

최대 가중치 합 그래프 마디 분리에 관한 연구 (이론분야)

서울대학교 산업공학과

홍성필, 정성진, 김영일

초록

이 논문에서는 호의 가중치가 음수가 될 수 있는 무향 그래프에서 마디집합들을 선택했을 때, 각 마디집합 사이에 걸쳐 있는 호의 가중치 합을 최대화하는 최대 가중치 합 그래프 마디 분리 문제에 대해 연구한다. 우선 $P \neq NP$ 라는 가정 하에서, 이 문제는 NP -hard이며 어떠한 근사 계수를 가지는 근사해법도 존재하지 않는다는 것을 밝힌다. 또한 이 문제를 정수 계획으로 풀기 위해, 그래프의 절단면을 나타내는 벡터들이 만드는 다면체의 극대면(facet)을 정의하는 부등식들을 추가 제약식으로 사용하는 branch-and-cut 알고리즘을 제시하고 구현하였다.

1. 서론

1.1 연구의 배경 및 목적

일상 혹은 산업 현장에서 발생하는 여러 최적화 문제를 표현하는 데 그래프를 좋은 도구로 사용할 수 있다. 그래프 위에서 만들어 낼 수 있는 문제는 여러 종류가 있으나, 이 논문에서는 그래프에서 특정한 마디를 분리하여 생기는 절단면(cut)에 대해 다루려 한다. 절단면 문제는 반도체의 설계, 통신망의 설계 및 관리 등에서 발생할 수 있으며, 지금까지 여러 형태의 절단면 문제에 관한 연구가 이루어졌다. 이 논문은 호의 가중치가 음수가 될 수 있는 무향 그래프에서 몇 개의 마디집합들을 선택하여, 마디집합 사이에 걸쳐 있는 호의 가중치 합을 최대화하는 문제를 다루고 있다. 이 문제는 지금까지 많은 연구가 진행되어 온

maximum cut 문제를 변형, 확장한 것이다.

1.2 논문의 구성

이 논문의 구성은 다음과 같다. 2 장에서는 관련된 문제들의 계산 복잡도와 근사해법을 소개한다. 3 장에서는 논문에서 다루고자 하는 문제를 정의하고 계산 복잡도를 밝힌다. 4 장에서는 문제의 근사 불가능성에 대해서 언급한다. 5 장에서는 문제를 정수계획으로 풀 때, 계산 시간을 줄이는 데 도움이 될 수 있도록 절단면에 대응하는 벡터들이 만들어 내는 가능해 공간의 극대면(facet)을 정의하는 부등식을 다룬다. 6 장에서는 문제를 정수 계획으로 풀기 위한 branch-and-cut 알고리즘을 제시한다. 7 장에서는 6 장에서 제시한 알고리즘의 계산 시간 측정 결과를 정리한다. 마지막으로 8 장에서는 논문에서 얻은 결론과 추후 연구 방향에 대해 언급한다.

2. 관련 연구

이 장에서는 논문에서 다루고자 하는 문제의 속성을 밝히는 데 관계가 있는 몇 가지 문제들을 소개한다.

2.1 Maximum Cut

문제 2.1.1 Maximum Cut

무향 그래프 $G=(V,E)$ 의 호 $e \in E$ 에 가중치 $w_e \in \mathbb{Z}_+$ 가 주어져 있다. 마디의 부분 집합 $U_1, U_2 \subseteq V$ 가 $U_1 \cap U_2 = \emptyset, U_1 \cup U_2 = V$ 를 만족하도록 하면서 한 쪽 끝 마디는 U_1 , 다른 한 쪽 끝 마

디는 U_2 에 있는 호의 가중치 합을 최대화한다.

다음과 같은 특별한 형태의 그래프에서는 다항시간 안에 maximum cut 문제의 최적해를 찾을 수 있다.

- 평면 그래프 (Planar Graph) [1]
- 약한 이분 그래프 (Weakly Bipartite Graph) [3]

하지만 일반적인 그래프에 대해서 maximum cut 문제는 모든 호의 가중치가 1일 때에도 NP-hard이다 [2].

