韓國經營科學會誌
第 14 券 第 1 號
1989年 6月

－研究論文－

중앙집중식 전산망의 경제적 설계
－단말기 배치문제와 쌍대기반 해법－

Optimal Design of Centralized Computer Networks
－The Terminal Layout Problem and A Dual-based Procedure－

김 형 욱*
노 형 봉*
지 원 철**

ABSTRACT

The terminal layout problem is fundamental in many centralized computer networks, which is generally formulated as the capacitated minimum spanning tree problem(CMSTP). We present an implementation of the dual-based procedure to solve the CMSTP. Dual ascent procedure generates a good feasible solution to the dual of the linear programming relaxation of CMSTP. A feasible primal solution to CMSTP can then be constructed based on this dual solution. This procedure can be used either as a stand-alone heuristic or, else, it can be incorporated into a branch and bound algorithm. A numerical result is given with quite favorable results.

1. 서론

중앙집중식 전산망의 설치시에 기본적으로 발생하는 Topology 설계 문제인 단말기 배치 문제(Terminal Layout Problem)는 다음과 같다. 우선 하나의 중앙처리장치(Central Processing Unit), {1}과 (n－1)개의 단말기, K＝{2, 3, …, n}의 위치가 주어져 있다(I＝{1}∪K). 각 단말기는 중앙처리장치와 정보를 교환하는데, 그 경로로서 직접 연결된 통신회선을 이용할 수도 있고, 또는 다른 단말기(앞으로는 마디라 칭함)들을 경유하는 간접경로를 이용할 수도 있다. 각 마디 사이에 통신회선을 설치하는 데 소요되는 비용은 C_{ij}이다. 이 경우 모든 마디들을 최소의 비용으로 중앙처리장치(앞으

* 홍익대학교 경영학과
** 홍익대학교 산업공학과

이 논문은 1987년도 문교부 지원 한국학술진흥재단의 자유공모과제 학술연구조성비에 의하여 연구되었음

로 마디 1이라 칭함)에 연결시키는 네트워크를 설계하려고 한다. 단 통신회선의 기술문제 때문에 각 통신회선(앞으로 가지라 칭함)을 통해서 마디 1과 통신할 수 있는 마디의 수는 M개 이하로 제한되어야 한다.

이 문제를 네트워크 문제로 일반화시키면 Capacitated Minimum Spanning Tree(CMST) 문제가 된다. 왜냐하면 최소비용의 네트워크는 항상 Spanning Tree이고, 마디 1에 연결된 각 subtree 내의 마디 수에 제한이 있기 때문이다. 만일 이 문제에서 각 subtree 내의 마디 수에 대한 제약이 없으면 단순한 Minimum Spanning Tree(MST) 문제가 된다.

CMST 문제에 대한 연구는 지난 20여년 동안 꾸준히 진행되어 왔다. 이 연구들 중 휴리스틱해법에 관한 것들을 대부분 계산복잡도(Computational Complexity)를 낮추면서 계산해의 질을 높이고자 한 것들이다. Gavish와 Altinkemer[14]는 이러한 연구들에 대해 비교적 상세히 분석·비교하여 놓았다. 이들은 또한 최근에 계산해의 최대오류한도가 해석적으로 구해지는 휴리스틱해법[1]도 개발한 바 있다.

CMST 문제에 대한 또 하나의 연구부류는 최적화 기법에 대한 것들이다. CMST 문제는 Papadimitriou[24]에 의해 NP-complete임이 증명되었기 때문에 계산복잡도가 문제크기(마디의 수)의 Polynomial 함수로 표현되는 최적화 기법은 찾을 수 없다. 따라서 기존의 최적화기법들은 모두 분단탐색법(Branch and Bound Method)의 형태를 띄우고 있으며, 그 차이는 주로 최적해의 하한(Lower Bound)를 구하는 방법에 있다고 볼 수 있다. 이들 방법을 열거해 보면 (1) 단순히 MST 문제를 푼다[3, 4, 6], (2) Benders Decomposition을 이용한다[11], (3) Matroid Intersection Algorithm을 이용한다[18], (4) Lagrangean Relaxation을 이용한다[12, 13] 등을 들 수 있다. 이 중 가장 효율적이라고 평가받고 있는 것은 Lagrangean Relaxation을 이용한 것이라고 볼 수 있다.

그런데 가장 효율적이라고 하는 이들 해법조차도 문제크기(마디의 수)가 50 이상인 큰 문제에 대해서는 계산시간이 기하급수적으로 증가하기 때문에 현실적으로 의미가 있는 큰 규모의 문제에 대해서는 적용하기가 매우 어려운 실정이다. 따라서 본 연구에서는 보다 큰 규모의 문제도 효과적으로 풀 수 있는 해법으로서 쌍대기반 해법(Dual-based Procedure)을 개발하고자 한다. 이 해법은 혼합정수계획모형인 원문제를 선형계획완화시키면 이로부터 최적해의 하한을 구할 수 있다는 점에 착안한 것이다. 그러나 이 경우 원문제 크기가 커지면 하나의 하부문제(subproblem)에 불과한 선형계획완화문제 조차 풀기가 용이하지 않다. 따라서 쌍대기반 해법에서는 선형계획완화문제 대신 단순한 구조를 가지는 이의 쌍대문제를 매우 효율적인 휴리스틱 해법인 Dual Ascent Procedure를 사용해서 풀고자 하는 것이다.

