

## $k$ -링크생존도 산출을 위한 분지한계절차의 개발

김현준

울산대학교 경영학부 경영정보학전공

### <요 약>

네트워크에 주어지는 서비스 요구에 기반을 둔 정량적 생존도 지표로서  $k$ -링크생존도는  $k$ 개의 링크에 동시장애가 발생하는 경우에도 여전히 처리되어지는 서비스 요구의 비율로서 정의되며, 그 값을 정확히 계산하는 문제는 이미  $NP$ -hard 클래스에 속하는 문제임이 밝혀져 있다. 본 연구에서는  $k$ -링크생존도 산출을 위한 분지한계절차(branch and bound procedure)를 구현하였다. 기존의  $k$ -링크생존도 산출을 위한 정수계획모형을 바탕으로 추가적인 유효부등식을 도입하고 선형계획 완화문제의 특성을 분석하여, 문제 단순화, 상한값 및 하한값 산출절차를 통합하였으며, 서울지역 전화망을 기반으로 하는 문제와 임의추출방식으로 만들어진 문제를 대상으로 이루어진 실험계산 결과를 제시하였다. 결과분석을 통하여 기 개발되었던 선형계획 완화모형이나 유효부등식, 그리고 상한값 및 하한값을 구하는 휴리스틱 절차의 효과를 분석할 수 있었으며, 적절한 규모까지의 네트워크에 대해서 분지한계절차를 통한 링크생존도 산출방법이 효과적으로 사용될 수 있음을 확인하였다.

## A Branch & Bound Procedure for the $k$ -Link Survivability

Kim, Hyun-joon

Dept. of Management Information Systems, University of Ulsan

### <Abstract>

The flow-based  $k$ -link survivability of a network is defined as the percentage of total traffic requirements surviving the worst case failures of  $k$  links and the problem

of computing  $k$ -link survivability is known to be  $NP$ -hard. This paper implements a Branch and Bound procedure for the exact value of the  $k$ -link survivability. Based on the Integer Programming model for the network survivability, an additional valid inequality is introduced to reinforce the Linear Programming relaxation problem. A preprocessing process and upper/lower bounding procedures for the LP relaxation are integrated into a Branch and Bound procedure. Computational results for evaluating the performance of proposed procedures are also presented. The results show that the valid inequality and bounding procedures incorporated are so efficient and effective that they are applicable to large-size real world problems.

## I. 서론 및 연구의 배경

1980년대 이후 정보하부구조로서의 통신망은 전송 및 교환의 측면에서 많은 변화를 겪어 왔다. 기술적 변화의 기반으로 정보의 디지털화와 컴퓨터/통신기술의 결합, 광섬유를 이용한 광전송 방식의 개발에 따른 대용량/초고속 전송 기반의 확보 등은 단순히 통신인프라를 확충시켰을 뿐만 아니라 사회의 정보기반을 바꾸어 가고 있다. 공간적 격리를 극복하여 정보의 신속하고 정확한 전달을 가능하게 하였고, 이를 기반으로 사회부문간의 상호작용은 매우 긴밀하게 이루어지고 있다. 하드웨어적인 정보전달 기반 위에 인터넷을 통한 개방형 정보공유가 가능하게 됨으로써 사회의 정보통신 의존도는 급속히 심화되고 있다. 인터넷을 이용한 카페, 동호회 등의 가상공간 뿐만 아니라, 경제주체들간의 거래를 대체/보완하는 전자상거래까지 폭넓은 사회의 정보화는 정보통신망의 중요성을 배가시키고 있는 것이다.

한편으로 빨라진 정보전송 가능성은 통신망의 구조를 변화시키고 있다. 제한된 전송용량을 가지는 전송매체와 상대적으로 큰 전송매체비용의 비중으로 인하여 복잡한 그물모양을 유지하던 통신망 구조가 높은 전송속도와 무제한적인 전송용량, 전송매체비용을 압도하는 설비설치비용 등의 영향으로 보다 단순화된 구조로 바뀌어 가는 것이다. 파장분할 다중화 등의 최신 광통신 기술에 의하여 전송매체의 단위용량이 수 Gbps에 이르는 것은 이러한 기술발전 흐름을 단적으로 보여주는 것이라 하겠다[Wu, 1992].

이러한 망 구조의 단순화는 사회 속의 통신 기능 비중의 증가와 더불어 통신망의 성능평가 기준 변화로 이어지고 있다. 통신에 의한 연결이 가능한가? 얼마나 기다려야 하는가? 언제라도 사용할 수 있는가? 등의 전통적 기준으로부터 언제라도 서비스는 보장되는가? 만일의 경우에도 서비스는 유지되는가? 등의 새로운 기준으로 변화되고 있다. 이러한 변화흐름 속에서 1980년대 후반 이후 통신망의 생존도(network survivability)는 많은 학자들과 통신업계의 관심을 끌어 왔고 지금도 여전히 연구의 중심 대상이 되고 있다[Cosares et al., 1995; Grötschel et al., 1995; Myung et al., 1999].

흔히 생존도는 '통신망을 구성하는 일부 구성요소에 기능장애가 발생하는 경우에 부과되는 서비스가 어느 정도 처리될 수 있는가?' 라는 관점에서 정의된다. 우선 전통적인 그래프 이론에서 널리 알려진 노드간의 연결도(connectivity)나 네트워크의 구조적 중복성의 개념을 이용한 정성적인 생존도가 사용되었으며[Monma and Shallcross, 1988], 이후 네트워

크에 부과되는 서비스 양에 기초하여 구성요소 장애 시에도 여전히 처리되는 서비스 양의 비율로 표현되는 정량적 생존도 등이 제시되고 활용되어 왔다. 이는 적정의 통신망 구성요소 장애를 전제로 하는 한계적인 상황 아래에서 통신망의 서비스 제공 정도를 측정하는 지표이며, 장애가 발생하는 구성요소에 따라 링크생존도 또는 노드생존도로 구분된다. 본 연구에서는 링크생존도의 산출을 대상으로 한다. (노드생존도의 산출도 유사한 방법을 통하여 분석이 가능하다). 1980년대 후반 이후, 연결도에 근거한 통신망의 구조적 특성 분석 [Monma et al., 1990] 이나 생존도가 강화된 네트워크의 구성 방법 [Myung et al., 1999] 등의 연구들이 있었으며, 정량적인 생존도 지표 및 측정방법의 제시 [Cardwell et al., 1989; Kim, 1995], 생존도 지표를 반영하는 네트워크 구성방법, 관련 수리모형의 분석 [Grötschel et al., 1995, Myung et al., 1999] 등도 활발히 이루어지고 있다.

특히, 정량적 생존도의 산출에 관하여서는 1990년대 이후 많은 진전이 있었다고 할 수 있다. 발생 가능한 상황을 나열하여 분석하는 열거적 방법 [Wu et al., 1988], 두 개의 구성요소 장애만을 반영하는 싸이클 분해법 [Kim, 1995], 네트워크의 구조에 제한적인 특성이 존재함을 전제로 노드설비 축약을 활용하는 반복적 측정방안 [Myung et al., 1999]이 제시되었다. 이러한 연구들의 배경에는 이미 아무런 제한이 없는 일반적인 네트워크에서의 정량적 생존도 산출문제는  $NP$ -hard임이 전제되고 있었다. 이후, 정량적인 생존도의 근사적 값을 도출하는 연구의 흐름이 보다 구체화되어 네트워크에 부과되는 서비스 요구를 정량적으로 반영하는 흐름기반 생존도 산출을 위한 정수계획 모형이 제시되었고, 선형계획 완화모형을 이용한 하한값과 발견적 방법을 이용한 상한값 산출절차도 제시되었다 [Myung et al., 2001].