Goemans와 Williamson은 SDP(SemiDefinite Programming)을 이용하여 0.87856의 근사계수를 갖는 근사해법을 개발하였다 [8].

2.2 Minimum Multicut

문제 2.2.1 Minimum Multicut

비음의 가중치를 가지는 호로 구성된 무향 그래프 $G=(V,E)$ 에 k 개의 특별한 마디 쌍 $(s_1, t_1), \dots, (s_k, t_k)$ 가 존재한다. 이 문제의 목적은 모든 마디 쌍 (s_i, t_i) 를 분리시키면서, 가중치의 합 $\sum_{e \in F} w_e$ 를 최소화하는 호의 부분집합 F 를 찾는 것이다.

$k=1$ 일 때, minimum multicut 문제는 (s, t) -minimum cut 문제가 된다. 이 경우에는 다항시간 안에 해결할 수 있지만, $k \geq 3$ 이면 NP-hard이다 [7].

Minimum multicut 문제는 $O(\log k)$ 의 근사계수를 갖는 근사해법이 존재한다 [9].

2.3 Maximum Sparsest Cut

문제 2.3.1 Maximum Sparsest Cut

그래프 $G=(V,E)$ 에서 호 $e \in E$ 는 용량 $c_e \in R_+$ 를 갖는다. k 개의 마디 쌍 $\{(s_1, t_1), \dots, (s_k, t_k)\}$ 사이에는, $dem(i)$ 만큼의 수요가 정해져 있다. 이 그래프의 마디집합 $S \subseteq V$ 에 대해서 $I(S)$ 를 마디 쌍 $\{(s_1, t_1), \dots, (s_k, t_k)\}$ 중에서

$\delta(S) = \{(u, v) \in E \mid u \in S, v \notin S\}$ 에 속하는 호들로 인해 연결이 끊어지게 되는 것들의 집합이라 하자. 즉, $I(S) = \{i \mid |S \cap \{s_i, t_i\}| = 1\}$ 로 정의할 수 있다. 이 때 sparsest ratio $\rho(S)$ 를 다음과 같이 정의한다.

$$\rho(S) = \frac{\sum_{i \in I(S)} dem(i)}{\sum_{e \in \delta(S)} c_e}$$

이 문제의 목적은 $\rho(S)$ 를 최대화하는 마디집합 S 를 찾는 것이다.

Maximum sparsest cut 문제는 $\frac{1}{\rho(S)}$ 을 최소화하는 minimum sparsest cut 문제와 같다. Minimum sparsest cut 문제는 NP-hard이고 [6], $O(\sqrt{\log k})$ 의 근사계수를 갖는 근사해법이 존재하며 [11], 지금까지 알려진 가장 좋은 근사계수는 $O(\sqrt{\log k \log \log k})$ 이다 [12].

3. 문제 정의와 계산 복잡도

이 장에서는 논문에서 다루고자 하는 문제를 정의하고 문제의 계산 복잡도에 대해 언급한다.

3.1 문제 정의

논문에서 다루고자 하는 문제를 다음과 같이 정의한다.

문제 3.1.1 최대 가중치 합 그래프 마디 분리 문제

무향 그래프 $G=(V,E)$ 의 호 $e \in E$ 는 가중치 $w_e \in Z$ 를 가지고 있다. $\delta(U)$ 를 $\{e=(u, v) \in E \mid u \in U, v \in V \setminus U\}$ 즉, 한 쪽 끝은 마디집합 U 에, 다른 한 쪽 끝은 $V \setminus U$ 에 있는 호의 집합으로 정의한다. 그림 3.1에서 점선들이 이런 호를 의미한다. 이 문제의 목적은 각 마디집합 사이에 걸쳐 있는 호의 가중치 합을 최대화하는 마디의 부분집합들을 찾는 것이다. 즉, $\sum_i \sum_{e \in \delta(U_i)} w_e$ 를 최대화하는 마디집

합의

집합

$U = \{U_i \subseteq V \mid U_i \cap U_j = \emptyset (i \neq j), \bigcup_i U_i = V\}$ 를 찾는 문제이다.