이러한 본 연구의 접근방법은 기존의 연구방법 중에서 네번째 부류에 속한다고 볼 수 있다. 왜냐하면 선형계획 완화문제를 푼다는 것은, 원문제의 임의의 제약식에 대한 Lagrangean Relaxation을 취하고 이의 Lagrangean 쌍대문제를 푸는 것과 동일하기 때문이다. 이는 Lagrangean 완화된 하부문제가 Integrality Property[15]를 보이기 때문이다[13]. 그러나 본 연구해법은 Lagrangean 쌍대문제를 풀기 위해 Subgradient Optimization을 사용하는 기존의 연구들과 다음과 같은 세 가지 점에서 다르다.

(1) 반복절차가 진행됨에 따라 모든 쌍대변수 값이 단조증가하기 때문에 쌍대목적함수가 단조증가한다 ; 기존의 연구에서는 Lagrange 쌍대목적함수가 단조증가하지 않는다.

(2) 매 반복 절차시 복잡한 하부문제를 풀 필요가 없기 때문에 계산시간이 대폭 절감된다 ; Gavish[13]의 연구에서는 매번 NP-complete인 Degree Constrained MST 문제를 푼다.

(3) 또한 최종적으로 계산된 쌍대변수 값과 Complementary Slackness 조건을 이용해서 원문제의 좋은 실현가능해(종종 최적해)를 매우 쉽게 구할 수 있다; 기존의 연구에서는 매 반복절차시 얻어지는 실현 불가능해로 부터 가지교환방법을 통해서 하나의 실현가능해를 구한다.

다음 장에서는 CMST 문제에 대한 혼합정수계획모형을 정립하고, 3장에서는 이를 풀기 위한 쌍대기반 해법을 서술하기로 한다. 또한 4장에서는 이의 효율성을 입증하기 위해 기존 문헌에서 발췌한 예제를 풀어 볼 것이며 마지막으로 결론을 맺기로 한다.

2. 수리모형

CMST 문제를 수리모형으로 정립하면 다음과 같은 혼합 정수계획모형이 된다.

(P) 최소화

$$Z_p = \sum_{i \in I} \sum_{j \in K} C_{ij} \cdot Y_{ij}, \qquad (1)$$

제약조건

$$\sum_{i \in I} X_{jik} - \sum_{j \in K} X_{ijk} = \begin{bmatrix} -1,\ i=1 \\ 1,\ i=k \\ 0,\ i \neq 1,k \end{bmatrix} k \in K, \quad (2)$$

$$Y_{ij} - X_{ijk} \geq 0, \qquad i \in I,\ j,\ k \in K, \qquad (3)$$

$$M \cdot Y_{1j} - \sum_{k \in K} X_{1jk} \geq 0, \qquad j \in K, \qquad (4)$$

$$X_{ijk} \geq 0, \qquad i \in I,\ j,\ k \in K, \qquad (5)$$

$$Y_{ij} = 0 \text{ 또는 } 1, \qquad i \in I, j \in K. \qquad (6)$$

Y_{ij}는 가지 (i,j)가 해에 포함되는 가를 나타내는 정수변수이고, X_{ijk}는 가지 (i,j)를 통과하는 마디 1과 k 사이의 정보통신량을 나타내는 실수변수이다. 제약식 (2)는 마디 1로 부터 정보 한 단위가 마디 k로 흐른다는 것을 뜻하고, 제약식 (3)은 가지 (i,j)가 가설된 경우에만 정보가 흐를 수 있음을 나타낸다. 또한 제약식 (2)와 (3)은 마디 1과 모든 마디와의 사이에 정보가 흐를 수 있는 하나의 경로가 반드시 존재하도록 한다. 한편 제약식 (4)는 마디 1에서 하나의 가지를 통해 유통되는 정보량은 M 이하이어야 한다는 것을 뜻한다. 원래는 모든 가지에 이 제약이 있어야 하지만, 마디 1에 연결된 가지가 이 조건을 충족시키면 그 가지에 매달린 나머지 가지들은 당연히 이 조건을 충족시키기 때문에 생략한 것이다.

이 모형은 Multicommodity Network Flow Model에 기반을 두고 정형화시킨 것이다. 따라서 Multicommodity 변수 X_{ijk} 대신 이들을 집성시킨 변수 $X_{ij}(=\sum_{k \in K} X_{ijk})$를 사용한 기존 연구모형들[4, 11, 12]보다 선형계획완화시에 보다 좋은 하한을 얻을 수 있다. 이러한 Multicommodity 모형에 대해서는 Gavish[13]에 의해서 간단히 소개된 바 있으나, 본 연구모형과는 약간의 차이가 있으며 또한 이 모형을 풀 수 있는 특정 해법에 대해서는 전혀 언급된 바 없었다. 이 외에 CMST 문제에 대한 모형으로는 Gavish[13]의 것을 들 수 있는데, 이는 단일 변수 Y_{ij}만으로 구성되어 있는 것이 특징이다. 또한 이 모형은 Bin-packing 문제의 해로 부터 유도되는 매우 많은 Subtour 제거 제약식을 내포하고 있기 때문에 선형계획완화시 본 연구모형보다 좋은 하한을 제공할 수 있다. 그러나 제약식이 너무 많기 때문에 문제규모가 조금만 크더라도 이를 효과적으로 다루기가 어려워서 그 실용성은 미지수라 하겠다.

3. 쌍대기반 해법

모형 (P)에서 정수조건 (6)을 단순한 비음조건, $Y_{ij} \geq 0$, 으로 대체시키면 선형계획 완화문제 ($\bar{P}$)를 얻을 수 있다. ($\bar{P}$)의 쌍대문제는 다음과 같다.