본 연구는 상기의 생존도 하한값 및 상한값 도출절차를 이용하여 정량적 생존도의 참값 (exact value) 도출과정을 구현하는데 목적이 있다. 물론  $NP$ -hard임이 확인된 문제에 대한 참값 산출절차를 제공하는 것이 현실적인 한계를 가지는 것이 당연하지만, 적정규모까지의 생존도 참값을 확보하는 것은 관련분야의 연구에 많은 도움을 줄 수 있다고 판단된다. 특히, 생존도에 대한 하한값과 상한값을 구하는 휴리스틱 절차의 효율성 분석, 각종 완화모형 및 다양한 유효부등식 (valid inequality)의 효과 분석 등에 폭넓게 적용될 수 있을 것이다.

우선 통신망의 링크생존도를 구하기 위하여 기존 연구에서 사용한 하한값 및 상한값을 구하는 절차를 포괄하는 분지한계절차를 구현하였다. 최악의 경우에는 열거적 방법에 준하는 시간이 소요될 수 있지만, 선형계획 완화기법과 유효 부등식을 이용한 하한값 절차, 휴리스틱에 의한 상한값, 선형계획 완화문제를 풀어 가는 중에 얻을 수 있는 적정 정수해의 값들을 폭넓게 활용함으로써, 효율성의 향상을 도모하고자 하였다. 정립된 분지한계절차의 효율성 검증을 위하여 폭넓은 시험계산을 실행하였다. 시험적용 결과를 통하여 정량적 생존도의 참값을 얻기 위한 분지한계절차가 적정규모의 현실 통신망 분석에 이용될 수 있음을 확인하였고, 더불어 기존에 제시된 하한값 및 상한값 도출절차들의 효율성 및 타당성도 부가적으로 확인할 수 있었다.

2장에서는 링크생존도의 수리적인 정의와 특성, 문제정의에 필요한 부호 등을 정리하며, 링크생존도의 일반 특성을 분석한다. 3장에서는 생존도 산출을 위한 정수계획 모형, 상한값 및 하한값을 얻기 위해서 사용되는 휴리스틱과 선형계획 완화모형을 제시한다. 또한 선형계획 완화문제를 강화시키기 위한 유효부등식을 포함하여 생존도 값을 얻기 위한 분지

한계절차를 설계하며, 분지변수의 선택기준, 하위문제 고려 순서 등을 구체화한다. 4장에서는 시험 계산과정과 임의추출방식으로 만들어진 가상의 문제 및 서술지역 전화망을 대상으로 만들어진 문제에서의 시험계산결과를 정리하며, 특히 기존연구 계산결과와의 상호비교를 통하여 휴리스틱 값이나 상한값, 하한값을 구하는 절차 등의 효율성을 분석한다.

## II. 용어정의 및 문제의 특성

교환설비는 노드(node), 전송링크설비는 아크(arc)로 표현함으로써 통신망은 그래프로 나타내어진다. 즉, 교환설비는 노드집합  $V = \{1, \dots, n\}$ 로 전송링크설비는 아크집합  $E = \{1, \dots, m\}$ 로 나타내면, 그래프  $G = (V, E)$ 로 통신망의 구조를 나타낼 수 있다. 한편, 노드설비  $i$ 와 노드설비  $j$ 간 서비스 요구의 양을  $t_{ij}$ 로 표시하고, 서비스 요구가 존재하는 노드쌍의 집합을  $T = \{t_{ij} | i, j \in V, t_{ij} > 0\}$ 로 표시하기로 하면, 그러면  $(G, T)$ 로서 서비스 요구를 가지는 통신망을 표현할 수 있다.

그래프  $G$ 에서 노드  $v$ 에 인접한 링크 수(degree)를  $\delta(v)$ 로 나타낸다. 링크의 축약(contraction)은 대상 링크를 삭제하고, 삭제되는 링크의 양쪽 종단노드를 하나의 노드로 합치는 것을 의미한다. 물론 두 노드중의 하나에 연결되어 있던 기존의 링크는 새로이 축약된 노드로 연결되게 되며, 두 노드 사이의 서비스 요구는 삭제된다.

노드 집합  $V$ 의 임의의 부분집합  $W$ 와  $V \setminus W$ 를 연결하는 링크집합을 컷이라 하며,  $\delta(W, V \setminus W)$  또는  $\delta(W)$ 로 나타낸다. 만일  $i \in W$ 이고  $j \in V \setminus W$ 이면  $\delta(W)$ 는  $i-j$  컷이라 부른다. 포함되는 모든 링크가 삭제되었을 때, 노드집합이 정확히 둘로 나누어지는 컷을 단일컷이라 부르며, 노드집합이 셋 이상으로 분리되는 컷을 다중컷이라 부른다.

그래프 상에서 노드간 연결도(connectivity)는 서로 떨어진 두 노드사이에 존재하는 중복되지 않는 연결경로의 수로 정의된다. 따라서 그래프로 표현된 통신망에서의 노드간 연결도는 통신망의 장애대비 능력을 표현해 준다. 그래프  $G$ 에서 노드  $i$ 와 노드  $j$ 사이의 연결도를  $\lambda(i, j, G)$ 로 나타내며, 그래프  $G$ 의 연결도  $\lambda(G)$ 는 노드간 연결도 중에서 최소값을 취한다. 통신망의 연결도가  $k$ 라고 하면 임의의 두 노드설비 사이에  $k$ 개 이상의 서로 중복되지 않는 연결경로가 존재하게 되므로, 어떠한  $(k-1)$ 개 이하의 링크에 동시장애가 발생하더라도 해당 통신망은 모든 노드간의 서비스 요구를 처리해 줄 수 있는 것이다. 그러나 연결도는  $k$ 개미만의 링크 장애에 따르는 영향만을 설명해 주며, 노드설비간에 부과되는 서비스 요구의 양을 감안하지 않기 때문에 그 활용의 범위가 한정적일 수밖에 없으며, 주로 노드설비간의 구조적 중복을 부과하는 통신망의 설계에 활용된다[Alevras et al., 1997; Grötschel et al., 1995].

### 2.1 정량적 생존도, $S_k(G)$

통신망에 부과되는 서비스 요구의 처리가 통신망의 본래적인 기능임을 감안하여, 구성요소 장애에 따르는 서비스 처리 능력을 보다 객관적으로 표현하기 위한 정량적 생존도의

개념이 1990년대 이후 활발하게 연구되어 왔다[Cardwell et al., 1989; Brush & Marlow, 1990]. 처리해야 될 서비스 요구가 적은 경우에는 부분적인 설비장애의 영향이 상대적으로 작고, 부과되는 서비스 요구가 많은 경우라면 부분적인 설비 장애도 매우 큰 영향을 미칠 수 있다는 사실을 반영하는 개념이며, 장애가 발생한 경우에도 여전히 처리되어 질 수 있는 서비스 양의 상대적인 비율로서 정의된다. 고려대상 구성요소 종류에 따라 링크생존도와 노드생존도로 구분되며, 감안하는 동시장애 설비의 수 ( $k$ )에 따라  $k$ -링크생존도 또는  $k$ -노드생존도 등으로 분류될 수 있다. 본 연구의 대상인  $k$ -링크생존도  $S_k$ 는  $k$ 개의 링크에 동시에 장애가 발생하는 경우에도 여전히 처리되어지는 서비스 양의 비율로서 정의된다.

## 2.2 정량적 생존도의 계산

통신망의 정량적 생존도는 1990년대 이후에 활발히 연구되어졌다. 망의 구조적 특징만을 표현하는 연결도와 달리 망에 부과되는 서비스의 양을 기준으로 분석할 수 있다는 장점을 가지지만, 특정시점에 부과되는 서비스 요구를 객관적으로 산출하기 어려울 수 있다는 제약도 가진다. 특히 서비스 요구를 감안하기 때문에 정확한 값의 산출과정이 매우 복잡하다.

근본적으로 정량적 생존도는 최악의 장애발생을 전제로 그 상황에서도 처리되는 서비스의 양을 통하여 계산된다. 최악의  $k$ 개 링크장애에 따른 최대손실(Maximum Loss)을  $L_k$ 라 하면  $S_k = (\sum t_{ij} - L_k) / \sum t_{ij}$ 와 같이 계산되며,  $L_k$ 의 산출과정이 링크생존도 계산의 핵심이 된다.