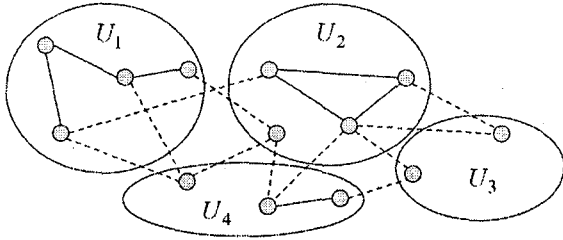


그림 3.1 최대 가중치 합 그래프 마디 분리 문제

3.2 계산 복잡도

다항시간 안에 문제 2.2.1을 문제 3.1.1으로 변환할 수 있다. 문제 2.2.1의 한 예에서 각 호의 가중치를 모두 음수로 만든다 ($w'_e = -w_e$). 각 마디 쌍 사이에 호를 연결하고 이것의 가중치를 M 으로 둔다. 만약 호가 이미 존재한다면 $M - w_e$ 로 둔다. M 이 충분히 큰 수라면 ($M > \sum_{e \in E} w_e$), 이 예에 대해서 문제 3.1.1의 최적해는 항상 모든 마디 쌍 (s_i, t_i) 를 분리하면서 선택된 호의 가중치 합을 최소화하여 문제 2.2.1의 최적해가 된다. 문제 2.2.1은 $k \geq 3$ 에서 NP-hard이기 때문에 [7], 양수의 가중치를 가지는 호가 3 개 이상인 최대 가중치 합 그래프 마디 분리 문제는 $P \neq NP$ 라는 가정하에서 NP-hard이다.

4. 근사 불가능성

이 장에서는 최대 가중치 합 그래프 마디 분리 문제에 대한 근사해법이 존재하지 않는다는 사실을 밝힌다.

4.1 관련된 문제들

문제 4.1.1 최적해의 목적함수 값이 음수가 될 수 있는 최대 가중치 합 마디 분리 문제

무향 그래프 $G = (V, E)$ 의 호 $e \in E$ 는

가중치 $w_e \in Z$ 를 가지고 있다. $\delta(U)$ 를 $\{e = (u, v) \in E \mid u \in U, v \in V \setminus U\}$ 즉, 한 쪽 끝은 마디집합 U 에, 다른 한 쪽 끝은 $V \setminus U$ 에 있는 호의 집합으로 정의한다. 이 문제의 목적은 각 마디집합 사이에 걸쳐 있는 호의 가중치 합을 최대화하는 마디의 부분집합들을 찾되, 마디집합 U_i 가 V 의 진부분집합이 되도록 하는 것이다. 즉, $\sum_i \sum_{e \in \delta(U_i)} w_e$ 를 최대화하는 마디집합의

집합

$U = \{U_i \subset V \mid U_i \cap U_j = \emptyset (i \neq j), \bigcup_i U_i = V\}$ 를 찾는 문제이다. 최대 가중치 합 그래프 마디 분리 문제는 $U = \{V\}$ 라는 간단한 해가 존재하여 최적해의 목적함수 값이 0보다 크거나 같다는 보장을 할 수 있지만, 이 문제에서는 최적해의 목적함수 값이 0보다 작을 수도 있다.

문제 4.1.2 Maximum sparsest multicut

무향 그래프 $G = (V, E)$ 의 호 $e \in E$ 는 용량 $c_e \in R_+$ 를 갖는다. k 개의 마디 쌍 $\{(s_1, t_1), \dots, (s_k, t_k)\}$ 사이에는 $dem(i)$ 만큼의 수요가 정해져 있다.

$\delta(S) = \{(u, v) \in E \mid u \in S, v \notin S\}$,
 $I(S) = \{i \mid S \cap \{s_i, t_i\} = 1\}$, 마디의 부분집합 S_i 에 대해서 $S = \{S_i \mid S_i \cap S_j = \emptyset (i \neq j), \bigcup_i S_i = V\}$ 라 할 때, sparsest ratio $\rho(S)$ 를 다음과 같이 정의한다.

$$\rho(S) = \frac{\sum_{S_j \in S} \sum_{i \in I(S_j)} dem(i)}{\sum_{S_j \in S} \sum_{e \in \delta(S_j)} c_e}$$

이 문제의 목적은 $\rho(S)$ 를 최대화하는 마디집합의 집합 S 를 찾는 것이다.