($\bar{D}$) 최대화

$$\sum_{k \in K} (V_{kk} - V_{1k}). \qquad (7)$$

제약조건

$$V_{jk}-V_{ik}-W_{ijk}\leqq 0, \qquad i,j,k\in K. \tag{8}$$

$$V_{jk}-V_{1k}-W_{1jk}-U_j \quad 0, \qquad j,k\in K. \tag{9}$$

$$\sum_{k\in K} W_{ijk}\leqq C_{ij}, \qquad i,j\in K. \tag{10}$$

$$\sum_{k\in K} W_{1jk}+M\cdot U_j\leqq C_{1j}, \qquad j\in K. \tag{11}$$

$$W_{ijk},\ U_j\geqq 0. \tag{12}$$

여기서 V_{ik}, W_{ijk}, U_j 각각은 문제 $(\bar{P})$의 제약식 (2),(3),(4)에 대응하는 쌍대변수이다.

문제 $(\bar{D})$의 최적해를 구하는 것은 문제 $(\bar{P})$의 최적해를 구하는 것 만큼 어렵기 때문에 시간이 많이 소요된다. 따라서 본 연구에서는 이의 최적해를 구한다는 보장은 없으나 대부분 최적해나 이에 가까운 해를 손쉽게 구할 수 있는 해법, 즉 Dual Ascent Procedure를 사용하여 $(\bar{D})$를 풀기로 한다.

Dual Ascent Procedure를 기반으로 하는 쌍대기반 해법은 이미 여러 유형의 설비입지 선정문제에 적용되어 놀랄만한 효과를 거둔 바 있다[22]. 더우기 최근에는 다른 유형의 문제에도 적용하려는 시도가 이루어지고 있으며, 이의 대표적 예로서 Steiner Tree 문제에 대한 Wong[27]의 연구를 들 수 있다.

쌍대기반 해법을 소개하기 전에 우선 쌍대문제 $(\bar{D})$를 단순화시키기로 한다. 쌍대변수 V_{ik}는 마디 k로의 정보가 마디에서 흐름의 균형이 이루어지도록 하는 제약식에 관련된 것이다. 그런데 일반적으로 이 제약식들은 최소한 하나가 중복되어 있기 때문에 이에 대응하는 쌍대변수들 중 하나는 0이 되어도 무방하다. 따라서 우리는 편의상 $V_{ik}=0$으로 놓겠다. 또한 W_{ijk}는 목적함수에 포함되어 있지 않기 때문에 제약식 (8), (9)와 (12)가 허용하는 범위 내에서 최소값을 가지더라도 목적함수값에 전혀 영향을 미치지 않는다. 즉

$$W_{ijk}=max\{0,\ V_{jk}-V_{ik}\}, \quad i,j,k\in K\ ; \tag{13}$$

$$W_{1jk}=max\{0,\ V_{jk}-U_j\}, \quad j,k\in K. \tag{14}$$

따라서 단순화된 쌍대문제는 다음과 같다.

(D) 최대화

$$Z_D=\sum_{k\in K} U_{kk}, \tag{15}$$

제약조건

$$\sum_{k\in K} max\{0,\ V_{jk}-\ V_{ik}\}\leqq C_{ij}, i,j\in K. \tag{16}$$

$$\sum_{k\in K} max\{0,\ V_{jk}-\ U_j\}+M\cdot U_j\leqq C_{1j}, \qquad j\in K. \tag{17}$$

$$U_j\geqq 0,\ j\in K. \tag{18}$$

3.1. Primal Procedure

주어진 쌍대실현가능(dual feasible)해로 부터 원문제의 실현가능 정수해를 구하기 위하여 다음의 Complementary Slackness 조건을 사용한다.

$$[C_{ij}-\sum_{k\in K} max\{0,\ V_{jk}-V_{ik}\}]\cdot Y_{ij}=0\ ; \tag{19}$$

$$[C_{1j}-\sum_{k\in K} max\{0,\ V_{jk}-U_j\}-M\cdot U_j]\cdot Y_{1j}=0\ ; \tag{20}$$

$$[\ \sum_{j\in I} X_{jik}-\sum_{j\in K} X_{ijk}-b_{ik}]\cdot V_{ik}=0\ ; \tag{21}$$

$$[Y_{ij}-\ X_{ijk}]\cdot max\{0,\ V_{jk}-V_{ik}\}=0\ ; \tag{22}$$

$$[Y_{1j}-\ X_{1jk}]\cdot max\{0,\ V_{jk}-U_j\}=0\ ; \tag{23}$$

$$[M\cdot Y_{1j}-\sum_{k\in K} X_{1jk}]\cdot U_j=0. \tag{24}$$

여기서 b_{ik}는 모든 $k\in K$에 대해서 $i=1$이면 -1, $i=k$이면 1, 다른 경우에는 0인 상수이다.

우선 하나의 네트워크 $G=(I,E)$가 주어져 있다고 하자. 여기서 E는 모든 가지들의 집합을 뜻한다. 또한 하나의 쌍대실현가능해 $\{V_{jk}^+,\ U_j^+\}$가 주어져 있다고 하자. 그러면 다음과 같은 보조가지 집합 A를 구할 수 있다.

$$A=\{(i,j)\in E\mid C_{ij}-\sum_{k\in K}\max\{0,\ V_{jk}^{+}-V_{ik}^{+}\}=0 \text{ 또는 } C_{1j}-\sum_{k\in K} max\{0,\ V_{jk}^{+}-U_j^{+}\}-M\cdot U_j^{+}=0\}. \tag{25}$$

집합 E내의 가지들은 방향에 의해서도 정의되었기 때문에 $(i,j) \in A$이나 $(j,i) \notin A$인 가지가 있을 수 있다.