한편, 특별한 제약이나 전제가 없는 일반적인 상황에서  $L_k$ 를 구하는 문제는 NP-hard임이 이미 밝혀졌다[Myung et al., 1999]. 3-terminal 컷 문제로부터의 변환을 통해서 증명되었으며, 모든 NP 클래스의 문제가  $L_k$ 를 구하는 문제로 쉽게 변환될 수 있음을 의미한다. 따라서  $L_k$ 의 값을 효율적으로 구하는 일반적 해법을 얻는 것은 가능하지 않다고 볼 수 있으며, 현실적인 효율을 감안하여  $L_k$ 의 상한값과 하한값을 구하는 연구가 필요하게 되었다. 최근에  $S_k$ 를 구하기 위한 정수계획모형과  $S_k$ 의 상한값을 구하는 휴리스틱이 개발되었고, 선형계획 완화모형을 이용한 하한값 산출절차가 제시되었다[Myung et al., 2001].

## 2.3 정량적 생존도를 위한 정수계획모형

$k$ -링크 장애에 따른 최대손실  $L_k$ 를 산출하기 위한 문제는 다음과 같은 정수계획모형으로 정형화될 수 있다. 최악의 피해를 규정하는 링크 단절 집합을 표현할 수 있도록, 링크의 장애 여부를 나타내는 변수  $x_e$ 와 두 노드설비간 서비스의 단절여부를 나타내는 변수  $y_{ij}$ 를 아래와 같이 정의하면 (P)의 최적해가  $L_k$ 의 참값이 된다.

$$x_e = \begin{cases} 1, & \text{링크 } e \text{에 장애가 발생하는 경우} \\ 0, & \text{링크 } e \text{가 정상일 경우} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{노드 } i \text{와 노드 } j \text{가 단절된 경우} \\ 0, & \text{노드 } i \text{와 노드 } j \text{가 연결되어 있는 경우} \end{cases}$$

$$(P) \quad \max \quad z = \sum_{i,j \in V} t_{ij} y_{ij} \quad (1)$$

$$\text{sub to} \quad \sum_{e \in E} x_e \leq k, \quad (2)$$

$$\sum_{e \in p} x_e \geq y_{ij}, \quad p \in P_{ij}, \quad \forall i, j \in V, \quad (3)$$

$$x_e \in \{0, 1\}, \quad e \in E, \quad (4)$$

$$y_{ij} \in \{0, 1\}, \quad i, j \in V, \quad (5)$$

여기서  $P_{ij}$ 는 노드  $i$ 와 노드  $j$ 사이의 모든 경로의 집합을 나타낸다. 제약식 (2)은 단절되어지는 링크의 수가  $k$ 개 이하여야 함을 나타내며, 제약식 (3)은 노드  $i$ 와 노드  $j$ 사이의 어떠한 경로에라도 하나 이상의 장애 링크가 있을 때에 두 노드가 단절됨을 의미한다. 즉, 두 노드사이에 어떠한 형태로라도 연결경로가 존재하면, 두 노드사이의 서비스 요구가 처리된다는 것을 의미하며, 제약식 (3)의 수는 매우 많을 수 있다. 제약식 (4)와 (5)는 변수들의 정수조건을 나타낸다.

모형 (P)의 최적해를 구하면 그 값이 곧  $L_k$ 이다. 그러나 일반적인 정수계획모형문제의 최적해를 구하는 것은 간단하지 않을 뿐만 아니라, 제약식 (3)의 수도 매우 많다. 그렇지만 (P)의 선형계획 완화문제의 최적해를 구함으로써  $L_k$ 의 상한값 즉  $S_k$ 의 하한값을 구할 수 있다.

## 2.4 LP 완화모형을 강화시키는 유효부등식

한편, 최근의 연구[Myung et al., 2001]는 모형 (P)에 대한 아래와 같은 추가적인 유효부등식을 제시하였다.

$$\sum_{(i,j) \in T_c} y_{ij} \leq 1 \quad \forall T_c \subset T \quad (6)$$

여기서  $T_c$ 는 서비스 요구  $T$ 의 부분집합이며,  $k$ 개 이하의 링크장애만으로는 동시에 차단되지 않는 서비스 요구들의 집합을 나타낸다. 즉  $k$ 개 이하의 링크장애만을 고려하는 경우, 하나의  $T_c$ 에 속하는 서비스 요구들 중에서는 하나이하만 단절될 수 있는 것이다. 정수계획모형 기준으로는 제약식 (6)이 단순한 하나의 중복되는 제약식이지만, 선형계획 완화모형에서는 목적식 값을 낮추어 주는 유효부등식이 된다. 결과적으로 제약식 (6)을 이용함으로써 보다 강화된  $S_k$ 의 하한값을 얻을 수 있는 것이다. 그러나 이러한 집합  $T_c$ 는 다양한 형태로 많은 수가 존재할 수 있기 때문에 모두 파악하여 이용하는 것은 쉽지 않은 과정이다. 3장에서 제약식 (6)의 활용을 위하여  $T_c$ 를 찾는 휴리스틱을 제시한다.

## 2.5 문제 단순화를 위한 사전처리

링크생존도를 정확하게 산출하기 위해서는 개념적으로 다중컷을 일일이 나열하게 되며, 컷의 수는 노드 수의 증가에 따라 기하급수적으로 증가하게 된다. 따라서 최적해에 영향을 주지 않는 노드나 링크를 고려대상에서 배제시킬 수 있다면 전체적인 계산 효율을 높일 수 있다. 특히 본 연구에서 다루는 분지한계절차에서는 상한값 및 하한값을 산출하기 위한 많은 반복계산이 이루어지므로 사전처리에 의한 문제 단순화는 매우 중요한 단계이다.

$k$ -링크생존도는 노드사이에 존재하는  $k$ -링크 다중컷에 의해서 결정되기 때문에, 노드 간의 연결도에 근거한 단순화가 가능하다. 즉, 그래프  $G$ 에서 두 노드  $i$ 와  $j$  사이의 연결도  $\lambda(i, j, G)$ 가  $k$ 보다 크면, 노드  $i$ 와 노드  $j$ 사이의 서비스 요구는 어떠한  $k$ -링크장애라도 견딜 수 있으며 노드  $i$ 와 노드  $j$ 는 하나의 노드로 축약될 수 있는 것이다. 두 노드 사이에  $k$ 개 보다 많은 수의 경로가 존재할 때, 하나의 경로를 차단하기 위해서는 최소한 하나이상의 링크가 차단되어야 하므로,  $k$ 개의 차단만으로는 두 노드 사이의 모든 경로를 차단할 수 없기 때문이다.

상호간에  $(k+1)$ 이상의 연결도를 가지는 노드집합의 축약을 위해서는 Gomory-Hu의 컷트리(Cut tree)[Hu, 1963]를 사용하는 것이 효과적이다. 노드간 모든 링크의 흐름용량을 1로 한 후, 최대흐름문제(maximum flow problem)를 반복적으로 풀어서 컷트리를 만들면, 이로부터 모든 노드간의 연결도를 확인할 수 있다. 컷트리에서  $(k+1)$ 이상의 연결도를 가지는 노드집합을 쉽게 찾을 수 있으며, 이를 축약하는 사전처리가 이루어진다. 그래프의 노드 수를  $n$ 이라 할 때, 모든 노드간의 연결도를 구하기 위해서 열거적 개념에 의존하면  $nC_2$ 번의 최대흐름문제를 풀어야 하지만, 컷트리를 이용하면  $(n-1)$ 번만 풀어서 구할 수 있는 것이다.

## III. $k$ -링크생존도를 위한 분지한계절차의 개발

2장에서 정리한 바와 같이 일반적인  $k$ 에 대한 정량적 링크생존도  $S_k$ 의 계산문제는  $NP$ -hard이므로  $S_k$ 의 정확한 값을 산출하는 효율적인 해법을 찾는 것은 쉽지 않다. 그렇지만 통신망의 설계 및 유지관리를 위한 망의 성능분석 과정에서 링크생존도  $S_k$ 의 값을 적절한 시간에 효율적으로 구하는 것은 필요한 절차이다. 본 장에서는  $S_k$ 의 정확한 값을 산출하기 위한 분지한계절차를 개발한다.