기본정리 4.1.3 문제 4.1.2는 NP -hard이다.

증명. 이것을 증명하기 위해서 [10]에서 언급한 다음 성질을 이용한다.

$$\max_i \frac{a_i}{b_i} \geq \frac{\sum_i a_i}{\sum_i b_i} \quad (a_i, b_i \in \mathbb{Z}, a_i \geq 0, b_i > 0)$$

이를 문제 4.1.2에 적용하면 다음과 같은 사실을 알 수 있다.

$$\max_j \rho(S_j) = \max_j \frac{\sum_{i \in I(S_j)} dem(i)}{\sum_{e \in \delta(S_j)} c_e} \geq \frac{\sum_{S_j \in S} \sum_{i \in I(S_j)} dem(i)}{\sum_{S_j \in S} \sum_{e \in \delta(S_j)} c_e} = \rho(S). \quad (4.1)$$

$\hat{S} = \{S, V \setminus S\}$ 라 하고, 문제 2.3.1과 문제 4.1.2의 목적함수에 각각 S 와 \hat{S} 을 대입해 보면 $\rho(S) = \rho(\hat{S})$ 임을 알 수 있다. 문제 4.1.2의 최적해 $S^* = \{S_1^*, \dots, S_c^*\}$ 에 대해, $\max_i S_i^* = S_k^*$ 라 하면, 식 4.1과 S^* 가 문제 4.1.2의 최적해라는 사실에 의해 $\rho(S_k^*) = \rho(S^*) = \rho(S^*)$ 이다. 문제 2.3.1의 최적해를 S^* 라 하면, $\rho(S^*) = \max \rho(S) = \max \rho(\hat{S}) = \rho(S_k^*) = \rho(S_k^*) = \rho(S^*)$ 이다. 따라서 문제 4.1.2의 최적해에서 문제 2.3.1의 최적해를 구할 수 있다. 문제 2.3.1은 NP -hard이므로 [6], 문제 4.1.2는 $P \neq NP$ 라는 가정 하에서 NP -hard이다. \square

4.2 근사 불가능성

기본정리 4.2.1 최대 가중치 합 그래프 마디 분리 문제에는 문제 입력 크기의 다항식으로 표현되는 근사계수를 가지는 어떠한 근사해법도 존재하지 않는다.

증명. 다음과 같은 결정 문제를 생각해 보자.

문제 4.2.2

$$\rho(S) = \frac{\sum_{S_j \in S} \sum_{i \in I(S_j)} dem(i)}{\sum_{S_j \in S} \sum_{e \in \delta(S_j)} c_e} > \lambda$$

인 S 가 존재하는가?

다항시간 안에 문제 4.2.2에 대해 대답을 할 수 있다면, $[\lambda_{\min}, \lambda_{\max}] = [0, \frac{\sum_i dem(i)}{\min_e c_e}]$ 구간 에 대한 이진탐색으로 $\max_S \rho(S)$ 를 다항시간 안에 찾을 수 있다. 그러나 기본정리 4.1.3에 의해서 문제 4.1.2는 NP -hard이므로 문제 4.2.2는 NP -complete이다.

문제 4.2.2를 다음과 같이 표현할 수 있다.

문제 4.2.3

$$\sum_{S_j \in S} \sum_{i \in I(S_j)} dem(i) - \lambda \sum_{S_j \in S} \sum_{e \in \delta(S_j)} c_e > 0$$

인 S 가 존재하는가?

문제 4.1.2의 예 $G = (V, E)$ 와, G 와 같은 구조를 가지는 그래프 $G' = (V, E)$ 를 생각하자. 그래프 G' 에서 마디 쌍 $\{(s_1, t_1), \dots, (s_k, t_k)\}$ 사이의 호에는 $w_e = dem(i) - \lambda c_e$ 로 놓고, 나머지 호에는 $w_e = -\lambda c_e$ 로 놓는다.