마디집합 I와 보조가지집합 A에 의해 정의된 망을 보조망 G'이라 하자. 그러면 G'은 문제 (P)의 실현가능 Spanning Tree를 최소한 하나 이상 포함하고 있다(이러한 요건을 충족시킬 수 있는 쌍대해를 구하는 방법은 다음 절에서 논의하기로 한다). 따라서 보조망 G'에서 문제 (P)의 실현가능 Spanning Tree를 추출하여 이에 대응하는 정수해 $\{Y_{ij}^+, X_{ijk}^+\}$를 구하면, 이들은 (24)를 제외한 모든 Complementary Slackness 조건을 충족시킴을 알 수 있다. 조건 (24)를 위배하는 경우는 단지 마디 1에 연결된 Subtree 중 마디수가 M보다 작은 것이 존재하는 경우에 한한다. 선형계획이론에 의해서 이 Complementary Slackness 조건의 위배는, $\{Y_{ij}^+, X_{ijk}^+\}$에 대응하는 원문제 목적함수 값 Z_P^+와 $\{V_{jk}^+, U_j^+\}$에 대응하는 쌍대 목적함수 값 Z_D^+의 차이와 다음과 같은 관계가 있다 :

$$Z_P^+ - Z_D^+ = \sum_{j \in K} [M \cdot Y_{1j}^+ - \sum_{k \in K} X_{1jk}^+] \cdot U_j^+. \tag{26}$$

즉 Complementary Slackness 조건을 위배하지 않는 실현가능해는 최적해인 것이다. 따라서 본 해법에서는 되도록 이 조건을 위배하지 않는 쌍대해와 원문제해를 구하고자 한다.

보조망 G'에서 최적의 실현가능 Spanning Tree를 찾는 것은 G에서 찾는 것과 같이 NP-complete 문제이다. 그렇지만 G'의 가지집합 A는 G의 가지집합 E에서 대폭 축소된 것이므로 G'에서 최적해를 찾는 것이 훨씬 쉬우리라 예상된다. 본 연구에서는 보조가지 집합 A가 충분히 작다고 생각하여 최적화기법 대신 단순한 휴리스틱 해법으로서 실현가능해를 찾고자 한다. 휴리스틱 해법으로서는 기존의 개발해법 중 어느 것을 사용해도 무방하지만, 짧은 계산시간으로 매우 좋은 해를 찾을 수 있는 Essau-Williams[8]의 것이 가장 바람직하다고 생각되어 본 연구에서는 이를 사용하기로 한다。

이제는 Primal Procedure를 공식적으로 서술하고자 한다。 여기서 마디 i의 가중치 W_i는 마디 i를 포함하는 요소와 마디 I과의 최소 거리이고, 가지 (i,j)의 Tradeoff 함수는 $t_{ij} = C_{ij} - W_i$이다.

〈Primal Procedure〉

단계 0.

각 마디는 모두 독립요소이다.

단계 1.

a. 각 마디에 대해 가중치 W_i를 계산한다.
b. 두개의 요소를 연결하는 모든 가지에 대해 t_{ij}를 구한다.

단계 2.

a. 모든 t_{ij}중에서 최소치 t_{pq}를 찾는다.
b. 가지 (p,q)를 연결시켜서 한 *subtree*의 마디수가 M보다 커지면 가지 (p,q)는 더이상 고려하지 않고 단계 2. a.로 간다.
c. (p,q)를 연결시킨다.

단계 3.

모든 마디가 한 요소에 포함되면 절차를 끝낸다. 그렇지 않으면 단계 1로 되돌아 간다. □

위의 절차에서 시간이 가장 많이 소요되는 부분은 단계 1이다. 만일 모든 가지가 다 고려되는 경우에는 계산복잡도가 O(N²*log* N)이다. 그런데 본 절차에서 대상으로 하는 보조망의 가지집합 A는 단지 소수의 가지들만으로 구성되어 있기 때문에 계산복잡도는 현저히 감소될 것이다. 예로써 각 마디에 연결된 가지 수의 최대가 H개라면 계산복잡도는 O(HN *log* N)으로 준다. 또한 이러한 경우에 Essau-Williams 해법의 효율도 더욱 증진되리라 여겨진다.

3.2. Dual Ascent Procedure

Dual Ascent Procedure(DAP)는 단순화된 쌍대문제 (D)의 구조를 이용해서 최적해 또는 이에 가까운 좋은 해를 용이하게 구하고자 한다. 이를 위해서 DAP는 두 단계의 절차로 나뉘어야 하는데, 첫 단계에서는 $U_j=0$으로 놓고 V_{ik}만을 증가시키고, 둘째 단계에서는 U_j 값을 단계적으로 증가시키면서 V_{ik} 값을 추가로 증가시키고자 한다.