분지한계절차는 정수계획법 문제를 풀기 위한 방법이며, 우선 의사결정변수의 정수조건을 완화시킨 선형계획 문제의 해를 구하는 것으로 시작된다. 일부 변수가 정수 조건을 만족시키지 못하는 경우, 정수 조건을 만족시키지 못하는 변수에 추가적인 제약을 가하여 원 문제를 두 개의 서로 다른 하위 문제로 분할하고, 각각의 하위문제에 대한 해를 구하게 된다. 하위문제에서 정수해를 얻거나 또는 더 이상 바람직한 해를 얻는 것이 가능하지 않다고 판단되는 경우에는 해당문제를 고려대상에서 제외하고 다른 하위문제로 옮겨가며, 그렇지 않은 경우에는 다시 하위문제로 분리하게 된다. 이처럼 하나의 문제를 둘 이상의 하위

문제로 나누는 과정을 분지(branch)라 하며, 각 하위문제에서는 목적식 값을 하한값(또는 상한값)으로 활용하여 보다 나은 정수해를 얻을 가능성을 확인(bound)하게 된다.

$L_k$ 를 구하는 문제는 최대화 문제이기 때문에 원문제의 선형계획 완화문제를 풀어서  $L_k$ 의 상한값을 얻을 수 있으며, 중간과정에서 얻어지는 정수해는 하한값이 된다. 본 연구에서는  $L_k$ 의 하한값을 얻는 휴리스틱을 도입하여 초기 하한값으로 활용하며, 선형계획 완화모형의 목적식 값을 강화하고자 추가적인 유효부등식(2장의 제약식 (6))을 반영하였다.

### 3.1 $L_k$ 의 하한값( $S_k$ 의 상한값) 산출과정

$L_k$ 는  $k$ 개의 링크 단절에 의해서 발생할 수 있는 서비스 요구의 최대 손실을 나타낸다. 따라서 임의의  $k$ -링크 단절에 의한 손실을 계산하면 임의의 하한값을 얻게 된다. 즉, 최악의 손실은  $k$ 개의 링크장애에 따라 처리되지 못하는 서비스 요구의 최대값이므로 어떠한 피해라도 최악의 피해보다는 적기 때문에 하한값이 되는 것이다. 물론 모든  $k$ -링크 다중컷을 모두 찾아보는 것은 현실적으로 타당하지 못하므로, 전체  $k$ -링크 다중컷 중 일부를 찾아보는 적정의 휴리스틱을 통해서 하한값을 구할 수 있다. 다만, 휴리스틱의 시간 효율은 검색의 범위와 반비례하며, 얻어지는 하한값의 성능 효율은 검색의 범위와 비례하게 되므로, 이러한 두 가지 상충을 적절히 조화시키는 것이 필요하며, 본 연구에서는 다음과 같은 휴리스틱을 이용하였다. 이는 기존의 연구[Myung et al., 2001]에서 제시된 상한 도출 절차를 개선하여 검색의 범위를 약간 넓힌 것이다.

기본적으로  $k$ -링크 다중컷은 몇 개의 단일컷으로 이루어지므로 고려할 수 있는 단일컷을 나열하고, 이들을 중첩시켜  $k$ -링크 다중컷을 찾는 방법을 사용하였다. 또 링크용량을 1로 둔 채로 Gomory-Hu의 컷트리를 구성하면,  $n-1$ 개의 서로 중복되지 않은 단일컷을 얻을 수 있으므로, 사전처리 단계에서 얻어진 컷트리를 사용하였다. 다중컷을 얻기 위하여 단일컷을 선택하는 경우에도 다양한 순서 개념을 적용할 수 있으나, 본 연구에서는 단일컷에 포함되는 링크의 수와 단일컷의 단절에 의해서 발생하는 서비스 요구 손실을 복합적으로 고려하여 두 가지 하한값을 구하였다. 먼저, 단일컷에 연계된 서비스 요구 손실을 그 단일컷에 포함되는 링크수로 나눈 비율을 구하고, 이 비율이 가장 큰 컷을 우선적으로 다중컷에 포함시켰다. 하나의 단일컷이 선택된 이후에 현재까지 다중컷에 포함된 링크의 수가  $k$ 개 미만이어서 컷을 추가하는 경우, 추가로 선택할 단일컷의 링크당 서비스 요구손실은 최초의 컷을 선택할 때와는 달리 순수 증가부분만을 계산하게 된다. 즉, 단일컷을 다중컷에 추가함으로써 늘어나는 서비스 요구 손실과 추가로 단절되는 링크수의 비율에 의해서 다음 선택될 컷을 정하는 것이다. 두 번째로 단일컷의 단절에 의해서 발생하는 서비스 요구손실의 절대량이 큰 컷을 우선적으로 선택하는 방법도 병행하여 사용하였다. 컷에 포함되는 링크 수를 감안하지 않는 극단적인 방법이지만 첫 번째 방법으로 얻어지는 하한값 보완적으로 사용함으로써 전체적인 효율을 높이려 한 것이다.

이상과 같이 얻어진  $L_k$ 의 두 가지 하한값 중 좋은 것이 최초의 정수해로 사용되어지며, 분지한계절차가 이루어짐에 따라 중간단계 하위문제에서 보다 나은 정수해를 얻게 되면, 이를 새로운 하한값으로 수정하여 사용하게 된다.



### 3.2 선형계획 완화문제의 풀이

( $P$ )의 선형계획 완화문제의 최적해는  $L_k$ 의 상한값이 된다. 그러나 제약식 (3)의 수는 매우 많을 수 있기 때문에 이 선형계획문제를 한 번에 풀려는 시도에는 무리가 따른다. 본 연구에서는 제약식 (3)중의 일부분만을 포함하는 선형계획문제를 풀어서 얻어진 해가 ( $P$ )의 실현가능 해인지를 확인하고, 실현가능하지 않으면 충족되지 못하는 제약식을 찾아서 추가하는 반복적 절차를 사용한다. 반복적으로 실행가능영역을 잘라 없애는 일종의 절단면 해법(cutting plane method)이라 하겠다. 그리고 분지한계절차의 중간에서 풀어야 하는 하위문제의 형태도 ( $P$ )와 동일하기 때문에 역시 절단면 해법을 이용하여 풀어 간다. 특히 특정변수를 이용하여 하위문제로 분지하는 경우, 제약식을 추가하지 않고 목적식 계수를 조정하는 방법을 사용하여 수리적 특성이 가능하면 그대로 유지되도록 하였다.

제약식 (3)중에서 일부만을 포함시키고 선형계획문제를 풀어가기 때문에, 얻어진 해를 대상으로 모든 제약식 (3)을 만족하는지를 확인해야 한다. 역시 이 때에도 모든 제약식 (3)을 나열하고 그 만족여부를 확인하는 것은 효율적이지 못하므로, 제약의 의미를 통해서 충족되지 않은 제약식을 직접 찾아내어 추가하는 방법을 사용한다. 제약식 (3)의 의미는 단절되어지는 두 노드 사이의 모든 경로는 차단되어야 한다는 의미이며, 두 노드사이의 모든 경로에 장애링크가 하나 이상씩 포함되면 두 노드는 차단되게 된다. 이 경우에  $x_e$ 를 링크  $e$ 의 길이라고 보면 두 노드사이의 경로의 길이는 해당경로에 포함되는 장애링크의 수를 의미하게 된다. 그리고, 두 노드  $i$ 와  $j$ 사이의 모든 링크의 길이가  $y_{ij}$ 이상이면 두 노드는  $y_{ij}$ 만큼 차단되는 것이다. 결국 두 노드 사이의 최단경로의 길이와  $y_{ij}$ 를 비교하면 해의 실현가능성을 확인할 수 있는 것이다.  $x_e$ 를 링크 길이로 두었을 때  $i$ 와  $j$ 사이의 최단경로 길이가  $y_{ij}$ 보다 작은 경우, 해당 최단경로에 대한 제약식 (3)을 추가하게 된다. 그러면 새로운 선형계획문제에서는 현재의 최적해가 배제되어진다. 이 과정을 반복하여 ( $P$ )의 선형계획 완화문제에 대한 최적해를 구할 수 있으며, 이 값은  $k$ -링크생존도를 규정하는 하한값으로 활용되어진다. 각 하위문제들도 원 문제에서  $x_e$ 의 목적식 계수만 조정되어진 형태이기 때문에 동일한 절차를 이용하여 풀이할 수 있다.