최대 가중치 합 그래프 마디 분리 문제에 대한 $p(n)$ -근사 알고리즘 ($p(n)$ 은 n 에 대한 다항함수)이 존재한다고 하자. 이 알고리즘으로 구한 해의 목적함수 값이 0보다 크다면, 이는 문제 4.1.1에 $\sum_j \sum_{e \in \delta(S_j)} w_e > 0$ 인 해가 존재한다는 의미이다. 반면, 이 알고리즘으로 구한 해의 목적함수 값이 0이라면, 이는 문제 4.1.1의 모든 가능해는 목적함수 값이 0보다 작거나 같다는 의미이다. 따라서 이 알고리즘은

$$\sum_j \sum_{e \in \delta(S_j)} w_e > 0 \quad , \quad \text{즉}$$

$$\sum_{S_j \in S} \sum_{i \in I(S_j)} dem(i) - \lambda \sum_{S_j \in S} \sum_{e \in \delta(S_j)} c_e > 0$$

인 그래프 G' 의 마디집합의 집합 $S = \{S_i \mid S_i \cap S_j = \emptyset (i \neq j), \bigcup_i S_i = V\}$ 가 존재하는지 다항시간 안에 알려줄 수 있다. 그러나 문제 4.2.2는 NP -complete이므로 $P \neq NP$ 라는 가정 하에서, 최대 가중치 합 그래프 마디 분리 문제에는 문제 입력 크기의 다항식으

로 표현되는 근사 계수를 가지는 어떠한 근사 해법도 존재하지 않는다.□

5. 극대면을 정의하는 부등식들

3장에서 최대 가중치 합 그래프 마디 분리 문제가 *NP-hard*임을 밝혔고, 4장에서 이 문제에 대한 근사해법이 존재하지 않는다는 것을 증명했다. 따라서 이 문제를 해결하는 데 있어 두 가지 접근 방법을 고려할 수 있다. 첫 번째 방법은 발견적 기법(heuristic)을 사용하여 해의 품질에 대한 보장은 하지 못하지만 문제 입력 크기의 다항시간 안에 결과를 내리는 접근 방법이다. 두 번째는 이 문제의 수리계획 모형을 만들고 그것을 풀어서 다항 시간 안에 문제를 해결하는 것을 보장하지 못하지만 최적해를 구하는 방법이다. 이 장에서는 후자의 접근 방법을 적용하는 데 있어 계산시간을 줄여 합리적인 시간 안에 문제를 해결하는 데 도움이 될 수 있는 추가 제약식들을 찾는 방법에 대해 언급한다.

다면체의 극대면은 다면체의 차원보다 한 차원 낮은 진면(proper faces)을 말한다 [13]. 가능해 집합의 극대면들은 현재 수리 모형의 제약식들이 만들어 내는 공간을 효과적으로 잘라낼 수 있는 추가 제약이다.

5.1 용어 및 관련 문제들

문제 2.1.1을 확장한 다음과 같은 문제를 생각해 보자.

문제 5.1.1 Maximum k -cut

그래프 $G=(V, E)$ 의 호 $e \in E$ 는 비음의 가중치를 가지고 있다. G 에서 $U_i \subseteq V, U_i \neq \emptyset, U_i \cap U_j = \emptyset (i \neq j), \bigcup_{i=1}^k U_i = V$ 를 만족하는 k 개의 마디집합 U_i 를 선택하여, 각 마디집합 사이에 걸쳐 있는 호 $e=uv (u \in U_i, v \in U_j (i \neq j))$ 의 가중치 합을 최대화한다.

문제 5.1.2 Maximum up-to k -cut

그래프 $G=(V, E)$ 의 호 $e \in E$ 는 음수의 가중치를 가질 수 있다. G 에서 $U_i \subseteq V, U_i \cap U_j = \emptyset (i \neq j), \bigcup_{i=1}^k U_i = V$ 를 만족하는 k 개의 마디집합 U_i 를 선택하여, 각 마디집합 사이에 걸쳐 있는 호 $e=uv (u \in U_i, v \in U_j (i \neq j))$ 의 가중치 합을 최대화한다.