논리 전개상 편의를 위해 몇 가지 용어를 정의하기로 한다. 보조망 G'에서 마디 i로 부터 j 까지의 경로가 존재하면 "연결되었다"고 한다. 또한 마디 j에서 i 까지의 경로 역시 존재하면 "상호연결되었다"고 한다. 하나의 마디집합 Q 내의 모든 마디 쌍이 상호연결되어 있으면 Q는 "상호연결집합"이라고 한다. 더 이상 확장시킬 수 없는 최대 상호연결 집합을 "상호연결요소"라고 한다. 마디 i는 j에 연결되어 있는데 j는 i에 연결되어 있지 않으면 i는 j를 "매달고 있다", 또는 j는 i에 "매달려 있다"고 한다. 상호연결요소에서 자체마디가 요소 밖의 다른 마디에 매달려 있지 않으면 "뿌리요소"라 한다. 뿌리요소 R 의 "cutset" CS(R)은 $(i,j)\notin A$ 이면서 $j\in R$이고 $i\in\bar{R}$인 가지 (i,j)의 집합이다.

〈절차 I〉

단계 0.

$V_{ik}^+=0,\ i,k\in K$; $U_j^+=0,\ j\in K$; $A=\phi$;
$S^+(i,j)=C_{ij}-\sum_{k\in K}max\{0,\ V_{jk}^+-V_{ik}^+\},\ i,j\in K$;
$S^+(i,j)=C_{ij}-\sum_{k\in K}max\{0,\ V_{jk}^+-U_j^+\}-M\cdot U_j^-$,
$i,j\in K$.

단계 1.

한 뿌리요소 R를 찾는다. 만일 뿌리요소가 없으면 절차를 끝낸다.

단계 2.

a. $\triangle=S^+(i^*,j^*)=\ min\ \{S^+(i,j)\mid(i,j)\in CS(R)\}$
b. $V_{ik}^-\leftarrow(V_{ik}^++\triangle/\mid R\mid),\ i,k\in R.$
c. $S^+(i,j)\leftarrow(S^+(i,j)-\triangle),\ (i,j)\in CS(R).$

단계 3.

a. $A\leftarrow A\cup\{(i^*,j^*)\}$.
b. 단계 1로 되돌아간다. □

절차 I 은 모든 쌍대변수 V_{ik}^+, U_j^-를 0으로 놓고 시작해서 이들 중 V_{ik}^+만을 증가시켜 줌으로써 쌍대목적함수를 증가시키고자 한다. 처음에는 모든 마디가 뿌리요소이어서 각 마디에 대응하는 V_{kk}^+가 증가대상이 된다. $S^+(i,j)$는 가지 (i,j)에 대응하는 제약식의 여유분(Slack)을 나타낸다. 따라서 단계 2에서 임의의 V_{ik}^+, $i,k\in R$ 를 증가시키면 그만큼 $S^-(i,j)$, $(i,j)\in CS(R)$를 감소시켜야 한다. 물론 이 경우 제약조건(16)과 (17)이 허용하는 한도 내에서 쌍대변수 값을 증가시켜야 한다. 또한 $S^-(i^*, j^*)$는 새로이 0이 되었기 때문에 (25)에서 정의한 바와 같이 가지 (i^*,j^*)는 A에 편입되어야 한다. 단계 3이 진행될 때마다 뿌리요소의 수가 감소되므로 절차 I 은 유한번의 반복절차에 의해 끝내진다.

절차 I 에서 보조가지 집합 A가 확장되는 순서는 Edmonds[5]의 해법이 Directed MST를 찾는 순서와 동일하다(이는 Wong[27]의 연구를 참조하면 증명이 가능하다). 따라서 절차 I 이 끝났을 때의 쌍대목적함수 값 Z_P^+는 망 G에서 구한 MST의 비용과 같다. 즉

$$Z_P^+=\sum_{k\in K}V_{kk}^+=v(MST). \qquad (27)$$

그러므로 DAP가 최종적으로 얻을 수 있는 하한은 최소한 MST를 풀어서 얻을 수 있는 것보다 월등히 좋다는 것을 알 수 있다.

다음에 소개할 절차 II 는 절차 I 에서 0으로 놓았던 U_j^-의 값을 단계적으로 증가시키면서 다른 쌍대변수V_{jk}^-의 값을 추가로 증가시켜서 쌍대목적

함수 Z_D^+를 개선시키고자 하는 것이다.

이를 설명하기 위해 몇가지 용어를 추가 정의하기로 한다. 우선 연결방향에 관계없이 가지들로 연결되어 있는 마디들의 집합을 "연결집합"이라 한다. 연결집합에서 자체마디가 마디 1 이외에는 매달려 있지 않으면 "의사뿌리집합"이라 한다. 의사뿌리집합 T의 cutset, CS(T)는 $(i,j)\not\in A$이면서 $j\in T$이고 $i\in\bar{T}$인 가지 (i,j)의 집합이다. $|Q|$는 마디집합 Q 내의 마디 수이다.

$$L(T)=\{j\in T\mid U_j^+=0,\ (1,j)\in A\}; \tag{28}$$
$$N(T)=\{j\in T\mid U_j^->0,\ (1,j)\in A\}; \tag{29}$$

〈절차 II〉

단계 1.

a. 다음 조건을 만족하는 의사뿌리집합들을 찾는다:
$0<M\cdot|L(T)|<|T|-|N(T)|$.

b. $|T|$가 가장 작은 의사뿌리집합 T를 찾는다. 더 이상 찾을 수 없으면 단계 6으로 간다.

단계 2.

a. $\triangle_1=\min\{C_{1j}/M-U_j^+\mid(1,j)\in A,\ j\in T\}=C_{1h}/M-U_j^+$;
$\triangle_2=\min\{S^+(i,j)/|T|\mid(i,j)\in CS(T),\ (i,j)\not\in A\}$.

b. 만일 $\triangle_1\le\triangle_2$이면 단계 4로 간다.