### 3.3 유효부등식을 이용한 $L_k$ 상한값의 강화

제약식 (6)을 추가하는 경우에 최적 정수해는 영향을 받지 않지만, 선형계획 완화문제에서는 목적식 값이 향상되므로 전체적으로 고려되는 하위문제의 수를 줄일 수 있게 된다. 다만  $k$ 개의 링크장애로는 동시에 차단될 수 없는 서비스 요구의 집합  $T_c$ 들을 찾는 것이 단순하지는 않다. 하나의 서비스 요구 쌍(예를 들면,  $i$ 에서  $j$ 로의 서비스 요구와  $p$ 에서  $q$ 로의 서비스 요구)에 대해서, 링크의 흐름용량을 1로 두고 두 서비스 요구를 동시에 흘러 보내는 이중최대흐름문제(bi-maximum flow problem)를 풀어서 그 최대 흐름이  $k$ 보다 큰지를 확인하는 것이 가능한 방법이다. 한편, 서비스 요구의 수가 증가함에 따라 미리 모든  $T_c$ 를 찾는 것도 효과적이지 못하다. 본 연구에서는 제약식 (6)을 배제한 선형계획문제를

풀어서 얻어진 해를 바탕으로  $y_{ij}$ 가 양수 값을 가지는 서비스 요구만을 대상으로  $T_c$ 을 찾는 방법을 사용하였다. 개별 서비스 요구 쌍에 대하여 이중최대화문제를 풀어서 하나의  $T_c$ 에 속하는 지를 확인해 두며, 모든 쌍에 대한 검토가 마쳐지면 이들의 관계를 정리하여 가능하면 많은 서비스 요구를 포함하는 제약식 (6)을 구성하고, 현재의 선형계획문제의 해가 이를 만족시키지 못하는 경우 제약식에 추가하는 것이다. 4장에서의 시험적용 계산결과 는 제약식 (6)을 활용하는 경우의 계산 효율을 보여주고 있다.

### 3.4 분지한계절차 구현

정수계획문제에 대한 좋은 분지한계절차를 구현하기 위해서는 하위문제의 해를 쉽게 얻을 수 있어야 하고 하위문제에 대한 하한값(최대화문제의 경우)으로 사용할 수 있는 적절한 정수해 산출절차가 필요하다. 또한 정수해가 아닌 하위문제에서 분지변수를 정하는 기준과 하위문제를 선택하는 기준도 명확하여야 한다.

본 연구에서는 Gomory-Hu의 컷트리로부터 누적방식으로 적정의 정수해를 구하는 절차를 초기 상한값 확보수단으로 사용했다. 또 (P)의 선형계획 완화문제를 풀어서 정수해가 얻어지지 않는 경우, 정수 값을 가지지 못하는 하나의  $x_e$ 를 선택하여 그 값을 각각 0과 1로 고정시키는 두 개의 하위문제를 만들어 분지하였다. 변수의 값을 0 또는 1로 고정시키기 위해서는 제약식을 추가하거나 또는 분지변수 값을 상수로 고정시키는 방법을 사용할 수도 있지만, 본 연구에서는 선형계획 완화문제의 제약식이 매우 많다는 문제의 특성을 감안하여 목적식 계수를 조정하는 방법을 이용하였다. 원 문제가 최대화 형태의 0-1문제이기 때문에 0으로 고정하는 경우에는 매우 작은 목적식 계수를 부과하고, 1로 고정하는 경우에는 매우 큰 값을 부과하였다. 4장의 시험적용에서 선형계획 완화문제를 풀기 위해서 CPLEX callable library를 이용하였기 때문에 이러한 목적식 계수 조정이 간단히 이루어질 수 있었으며, 선형계획모형의 제약식이 추가되거나 삭제되는 형태가 아니기 때문에 여러 개의 하위문제를 반복적으로 풀어가기에 보다 효과적이라고 판단하였다. 또 정수조건을 가장 크게 어기는 변수를 분지변수로 정하였으며, 1로 고정시킨 하위문제를 먼저 선택하였고, 깊이우선탐색(depth first search)을 통하여 적정의 정수해가 보다 빨리 얻어질 수 있도록 하였다.  $L_k$ 를 계산하기 위한 분지한계절차는 아래와 같이 정형화된다.

Procedure B&B (최대 손실  $L_k$ 를 구하는 과정)

Step 0. (초기화)

- ① 상한값을 얻는 휴리스틱으로  $L_k$ 의 하한  $z_{LB}$ 를 구한다.
- ② 하위문제의 집합을 공집합으로 둔다:  $SP = \emptyset$

Step 1. (원문제의 풀이)

- ① (P)의 선형계획 완화문제를 풀이하여 최적 목적식 값  $z_{LP}$ 를 구한다.
- ② 정수해가 얻어졌으면 종료한다( $L_k = z_{LP}$ ).
- ③ 분지변수  $x_e$ 를 정하고 두 하위문제를  $SP$ 에 추가한다.

Step 2. (하위문제의 선택)

- ①  $SP = \emptyset$ 이면 종료한다( $L_k$ ).
- ② 가장 최근에 추가된 하위문제를 선택한다.

Step 3. (하위문제의 풀이)

- ① 선택된 하위문제를 풀어서 목적식 값  $z_{LP}$ 를 구한다.
- ② 정수해가 얻어졌으면

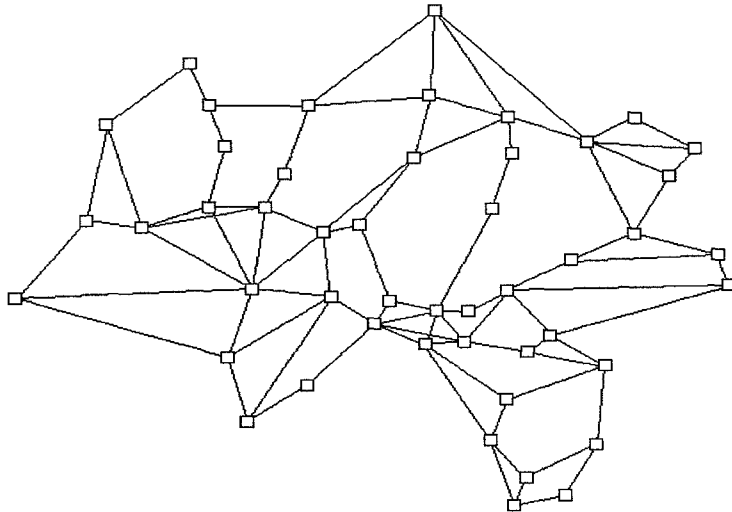
$$z_{LB} \text{를 수정한다: } z_{LB} = \max \{z_{LB}, z_{LP}\}$$

Step 2로 간다.

- ③  $z_{LP} < z_{LB}$ 이면 더 이상 좋은 해를 기대할 수 없으므로 Step 2로 간다.
- ④ 분지변수  $x_e$ 를 정하고 두 하위문제를 만들어  $SP$ 에 추가한다.  
Step 2로 간다.

### IV. 시험계산 및 결과 분석

3장에서 정리된 제시된 문제단순화,  $L_k$ 의 하한값( $S_k$  상한값) 도출절차, 선형계획 완화 문제 풀이절차를 통합하여 만들어진 분지한계절차는 C언어를 통하여 프로그램으로 구현되었다. 시험 적용 대상 문제를 만들기 위한 프로그램도 같이 만들어졌으며, 중간단계의 선형계획 완화문제를 풀기 위해서는 CPLEX Library를 이용하였다.