문제 2.1.1은 maximum 2-cut 문제와 같다. 문제 5.1.2는 U_i 가 공집합이 될 수 있으므로, G 에서 최대 k 개 까지의 공집합이 아닌 마디 집합을 선택하여 각 마디 집합 사이에 걸쳐 있는 호의 가중치 합을 가장 크게 만드는 문제와 같은 문제가 된다. G 의 마디가 n 개라고 하면, 최대 가중치 합 그래프 마디 분리 문제는 maximum up-to n -cut 문제와 같다.

그래프 $G=(V, E)$ 의 마디집합 $U \subseteq V$ 에 대해서 $U, V \setminus U$ 사이에 걸쳐 있는 호의 집합을 $\delta(U)$ 라 하자. 그리고 G 에서 $F \subseteq E$ 에 대응하는 벡터 x^F 를 다음과 같이 정의한다.

$$x_e^F = \begin{cases} 1 & e \in F \\ 0 & e \in E \setminus F \end{cases}$$

그래프 $G=(V, E)$ 의 마디집합 $U_1, U_2 \subseteq V, U_1 \cap U_2 = \emptyset, U_1 \cup U_2 = V$ 를 만족하는 U_1, U_2 가 주어졌을 때, 마디집합 U_1, U_2 사이에 걸쳐 있는 호의 집합 $C(G) = \delta(U_1) \cup \delta(U_2)$ 를 G 의 절단면(cut)이라 하자. 그리고 $P_C(G)$ 를 G 의 절단면에 대응하는 벡터집합 $x^{C(G)}$ 의 볼록껍개(convex hull)로 정의하고 G 의 절단면 유계다면체(cut polytope)라 부르기로 하자.

이와 비슷하게, 그래프 $G=(V,E)$ 의 마디 집합

$U_i \subseteq V, U_i \cap U_j = \emptyset (i \neq j), \bigcup_{i=1}^k U_i = V$ 가 주어져 있을 때, 마디집합 사이에 걸쳐 있는 호의 집합 $C_k(G) = \bigcup_{i=1}^k \delta(U_i)$ 를 G 의 k -절단면(k -cut)이라 하자. 그리고 $P_{kc}(G)$ 를 그래프 G 의 k -절단면에 대응하는 벡터집합 $x^{C_k(G)}$ 의 볼록덮개로 정의하고 G 의 k -절단면 유계다면체(k -cut polytope)라 부르기로 하자.

호의 가중치 벡터 $w \geq 0$ 일 때, 다음 문제를 풀면 문제 2.1.1의 최적해를 구할 수 있다.

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e x_e \\ \text{s.t.} \quad & x \in P_c(G) \end{aligned}$$

문제 5.1.2에서 $w \geq 0$ 일 때, 최적해 $U^* = \{U_1^*, \dots, U_k^*\}$ 에 $U_i^* = \emptyset$ 인 i 가 있다고 하자. 공집합이 아닌 마디집합을 $V_i^* (i=1, \dots, k^*)$ 라 하고, 공집합인 마디집합을 $W_i^* (i=1, \dots, k-k^*)$ 라 하면, $U^* = \{V_1^*, \dots, V_{k^*}^*, W_1^*, \dots, W_{k-k^*}^*\}$ 로 다시 표현할 수 있다.

U^* 가 최적해이므로 V_i^* 에서 임의의 마디를 골라 W_i^* 에 집어 넣어도 목적함수 값을 증가시키지 못한다. 또한 호의 가중치가 비음이므로 목적함수 값을 감소시키지도 않는다. 따라서 k 개의 공집합이 아닌 마디집합으로 구성된 최적해를 구할 수 있다.

따라서 $w \geq 0$ 일 때, 다음 문제를 풀면 문제 5.1.1의 최적해를 구할 수 있다.