단계 3.

a. 각각의 $j\in N(T)$에 대해서
$V_{jk}^-\leftarrow[V_{jk}^++\max\{0,\ \triangle_2-V_{jk}^+\}],\ k\neq j,\ k\in T.$

b. 각각의 $j\in L(T)$에 대해서
$U_j^+\leftarrow(U_j^++\triangle_2)$; $V_{jj}^+\leftarrow(V_{jj}^+-(M-1)\triangle_2)$; $V_{jk}^+\leftarrow(V_{jk}^++\triangle_2)$, $k\neq j,\ k\in T$.

c. $(i,j)\not\in A$인 각각의 $j\in T$에 대해서
$V_{jj}^+\leftarrow(V_{jj}^+\triangle_2)$; $V_{jk}^+\leftarrow(V_{jk}^++\max\{0,\ \triangle_2-V_{jk}^-\})$, $k\neq j,\ k\in T$.

d. 쌍대변수 값의 변화에 따라서 $S^+(i,j)$와 A를 재조정한다.

e. 단계 5로 간다.

단계 4.

a. $(i,j)\in A,\ j\neq h$인 각각의 $j\in T$에 대해서 다음식을 만족하는 ∇_j를 구한다.
$(V_{jj}^++\nabla_j)+(M-1)[\max\{V_{jk}^+\mid k\neq j,\ k\in T\}+\nabla_j]=C_{1j}$.

b. 만일 $\triangle_1>\nabla=\max\{\nabla_j\mid(i,j)\in A,\ j\neq h,\ j\in T\}$이면 $\triangle_1=\triangle$로 놓는다.

c. $U_h^+\leftarrow(U_h^++\triangle_1)$; $V_{hh}^+\leftarrow(V_{hh}^+-(M-1)\triangle_1)$; $V_{hk}^+\leftarrow(V_{hk}^++\triangle_1)$, $k\neq h,\ k\in T$.

d. $(i,j)\in A,\ j\neq h$인 $j\in T$에 대해서 $U_j^+>U_h^+$이면 $U_j^+\leftarrow U_h^+$; $V_{jk}^-\leftarrow(V_{jk}^++\triangle_1)$, $k\in T$.

e. $(i,j)\not\in A,\ j\neq h$인 $j\in T$에 대해서 $V_{jk}^+\leftarrow(V_{jk}^++\triangle_1)$, $k\in T$.

f. 쌍대변수 값의 변화에 따라서 $S^+(i,j)$와 A를 재조정한다.

단계 5.

a. 현재의 의사뿌리집합 목록중 T를 포함하는 것이 없으면 단계 1로 간다.

b. 단계 3 또는 4의 결과로 새로이 등장한 의사뿌리집합을 T'이라 하자. $T'=\phi$ 이거나 $T'\supseteq T$이면 단계 1로 간다.

c. $\triangle=\min\{S^+(i,j)/|T'|:(i,j)\in CS(T')\}$.

d. $V_{jk}^-\leftarrow(V_{jk}^++\triangle)$, $j,k\in T'$.

e. 쌍대변수 값의 변화에 따라서 $S^+(i,j)$와 A를 재조정하고 단계 1로 간다.

단계 6.

a. $(i,j) \in A$, $U_i^+ = 0$인 모든 j에 대해서, 쌍대목적함수가 감소되지 않으면서 U_i^+를 증가시킬 수 있으면 $U_i^+ = \varepsilon > 0$로 놓는다 ; $(1,j)$를 A에서 없앤다.

b. 절차를 끝낸다 □

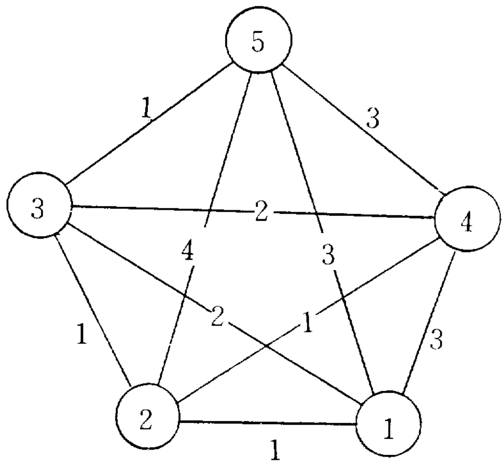

그림 1. CMST 문제의 예

단계 1에서는 쌍대변수 U_i^+를 증가시킴으로서 목적함수를 증가시킬 수 있는 의사뿌리집합을 찾는다. 이때 U_i^+를 증가시킬 수 있는 양은 제약식 (15)와 (16)에 의해 제한을 받는데, 단계 2에서는 가능한 최대증가량을 구한다. 단계 3과 4는 U_i^+의 증가량이 제약식 (15)와 (16)에 의해 제한되는 경우 각각을 다룬다. 그런데 제약식 (16)에 의해 U_i^+의 증가량이 결정되는 경우에는 증가가능량 모두를 증가시킨다고 해서 목적함수가 반드시 증가한다고 볼 수 없다. 따라서 목적함수의 증가가 보장되는 한도 내에서 증가량을 조정해야 하는데 이것이 단계 4의 a, b에서 결정된다. U_i^+의 증가량이 결정되면 나머지 쌍대변수 V_{ik}^-들을 증가시켜서 목적함수를 개선시킨다. 이때 쌍대변수 값의 변화에 따라서 대응하는 여유 $S^-(i,j)$들이 조정되는데, 이는 보조가지집합 A의 변화를 의미하므로 이의 조정이 수반되어야 한다.