(그림 1) 서울지역 기반 네트워크의 구조

대상문제는 크게 두 그룹으로 나누어 만들어졌다. 첫 번째 그룹은 (그림 1)과 같은 서울지역의 전화망구조를 기반으로 하여 네트워크 모양은 고정시킨 채, 서비스 요구를 다양하게 변

화시켜서 만들었으며, 두 번째 그룹은 네트워크의 구조와 서비스 요구를 모두 변화시켜서 만들었다. 소규모 네트워크로서 노드 수 30개인 경우와 대규모 네트워크로서 노드 수 80개 및 100개인 경우를 포함시켰으며, 서비스 요구의 수도 다양하게 변화시켰다. 네트워크 모양을 얻기 위해서 (100×100) 형태의 사각형을 전제하고 임의의 위치에 노드 설비를 위치시켰고, 이들이 상호 연결되도록 하나의 트리로 연결한 후, 지정된 수만큼의 추가링크를 설치하였다. 노드 간의 서비스 요구는 주어진 수만큼 일정한 범위에서 임의추출방식으로 생성하였다.

시험적용은 600MHz CPU를 가지는 펜티엄급 PC에서 실행되었다. 모두 150문제를 대상으로  $k$ 를 2에서 5까지 변화시키며 시험계산을 수행하였고, 그 결과는 <표 1>~<표 4>로 정리하였다. 표에 나타난 바와 같이 대상 네트워크의 크기( $n$ : 노드 수  $m$ : 링크 수  $t$ : 서비스 요구 수), 사전처리가 이루어진 후 네트워크의 크기( $n'$ : 노드 수  $m'$ : 링크 수  $t'$ : 서비스 요구 수), 링크생존도( $S_k$ )의 값과 소요된 시간을 기존 연구의 결과인 링크생존도의 상한값(UB) 및 하한값(LB1, LB2), 소요시간과 같이 정리하였다. UB는 휴리스틱에 의해서 얻은 초기 상한값이며, LB1은 제약식 (6)을 고려하지 않은 상태에서 얻은 하한값이고, LB2는 제약식 (6)까지를 고려했을 때의 하한값이다. 이와 더불어 중간단계에서 만들어진 제약식 (3)의 수( $n_{SP}$ )와 제약식 (6)의 수( $n_C$ ) 등을 동시에 정리하였다. 표에 나타난 모든 결과는 10개 문제에 대한 결과를 산술 평균한 값이다.

<표 1>은 서울지역 전화망 구조를 기반으로 작성된 30문제에 대한 결과이다. 우선 기존 연구 상한값의 경우,  $k$ 가 2 또는 3인 경우에 최적해와 동일하며  $k$ 가 커짐에 따라 최적해와 차이가 커짐을 알 수 있다. 선형계획 완화문제의 해인 하한값의 경우도 최적해에 상당히 가까이 있음을 알 수 있다. 특히 추가적인 유효부등식을 사용하는 경우에는  $k$ 가 커지는 경우라도 최적해에 근접한 하한값을 제공하고 있으며, 다만  $k$ 가 커질수록 분지한계절차에 소요되는 시간이 급격히 증가하고 있고, 제약식 수와 관련시켜 볼 때, 추가 유효부등식의 적용에 많은 시간이 소요됨을 알 수 있다.

<표 1> 서울지역 네트워크를 대상으로 한 시험계산 결과 ( $n=48$ )

$k$	원문제			단순화 문제			$S_k$	CPU	UB	LB1	LB2	CPU	$n_{SP}$	$n_C$
	$n$	$m$	$t$	$n'$	$m'$	$t'$								
2	48	87	96	10.0	17.0	11.6	90.1	0.10	90.1	90.1	90.1	0.07	12.2	0.0
2	48	87	384	10.0	17.0	22.0	91.7	0.08	91.7	91.7	91.7	0.10	23.0	0.0
2	48	87	672	10.0	17.0	31.1	91.9	0.06	91.9	91.9	91.9	0.10	31.1	0.0
3	48	87	96	28.0	54.0	52.1	79.5	0.26	79.9	78.9	79.5	0.22	85.3	100.3
3	48	87	384	28.0	54.0	145.7	81.2	0.18	81.2	81.2	81.2	0.14	223.5	0.0
3	48	87	672	28.0	54.0	237.4	80.7	0.26	80.7	80.7	80.7	0.21	332.0	0.0
4	48	87	96	38.0	72.0	79.5	71.3	2.18	72.7	70.3	71.3	1.19	272.7	192.3
4	48	87	384	38.0	72.0	263.7	73.2	2.19	74.3	72.9	73.2	3.25	878.0	487.8
4	48	87	672	38.0	72.0	435.7	74.0	4.67	77.0	73.9	74.0	15.89	1451.2	1317.0
5	48	87	96	46.0	86.0	94.9	65.1	34.39	69.8	60.6	64.2	2.60	504.4	132.3
5	48	87	384	46.0	86.0	368.0	68.3	463.89	72.7	63.4	67.9	66.81	1805.6	2070.8
5	48	87	672	46.0	86.0	631.1	68.5	998.03	73.5	64.2	68.5	187.38	2847.0	4387.4

<표 2>~<표 4>는 임의추출방식으로 만들어진 문제에서의 시험적용 결과이다. 노드 설비 수 30개, 80개, 100개인 경우를 포함하고 있으며, 링크설비 수 및 서비스 요구의 수를 다양하게 변화시킨 경우를 포함하고 있다. 역시 동일한 규모의 10개 문제결과의 평균을 표시하고 있다.  $k$ 가 적은 경우에 분지한계절차를 통한 최적해의 산출이 상당히 효율적임을 보여주지만, 예상대로  $k$ 의 증가에 따라 소요시간이 크게 늘어나는 것을 알 수 있다.  $k$ 의 증가에 따라 휴리스틱에 의한 링크생존도의 상한값도 최적해와 상당한 차이를 보이는 경우도 적지 않아서 향후 보다 체계화된 휴리스틱의 연구가 필요하다는 것을 보여준다. 다만, 선형계획 완화문제의 풀이를 통한 하한값은 안정적으로 최적해에 근접하고 있는 바, 통신망 계획공자의 입장에서는 통신망 생존도의 최소 성능 확인에 활용할 수 있다고 판단된다.

<표 2> 소규모 네트워크에 대한 시험적용 결과( $n=30$ )