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e x_e \\ \text{s.t.} \quad & x \in P_{kc}(G) \end{aligned}$$

그래프 $G=(V,E)$ 의 호에 대응하는 변수 x 에 대한 부등식 $a^T x \leq \alpha$ 를 생각할 수 있다. 여기서 a_e 는 호 $e \in E$ 에 해당하는 변수 x_e 의

계수를 의미한다. 그리고 간결한 표기를 위해서 $x(F) (F \subseteq E)$ 를 $\sum_{e \in F} x_e$ 로 정의한다.

다면체 P 에 대해서 부등식 $a^T x \leq \alpha$ 가 $P \subseteq \{x \in R^m \mid a^T x \leq \alpha\}$ 를 만족한다면, $a^T x \leq \alpha$ 는 다면체 P 에 대해서 타당하다(valid)고 한다. 그리고 P 에 대해 타당한 부등식 $a^T x \leq \alpha$ 에 대해서 $P \cap \{a^T x = \alpha\}$ 안에 P 의 차원 수만큼의 의사선형독립(affinely independent)인 점들이 존재하면 $a^T x \leq \alpha$ 는 P 의 극대면을 정의하는 부등식(facet defining inequality)이라 부른다.

그래프 $G=(V,E)$ 에서 $2 \leq k \leq n$ 일 때, 문제 5.1.2의 가능해는 최대 가중치 합 그래프 마디 분리 문제의 가능해가 된다. 따라서 $P_{kc}(G) \subseteq P_{nc}(G) (2 \leq k \leq n)$ 이다. 그리고 $P_c(G) = P_{2c}(G)$ 가 $|E|$ 차원 다면체이므로 [4], $P_{nc}(G)$ 역시 $|E|$ 차원 다면체이다.

기본정리 5.1.3 그래프 $G=(V,E)$ 에서 $P_{kc}(G) (2 \leq k \leq n)$ 의 극대면을 정의하는 부등식 $a^T x \leq \alpha$ 가 $P_{nc}(G)$ 에 대해 타당하다면 이 부등식은 $P_{nc}(G)$ 의 극대면을 정의한다.

증명. $2 \leq k \leq n$ 이므로 $P_{kc}(G)$ 는 $|E|$ 차원 다면체이다. 그리고 $a^T x \leq \alpha$ 는 $P_{kc}(G)$ 의 극대면을 정의하므로 $a^T x = \alpha$ 를 만족시키는 $|E|$ 개의 의사선형독립인 벡터 $x \in P_{kc}(G) \subseteq P_{nc}(G)$ 가 존재한다. $P_{nc}(G)$ 역시 $|E|$ 차원 다면체이고, $a^T x \leq \alpha$ 가 $P_{nc}(G)$ 에 대해 타당하므로 $a^T x \leq \alpha$ 는 $P_{nc}(G)$ 의 극대면을 정의한다. □

기본정리 5.1.3의 내용을 간단히 표현해 보면 그림 5.1과 같다.

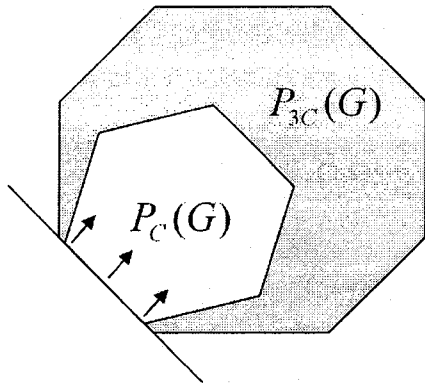


그림 5.1 두 다면체와 극대면 사이의 관계

5.2 극대면을 정의하는 부등식 찾기

기본정리 5.1.3에 의해서 절단면과 3-절단면의 극대면을 정의하는 부등식 중에서 n -절단면의 극대면을 정의하는 부등식들을 찾을 수 있다.