단계 3 또는 4의 결과로 기존의 의사뿌리집합

표 1. 그림 1의 예에 대한 절차 II 계산과정

단계	T	$\triangle_1$	$\triangle_2$	새로운 쌍대변수 값	변화된 보조가지
1	(2,3,4,5)				
2		1/2	1/4		
3.b.				$U_2^+ = 1/4,\ V_{22}^- = 3/4,\ V_{23}^- = V_{24}^- = V_{25}^- = 1/4$	
3.c.				$V_{33}^- = V_{44}^- = V_{55}^- = 5/4,\ V_{32}^+ = V_{34}^+ = V_{35}^+ =$ $V_{42}^+ = V_{43}^+ = V_{45}^+ = V_{52}^+ = V_{53}^+ = V_{54}^+ = 1/4.$	
3.d.					$(3,2)^-$, $(4,2)^-$, $(1,3)^*$
1	(2,3,4,5)				
2		1/4	1/4		
4.c.				$U_2^+ = 1/2,\ V_{22}^+ = V_{23}^+ = V_{24}^+ = V_{25}^+ = 1/2.$	
4.d.				$U_3^- = 1/2,\ V_{33}^+ = 3/2,\ V_{32}^+ = V_{34}^+ = V_{35}^+ = 1/2.$	
4.e.				$V_{44}^+ = V_{55}^+ = 3/2,$ $V_{42}^- = V_{43}^+ = V_{45}^+ = V_{52}^+ = V_{53}^+ = V_{54}^+ = 1/2.$	
4.f					$(1,4)^*$, $(1,5)^*$
6				$U_4^+ = U_5^+ = \varepsilon$	$(1,4)^-$, $(1,5)^-$

* : 추가된 가지, − : 제거된 가지.

목록에 없는 새로운 의사뿌리집합이 등장하는 경우가 있는데, 이 경우에는 쌍대변수 U_i^+의 증가 없이 나머지 변수들의 증가만으로 목적함수가 개선될 수 있다. 단계 5에서는 이러한 의사뿌리집합을 찾아서 쌍대변수 값을 조정하고 이에 대응하는 $S^+(i,j)$, A의 조정을 수행한다. 이러한 모든 작업이 반복적으로 수행되어 더 이상 목적함수의 증가가 불가능하면 절차를 끝내게 된다. 그런데 최종적으로 구해진 보조가지집합 A를 보면 이에는 마디 1에 불필요하게 연결되어 있는 가지들이 많이 있음을 알 수 있다. 따라서 Primal Procedure가 보다 좋은 해를 구할 수 있도록 이러한 불필요 가지를 A로부터 제거할 필요가 있는데 이것이 최종 단계 6에서 수행되는 것이다.

4. 계산예제

앞 절에서 소개한 쌍대기반 해법의 효율성을 살펴보기 위해 그림 1에서 보는 바와 같은 네트워크 G에서 CMST를 구해보기로 한다. 이 예는 Chandy와 *Lo*[3]가 제시한 것이다. 가지 위의 숫자는 가지 비용 C_{ij}를 나타내는데 그 비용은 방향에 관계없이 같다. 즉 $C_{ij}=C_{ji}$이다. 또한 가지능력상한이 $M=2$로 주어져 있다.

Dual Ascent Procedure의 절차 I을 적용했을 때 그 결과는 $V_{22}^+=V_{33}^+=V_{44}^+=V_{55}^+=1$이고, 쌍대목적함수 값은 $Z_D^+=\sum_{k=2}^{5} V_{kk}^+=4$, 보조가지집합 $A=\{(1,2), (2,3), (2,4), (3,2), (3,5), (4,2), (5,3)\}$. 앞 장에서 밝힌 바와 같이 위의 예에 대한 MST는 가지 $(1,2)$, $(2,3)$, $(2,4)$, $(3,5)$로 구성되어 있고(이는 A의 부분집합이지만 방향을 무시하면 A와 같다고 볼 수 있음), 그 비용 또한 Z_D^+와 같은 4이다.

다음 DAP의 절차 II를 적용한 결과를 다음 표 1이 보여주고 있다. 최종적으로 얻어진 보조가지집합은 $A=\{(1,2), (1,3), (2,3), (2,4), (3,5), (5,3)\}$. 즉 절차 I에서의 A에서 $(3,2)$, $(4,2)$가 없어지고 $(1,3)$이 추가된 결과와 같다. 즉 가지능력제약을 충족시키기 위해서 마디 1에 연결된 가지가 하나 증가되었고, 상호 연결된 가지에서 불필요한 방향의 것이 없어졌다. 또한 최종 쌍대변수 값은 $V_{22}^-=V_{23}^+=V_{24}^+=V_{25}^+=1/2$, $V_{33}^+=V_{44}^+=V_{55}^+=3/2$, $V_{32}^+=V_{34}^+=V_{35}^+=V_{42}^+=V_{43}^+=V_{45}^+=V_{52}^+=V_{53}^+=V_{54}^+=1/2$, $U_2^+=U_3^+=1/2$, $U_4^+=U_5^+=\varepsilon$이고 나머지는 모두 0이다. 최종쌍대목적함수 값은 $Z_D^+=\sum_{k=2}^{5} V_{kk}^-=5$이다. 즉 절차 I에서의 목적함수값 4에서 1만큼 증가한 결과이다.

