영역(feasible region)을 줄여 가는 이러한 방법을 절단면해법(cutting plane algorithm)이라고 부른다. 우리의 절단면해법에서 중요한 것은 현재의 선형계획문제의 해가 모든 제약식 (3)을 만족하는지를 판정하고, 아닌 경우에 현재의 해가 만족시키지 못하는 제약식이 어떤 것인지를 결정하는 분리문제(separation problem)를 쉽게 풀 수 있는가의 여부이다. 절단면해법과 분리문제에 대한 자세한 설명은 Nemhauser와 Wolsey [4]를 참조하기 바란다. 제약식 (3)에 해당하는 분리문제는 아래의 절차 단계 ②에서 설명하는 것과 같이 최단경로문제(shortest path problem)를 풀어서 해결할 수 있다.

상한 계산절차

① 목적식과 제약식 (2), $0 \le x_e \le 1$, $\forall e \in E$, $0 \le y_i \le 1, \forall i \in V_1$만을 포함하는 선형계획문제를 구성한다.
② 최적해 $\{x_e, y_i\}$를 구하여, x_e를 에지 e의 길이로 하는 그래프에서 노드 1로부터 각 노드 $i \in V_1$까지의 최단경로를 구하고 그 길이 l_i를 구하여, 만일 $l_i < y_i$이면 이 최단경로 p에 대해서 제약식(3)을 추가한다.
③ 추가될 제약식이 없으면 현재의 해가 최적이므로 종료하고, 추가된 제약식이 있으면 추가된 제약식을 포함하여 선형계획문제를 푼 뒤에 다시 ②로 간다.

제약식 (3)에 해당하는 분리문제의 근거는 다음과 같다. x_e를 링크 길이로 하는 그래프에서, 두 노드 1과 i사이의 최단 경로의 길이가 y_i이상이면 1과 i사이의 모든 경로의 길이가 y_i이상이므로 현재의 선형계획문제의 해 $\{x_e, y_i\}$는 노드 i에 대한 모든 제약식 (3)을 만족하게 되고, 만약 어떤 경로 p에 대한 경로의 길이가 y_i미만이면 $\{x_e, y_i\}$는 해당 경로에 대해서 (3)을 만족시키지 못하게 된다. 이 과정을 반복하여 (LP)에 대한 최적해를 얻게되면, 이 때의 목적함수의 값이 네트워크 단절문제의 상한이 된다. 이와 같이 제약식 (3)에 대한 분리문제를 최단경로문제의 해법을 이용하여 다항시간(polynomial time)에 풀 수 있음으로써 선형계획완화문제 (LP)도 다항시간에 풀 수 있음을 알 수 있다.

5. 계산실험 및 결과 분석

본 연구에서 제시된 네트워크 단절문제의 상한과 하한을 구하기 위한 절차, 그리고 문제의 단순화 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 50개까지의 노드로 구성되는 가상의 문제를 만들었다. 가상의 문제를 만들기 위해서는 먼저 (100×100) 사각형 모양의 격자에 필요한 수만큼의 노드를 임의로 위치시키고, 다음에 노드간에 적정수의 링크설비를 위치시켰다. 우선 노드들이 상호 연결되도록 하나의 트리로 연결한 후, 지정된 수만큼의 추가적인 링크를 설치하였다. 노드의 가중치와 에지의 제거비용은 일정한 범위 내에서 임의추출방식(random sampling)으로 생성하였으며, 에지 제거를 위한 예산은 에지비용의 합의 일정한 비율로 설정하였다.

작성된 코드는 150Mhz CPU를 가진 펜티엄급 PC에서 실행되어 졌다. 현재 다양한 크기의 네트워크에서 서로 다른 비용 및 가중치를 갖는 많은 문제에 대한 시험계산이 이루어지고 있는 중이나 여기에서는 일부 결과만을 소개하기로 한다. <표 1>에는 20문제에 대한 결과가 나타나 있다. 문제별로 주어진 그래프의 크기를 노드의 수와 에지의 수로 표시하였고, 예산의 경우는 모든 에지에 대한 에지제거비용의 합계에 대한 비율로 표시하였다. 사전처리과정의 효율을 보여주기 위하여, 사전처리가 이루어진 이후 그래프의 크기를 대비하였고 소요된 시간을 표시하였다. 예상할 수 있는 것처럼 예산 규모가 커질수록 사전처리의 효과는 줄어들고 있음을 알 수 있다. 분석이 용

[표 1] 계산결과

문제크기 및 예산			사전처리		실행가능해 (%)	상한 (%)	CPU	상한 − 하한 / 하한
|V|	|E|	%	|V|	|E|				
30	50	1	5	4	8.6	9.1	0.06	0.06
		2	8	8	12.0	13.1	0.05	0.09
		4	15	18	25.7	30.0	0.05	0.17
		6	18	23	31.0	33.9	0.06	0.09
		8	23	33	34.3	39.1	0.05	0.14
30	60	1	3	2	2.1	2.5	0.05	0.19
		2	3	2	3.3	3.3	0.06	0.00
		4	10	12	3.3	3.3	0.05	0.00
		6	17	28	12.2	12.2	0.06	0.00
		8	25	48	25.8	25.8	0.16	0.00
50	70	1	10	9	4.1	4.5	0.06	0.10
		2	20	22	7.4	8.9	0.11	0.20
		4	39	55	16.4	20.6	0.27	0.26
		6	49	67	96.3	98.0	0.22	0.02
		8	50	70	99.0	99.4	0.22	0.00
50	80	1	15	14	7.3	8.0	0.06	0.10
		2	20	21	11.8	13.5	0.11	0.14
		4	32	42	19.6	21.6	0.07	0.10
		6	50	80	84.8	87.5	0.55	0.03
		8	50	80	87.6	98.1	0.33	0.12

이하도록 실행가능해와 상한을 총 가중치에 대한 비율로 표현하였다.

제시되는 하한 및 상한의 성능을 기준으로 할 때, 현재의 방법들이 큰 규모의 망에서도 대체로 만족스럽다고 평가할 수 있으나 좀 더 나은 상한과 하한을 얻기 위해서는 보다 세밀한 형태의 휴리스틱의 개발이나, 유효부등식(valid inequalities)를 이용한 선형계획 완화문제의 상한의 향상 등을 고려해 볼 수도 있을 것으로 판단된다.

<참고문헌>

[1] 명영수, 김현준 "흐름량을 고려한 네트워크 생존도 계산방법에 관한 연구", 「한국경영과학회지」, 제26권 3호 (2001), pp.65-78.

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

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

[4] Nemhauser, G.L. and L.A. Wolsey, *Integer and Combinatorial Optimization,* Wiley, New York 1988.