$k$	원문제			단순화 문제			$S_k$	CPU	UB	LB1	LB2	CPU	$n_{SP}$	$n_C$
	$n$	$m$	$t$	$n'$	$m'$	$t'$								
2	30	40	120	17.3	24.0	51.3	68.7	0.09	69.5	68.5	68.7	0.10	66.6	16.2
2	30	40	240	17.3	24.0	87.4	71.4	0.13	73.7	71.2	71.4	0.14	113.0	104.9
2	30	50	120	9.6	14.1	18.6	83.5	0.06	83.8	83.0	83.5	0.10	21.1	3.3
2	30	50	240	9.6	14.1	29.4	83.4	0.07	84.0	82.9	83.4	0.10	34.4	10.5
2	30	70	120	3.8	5.2	3.5	91.5	0.05	92.3	91.7	92.1	0.08	3.5	0.9
2	30	70	240	3.8	5.2	5.0	92.5	0.05	93.9	92.3	93.0	0.08	5.0	2.1
3	30	40	120	26.6	36.8	105.7	57.4	0.34	61.5	56.5	57.2	0.32	261.9	61.9
3	30	40	240	26.6	36.8	199.9	57.9	1.34	60.8	57.5	57.8	0.73	459.3	138.5
3	30	50	120	19.3	35.2	63.2	76.2	0.26	77.4	74.9	76.2	0.14	119.5	110.4
3	30	50	240	19.3	35.2	108.9	76.6	0.52	77.7	75.9	76.6	0.31	217.7	279.9
3	30	70	120	8.5	18.2	14.5	87.6	0.08	87.9	86.3	87.6	0.10	18.6	18.0
3	30	70	240	8.5	18.2	22.9	88.5	0.09	88.6	86.8	88.5	0.11	31.5	59.8
4	30	40	120	29.9	39.9	119.8	46.9	4.40	52.4	45.7	45.8	0.53	330.4	4.1
4	30	40	240	29.9	39.9	239.0	49.4	7.47	52.6	48.4	48.6	1.87	626.5	45.3
4	30	50	120	26.4	46.8	105.4	68.7	12.79	70.6	66.2	67.9	1.30	365.6	221.3
4	30	50	240	26.4	46.8	197.9	70.2	20.98	71.3	67.6	69.9	5.95	712.6	785.5
4	30	70	120	16.3	41.0	45.0	82.4	0.21	84.8	81.5	82.0	0.19	98.9	83.6
4	30	70	240	16.3	41.0	76.3	83.4	1.06	86.1	82.1	83.1	0.31	165.1	206.8
5	30	40	120	29.9	39.9	119.8	38.9	1.26	43.5	38.0	38.0	0.60	325.4	0.6
5	30	40	240	29.9	39.9	239.0	41.8	6.18	46.6	40.3	40.4	2.08	629.0	3.2
5	30	50	120	29.3	49.6	117.4	60.6	3.74	66.9	58.7	59.6	1.38	458.2	55.6
5	30	50	240	29.3	49.6	233.2	62.6	29.89	68.0	60.7	61.6	4.94	855.8	189.4
5	30	70	120	22.3	57.1	79.0	78.8	3.48	82.0	76.8	78.4	1.89	328.5	413.9
5	30	70	240	22.3	57.1	143.6	79.6	17.86	82.8	77.8	79.3	9.28	557.0	1537.8

<표 5>는 하한값을 강화하기 위한 추가적인 유효부등식(계약식 (6))의 효과를 보여주고 있다. 제약식 (6)을 제외하고 이루어진 분지한계절차에서의 결과와 제약식 (6)을 포함시킨 경우의 결과를 대비시켜서 정리하고 있으며, 임의적으로 선택된 문제를 대상으로 시험계산이 실행되었다. 각각의 경우에 소요된 시간(CPU)과 분지 수( $n_B$ ), 중간과정에 얻어진 정

수해의 수( $n_I$ ), 상한값에 의해서 고려대상에서 제외된 하위문제의 수( $n_F$ ), 중간과정에서 풀이된 선형계획문제의 수( $n_{LP}$ ) 등이 비교되고 있다. 표에서 보는 바와 같이 유효부등식을 추가하여 활용함으로써, 분지 수, 소요시간, 하위문제의 풀이 수 등 모든 면에서의 향상된 결과를 보여주고 있다. 결론적으로 제약식 (6)의 형태로 추가된 유효부등식은 선형계획 완화모형을 강화시켜 보다 향상된 링크생존도의 하한값을 제공할 뿐만 아니라, 분지한계절차의 효율도 크게 높여주는 것으로 확인되었다.

<표 3> 대규모 네트워크에 대한 시험적용 결과( $n=80$ )

$k$	원문제			단순화 문제			$S_k$	CPU	UB	LB1	LB2	CPU	$n_{SP}$	$n_C$
	$n$	$m$	$t$	$n'$	$m'$	$t'$								
2	80	100	100	50.3	69.3	69.5	80.8	0.22	81.1	80.5	80.8	0.15	87.4	18.5
2	80	150	100	17.8	29.3	19.8	93.0	0.10	93.2	93.0	93.0	0.09	21.1	1.5
2	80	200	100	7.6	12.3	7.2	94.0	0.10	94.1	93.9	94.0	0.10	7.3	0.9
3	80	100	100	73.4	96.5	97.0	72.8	9.60	74.7	71.1	72.0	2.07	390.6	84.1
3	80	150	100	39.1	79.4	53.7	90.3	0.46	90.8	89.7	90.1	0.24	95.7	66.4
3	80	200	100	16.9	39.2	19.0	92.1	0.13	92.8	91.4	92.1	0.09	20.2	11.6
4	80	100	100	79.7	99.9	99.9	64.0	53.77	67.6	61.4	61.7	3.80	520.4	12.4
4	80	150	100	59.8	125.9	85.0	86.9	3.73	88.4	86.3	86.6	0.90	283.7	42.9
4	80	200	100	33.9	94.8	44.9	90.0	0.60	90.3	89.2	89.7	0.17	68.5	40.9
5	80	100	100	80.0	100.0	100.0	57.0	74.42	62.6	53.1	53.2	3.09	477.7	0.2
5	80	150	100	71.2	142.7	95.9	83.7	283.63	86.0	82.9	83.1	4.49	620.4	29.5
5	80	200	100	52.7	151.5	74.2	87.8	8.06	88.4	87.0	87.2	0.67	219.1	32.0

<표 4> 대규모 네트워크에 대한 시험적용 결과( $n=100$ )

$k$	원문제			단순화 문제			$S_k$	CPU	UB	LB1	LB2	CPU	$n_{SP}$	$n_C$
	$n$	$m$	$t$	$n'$	$m'$	$t'$								
2	100	120	150	70.5	93.5	115.2	80.4	0.51	80.7	80.0	80.4	0.44	148.2	44.4
2	100	150	150	41.7	63.9	61.6	90.5	0.19	90.7	90.1	90.5	0.13	70.2	27.2
2	100	200	150	18.3	30.6	21.0	94.9	0.09	94.9	94.8	94.9	0.05	22.3	2.1
3	100	120	150	94.1	117.7	147.5	72.8	31.42	73.4	70.5	71.2	3.62	548.6	51.7
3	100	150	150	73.8	121.9	121.4	86.2	1.19	86.2	85.9	86.2	1.02	246.9	56.2
3	100	200	150	41.9	90.9	62.7	92.7	0.35	92.8	92.4	92.5	0.20	84.7	25.8
4	100	120	150	99.6	119.9	149.9	62.5	43.27	67.0	61.2	61.5	8.53	763.3	3.2
4	100	150	150	91.6	144.8	146.0	82.5	168.19	82.7	81.3	81.7	10.18	930.8	58.8
4	100	200	150	67.5	154.5	113.9	90.5	4.50	90.8	89.7	90.2	2.76	347.5	215.4
5	100	120	150	100.0	120.0	150.0	54.2	71.10	61.2	53.2	53.2	9.14	704.1	0.3
5	100	150	150	98.3	149.2	149.9	78.4	639.05	79.2	77.1	77.3	31.45	1216.0	16.5
5	100	200	150	85.2	186.3	139.8	88.1	110.26	88.7	87.3	87.6	16.41	1063.6	102.3