최종 가지집합 A에서 Primal Procedure를 적용하면 다음과 같은 실현가능해가 얻어진다. 즉 $(1,2)$, $(2,4)$, $(1,3)$, $(3,5)$의 네개의 가지로 구성된 Spanning Tree이다. 이때 비용함수값은 $Z_P^+=C_{12}+C_{24}+C_{13}+C_{35}=5$. 그런데 $Z_P^+=Z_D^+$이므로 이 실현가능해는 최적해라는 것을 알 수 있다. 이는 또한 Complementary Slackness 조건 (24)를 모두 충족시킨다는 것을 보아도 알 수 있다.

5. 결론

본 연구에서는 중앙집중식 통신망의 설계시 빈번히 발생하는 CMST 문제를 효율적으로 풀 수 있는 쌍대기반 해법을 개발하였다. 이 해법은 Dual Ascent Procedure와 Primal Procedure에 의해 구성되어 있는데, 이 각각은 최적목적함수 값의 하한과 상한을 구한다. 따라서 본 해법은 분단탐색법과 연결되면 CMST 문제의 최적화기법으로 사용될 수 있고, 만일 문제 규모가 너무 커서 최적해를 찾는 것이 어렵다면 독립적으로 쓰여서 하나의 휴리스틱해법이 될 수도 있는 것이다.

쌍대기반 해법의 효율성을 살펴보기 위해 한 예제를 풀어보았다. 그 결과 MST로 부터 얻은 하한보다 월등히 좋은 하한을 얻을 수 있었으며, 이를 위해 필요한 계산량은 매우 작다는 것을 알 수 있었다. 또한 상한과 하한의 일치로 최적해임이 입증되어 더 이상 분단탐색의 필요가 없었다.

참 고 문 헌

1. Altinkemer, K and B. Gavish, "Heuristics with constant error guarantees for the design of tree networks", Management Sci. 34, 331-341, 1988.
2. Boorstyn, R. R. and H. Frank, "Lange scale network topological optimization", IEEE Trans. Comm. 25, 29-47, 1977.
3. Chandy, K. M. and T. Lo, "The capacitated minimum spanning tree", Networks 3, 173-182, 1973.
4. Chandy, K. M. and R. A. Russell, "The design of multipoint linkage in a teleprocessing tree network", IEEE Trans. Computers 21, 1062-1066, 1972.
5. Edmonds, J., "Optimum branchings", J. of Res. of the National Bureau of Standards—B. Mathmatics and Math. Physics 71B, 233-240, 1967.
6. Elias, D. and M. J. Ferguson, "Topological design of multipoint teleprocessing networks", IEEE Trans. Comm. 22, 1753-1762, 1974.
7. Erlenkotter, D., "A dual-based procedure for uncapacitated facility location", Operations Res. 26, 992-1009, 1978.
8. Essau, L. R. and K. C. Williams, "On teleprocessing system design Part 2", IBM Systems J. 5, 142-147, 1966.
9. Frank, H. and W. Chou, "Topological optimization of computer networks", Proc. IEEE 60, 1385-1397, 1972.
10. Frank, H., I. T. Frisch, R. Van slyke and W. S. Chon, "Optimal design of centralized computer networks", Networks 1, 43-47, 1971.
11. Gavish, B., "Topological design of centralized computer networks-formulations and algorithms", Networks 12, 355-377, 1982.
12. Gavish, B., "Formulations and algorithms for the capacitated minimal directed tree", J. Assoc. Comput. Mach. 30, 118-132, 1983.
13. Gavish, B., "Augmented Lagrangean based algorithms for centralized network design", IEEE Trans. Comm. 33, 1247-1257, 1985.
14. Gavish, B. and K. Altinkemer, "A parallel savings heuristic for the topological design of local tree networks", Proc. IEEE-INFOCOM'86, 130-139, 1986.
15. Geoffrion, A. M. "Lagrangean relaxation and its uses in integer programming", Math. Programming Study 2, 82-114, 1974.
16. Karnaugh, M., "A new class of algorithms for multipoint network optimization", IEEE Trans. Comm. 24, 500-505, 1976.
17. Kershenbaum, A., "Computing capacitated minimum spanning trees efficiently", Networks 4, 299-310, 1974.
18. Kershenbaum, A. and R. Boorstyn, "Centralized teleprocessing network deoign", Networks 13, 279-293, 1983.

19. Kershenbaum, A., R. Boorstyn and R. Oppenheim, "Second order greedy algorithms for centralized teleprocessing network design", IEEE Trans. Comm. 28, 1835-1838, 1980.
20. Kershenbaum, A. and W. Chou, "A unified algorithm for designing multidrop teleprocessing networks", IEEE Trans. Comm. 22, 1762-1772, 1974.
21. Kobayashi, H., "Communication network design and control algorithms-a survey", Computer Sci. Res. Report RC 9223, IBM T. J. Watson Res. center, Yorktown Heights, 1982.
22. Krarup, J. and P. M. Pruzan, "The simple plant location problem", Europ. J. of Operational Res. 12, 36-81, 1983.
23. McGregor, P. M. and D. Shen, "Network design : an algorithm for access facility location problem", IEEE Trans. Comm. 25, 61-73, 1977.
24. Papadimitriou, C. H., "The complexity of the capacitated tree problem", Networks 8, 217-230, 1978.
25. Sharma, R. L., "Design of an economical multidrop network topology with capacity constraints", IEEE Trans. Comm. 31, 590-591, 1983.
26. Van Roy, T. J. and D. Erlenkotter, "A dual-based procedure for dynamic facility location", Management Sci. 28, 1091-1105, 1982.
27. Wong, R. T., "A dual ascent approach for Steiner tree problems on a directed graph", Math. Programming 28, 271-287, 1984.