<표 5> 추가 유효부등식 활용 여부에 따른 효율 비교

k	문제 크기			S <sub>k</sub>	원 제약식만 활용하는 경우					추가 컷을 활용하는 경우				
	n	m	t		CPU	n <sub>B</sub>	n <sub>I</sub>	n <sub>F</sub>	n <sub>LP</sub>	CPU	n <sub>B</sub>	n <sub>I</sub>	n <sub>F</sub>	n <sub>LP</sub>
2	30	50	120	83.5	0.1	0.6	1.0	0.0	4.2	0.1	0.5	0.8	0.1	4.0
3	30	50	120	76.2	0.3	4.1	3.7	2.0	18.2	0.2	3.4	3.1	1.9	16.3
3	30	50	240	76.6	0.5	5.3	4.0	2.3	20.0	0.3	2.9	2.3	1.6	14.2
3	30	70	240	88.5	0.1	3.8	2.3	2.9	12.5	0.1	3.7	2.1	3.0	12.2
4	30	40	120	46.9	4.4	9.3	9.6	0.9	31.6	3.2	7.2	7.2	1.2	26.7
4	30	40	240	49.4	7.5	5.0	4.3	1.9	22.2	7.1	4.6	3.9	1.9	21.3
4	30	50	120	68.7	12.8	14.9	7.2	8.9	43.9	6.0	6.9	3.1	5.0	26.8
4	30	50	240	70.2	21.0	11.4	5.8	7.2	39.4	14.7	7.0	3.0	5.6	27.1
4	30	70	120	82.4	0.2	5.2	3.7	2.5	20.9	0.2	3.4	2.6	1.8	15.9
4	30	70	240	83.4	1.1	18.9	12.9	7.4	54.5	0.8	10.2	6.9	4.7	34.1
5	30	40	120	38.9	1.3	3.5	3.5	1.4	17.1	1.1	2.3	2.3	1.4	14.1
5	30	40	240	41.8	6.2	5.1	3.4	3.1	21.5	5.1	3.6	2.4	2.6	17.7
5	30	50	120	60.6	3.7	2.9	1.1	2.8	16.6	3.8	2.8	0.9	2.9	16.4
5	30	50	240	62.6	29.9	8.9	2.3	7.8	31.7	28.4	8.2	2.0	7.4	29.9
5	30	70	120	78.8	3.5	12.6	6.9	7.3	42.7	3.5	12.4	6.7	7.3	42.2
5	30	70	240	79.6	17.9	17.0	8.8	9.8	56.2	17.5	15.2	7.3	9.5	50.8
2	80	150	100	93.0	0.1	0.9	1.5	0.0	5.0	0.1	0.8	1.3	0.1	4.8
3	80	100	100	72.8	9.6	8.7	4.6	5.5	33.2	9.4	8.2	4.2	5.4	31.9
3	80	150	100	90.3	0.5	8.1	6.4	2.7	28.4	0.5	7.9	6.0	2.9	28.0
4	80	100	100	64.0	53.8	22.1	8.4	15.5	66.4	30.4	13.0	3.7	11.1	43.8
4	80	150	100	86.9	3.7	9.3	6.8	3.9	40.0	2.5	6.5	4.9	3.0	33.6
4	80	200	100	90.0	0.6	16.9	11.8	6.5	50.1	0.6	16.1	11.1	6.4	48.2
5	80	100	100	57.0	74.4	35.8	15.0	22.8	96.9	44.2	22.3	9.5	14.8	65.9
5	80	150	100	83.7	283.6	50.6	35.1	16.9	133.7	108.8	24.0	18.1	7.3	78.0
5	80	200	100	87.8	8.1	31.5	19.8	13.5	101.2	5.7	18.7	11.1	9.4	69.2
3	100	120	150	72.8	31.4	13.2	8.6	5.6	40.8	22.2	7.5	5.0	3.5	27.9
3	100	200	150	92.7	0.4	4.7	5.1	0.4	17.9	0.4	4.6	4.9	0.5	17.7
4	100	120	150	62.5	43.3	11.0	5.8	6.4	35.8	30.2	6.0	3.2	4.0	25.0
4	100	150	150	82.5	168.2	21.8	15.2	8.0	68.0	111.7	15.2	11.4	5.2	53.3
4	100	200	150	90.5	4.5	8.7	4.6	5.5	40.8	4.4	8.5	4.3	5.6	40.2
5	100	120	150	54.2	71.1	17.0	10.1	8.1	50.2	44.9	9.8	5.1	5.9	34.1
5	100	150	150	78.4	639.1	32.3	10.5	23.6	98.6	240.6	12.2	2.6	11.4	51.0
5	100	200	150	88.1	110.3	9.5	4.4	6.9	48.6	83.4	7.2	3.2	5.8	42.0
3	48	87	967	79.5	0.3	5.1	3.8	1.7	19.2	0.3	4.9	3.6	1.7	18.8
4	48	87	967	71.3	2.2	9.7	7.1	3.6	33.7	1.2	5.6	4.2	2.4	23.9
5	48	87	966	65.1	34.4	20.5	10.6	11.9	63.5	25.7	16.9	8.5	10.4	55.5
5	48	87	384	68.3	463.9	25.6	15.5	12.1	79.3	394.6	23.0	13.6	11.4	73.1

## V. 결 론

정량적 링크생존도는 통신망의 구성요소 중 링크설비에 장애가 발생하는 경우에 부과되는 서비스 요구가 어느 정도 처리될 수 있는지를 나타내는 성능지표이다.  $k$ 개의 동시 링크 장애를 대상으로 하는  $k$ -링크생존도의 산출은  $NP$ -hard 문제로 알려져 있으며, 본 연구에서는 그 정확한 값을 계산하는 분지한계절차를 구현하였다. 기존 연구에서 제시된 정수계획 모형을 기반으로 한 문제 단순화 과정, 휴리스틱을 통한 적정해 도출, 절단면 법을 이용한 선형계획 완화모형의 해법 등을 구현하였고, 휴리스틱 해법의 보완 및 선형계획 완화문제의 해를 개선할 수 있는 유효부등식도 추가하였다.

분지변수 선정방법과 하위문제 탐색 방법 등을 구체화하여 C언어와 CPLEX Library를 이용한 코드를 작성한 후, 시험계산을 수행하였다. 서울지역 전화망에 근거한 문제와 임의 추출방식으로 작성된 문제를 대상으로 노드 설비의 수 100개에 이르는 정도까지 폭넓은 범위의 시험적용을 행하였으며, 결과를 통하여 제시된 절차를 통하여 적정한 크기의 현실 문제에 대한 링크생존도 산출이 효과적으로 이루어질 수 있음을 확인하였다.

본 연구에서 제시된 절차는  $NP$ -hard문제에 대한 해법이기에 이론적인 가치나 현실적 적용범위에서 제한을 받을 수밖에 없다. 그렇지만 관련 휴리스틱의 효율 검토를 위한 객관적인 수단으로서, 그리고 적정한 범위내의 현실문제에 대한 해법으로서 분지한계절차가 효과적으로 사용될 수 있음을 보여준 실증적 의미를 가진다고 하겠다.

## 참고문헌

1. Oalevras, D., M. Grötschel, and R. Wessälly, "Capacity and survivability models for telecommunications networks," in Tech. reports SC 97-24, Konard-Zuse-Zentrum für Informationstechnik, Berlin, 1997.
2. Brush, G. G. and N. A. Marlow, "Assuring the dependability of telecommunications networks and services," *IEEE Network magazine* Jan. (1990), 29-34.
3. Cardwell, R., C. L. Monma, and T. Wu, "Computer-aided design procedures for survivable fiber optic networks," *IEEE J. SAC* 7 (1989), 1188-1197.
4. Cosares, S., N. D. Deutch, I. Saniee, and O. J. Wasem, "SONET toolkit: A decision support system for designing robust and cost-effective fiber-optic networks," *Interfaces* 25 (1995), 20-40.
5. Grötschel, M., C. L. Monma, and M. Stoer, "Design of survivable networks," in *Network Models*, M. O. Ball et al. (eds.) North-Holland, Amsterdam, 1995.
6. Hu, T. C., "Multi-commodity network flows," *Operations Research* 11 (1963), 344-360.
7. Kim, H. J., "Quantitative survivability in communication networks," *Journal of Management* (University of Ulsan) 2 (1995), 1-16.
8. Monma, C. L., B. S. Munson and W. R. Pullyblank, "Minimum-weighted two-connected spanning networks," *Mathematical Programming* 46 (1990) 153-171.



9. Monma, C. L. and D. F. Shallcross, "Methods for designing communications networks with certain two-connected survivability constraints," *Operations Research* 37 (1989), 531-541.
10. Myung, Y.-S., H.-J. Kim, and D.-W. Tcha, "Design of communication networks with survivability constraints", *Management Science* 45 (1999), 238-252.
- 11.. Myung, Y.-S. and H.-J. Kim, "A Cutting plane algorithm for computing  $k$ -edge survivability of a network," submitted to *European Journal of Operational Research*, 2001.
12. Wu, T., *Fiber network survivability*, Artech House, Boston 1992.
13. Wu, T., D. J. Kolar and R. C. Cardwell, "Survivable network architectures for broad-band fiber optic network: model and performance comparison," *Journal of Lightwave Technology* 6 (1988) 1698-1709.