기본정리 5.2.1 그래프 $G=(V,E)$ 에서 $e \in E$ 가 삼각형 위에 있지 않으면, 식 다음 식은 $P_{nC}(G)$ 의 극대면을 정의한다.

$$0 \leq x_e \leq 1 \quad (5.1)$$

증명. 식 5.1은 $P_C(G)$ 의 극대면을 정의한다 [5]. 식 5.1은 $P_{nC}(G)$ 에 타당하고 $P_C(G) \subset P_{nC}$ 이므로 기본정리 5.1.3에 의해 $P_{nC}(G)$ 의 극대면을 정의한다.□

Barahona et al.은 $P_C(G)$ 의 극대면을 정의하는 부등식을 가지고 다음과 같은 방법으로 새로운 극대면을 정의하는 부등식을 만들 수 있다는 것을 증명했다.

기본정리 5.2.2 그래프 $G=(V,E)$ 에서 $a^T x \leq \alpha$ 가 $P_C(G)$ 의 극대면을 정의하는 부등식이라면 임의의 $v \in V$ 에 대해 다음 부등식도

$P_C(G)$ 의 극대면을 정의한다 [5].

$$\sum_{e \in \delta(v)} a_e x_e - \sum_{e \in \delta(v)} a_e x_e \leq \alpha - \sum_{e \in \delta(v)} a_e \quad (5.2)$$

Barahona et al.은 그래프 G 에서 임의의 마디 v_1, v_2, v_3 를 선택해 만들어지는 부분 그래프가 완전호 그래프일 때, 다음 부등식들이 $P_C(G)$ 의 극대면을 정의한다는 것을 보였다 [5].

$$\begin{aligned} x_{v_1 v_2} - x_{v_1 v_3} - x_{v_2 v_3} &\leq 0 \\ x_{v_2 v_3} - x_{v_1 v_2} - x_{v_1 v_3} &\leq 0 \quad (5.3) \\ x_{v_1 v_3} - x_{v_1 v_2} - x_{v_2 v_3} &\leq 0 \end{aligned}$$

기본정리 5.2.3 식 5.3은 $P_{nC}(G)$ 에 대해서 타당한 부등식이다.

증명. x_e 는 0 아니면 1 이므로 식 5.3을 만족하지 않으려면 다음 그림과 같은 경우밖에 없다. 그림 5.2에서 점선은 $x_e = 1$, 실선은 $x_e = 0$ 을 의미한다.

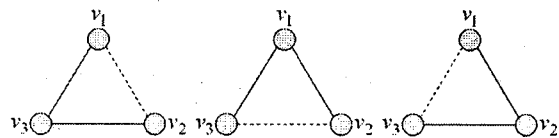


그림 5.2 식 5.3을 만족하지 않는 경우

이것을 만족하는 x 는 $P_{nC}(G)$ 에 속하지 않는다는 것을 알 수 있다.□

따름정리 5.2.4 식 5.3은 $P_{nC}(G)$ 의 극대면을 정의한다.

그래프 G 에서 임의의 마디 v_1, v_2, v_3, v_4, v_5 를 선택해 만들어지는 부분 그래프가 완전호 그래프일 때, 다음 부등식은 $P_C(G)$ 의 극대면을 정의한다 [5].

$$x_{v_1v_2} + x_{v_1v_3} + x_{v_1v_4} + x_{v_1v_5} + x_{v_2v_3} + x_{v_2v_4} + x_{v_2v_5} + x_{v_3v_4} + x_{v_3v_5} + x_{v_4v_5} \leq 6 \quad (5.4)$$

식 5.4에서 v_1, v_2 에 기본정리 5.2.2를 한번씩 적용하면 다음 부등식이 $P_C(G)$ 의 극대면을 정의한다는 사실을 알 수 있다.

$$x_{v_1v_2} - x_{v_1v_3} - x_{v_1v_4} - x_{v_1v_5} - x_{v_2v_3} - x_{v_2v_4} - x_{v_2v_5} + x_{v_3v_4} + x_{v_3v_5} + x_{v_4v_5} \leq 0 \quad (5.5)$$

식 5.5는 $P_{nC}(G)$ 에 대해 타당하므로 $P_{nC}(G)$ 의 극대면을 정의한다.

6. Branch-and-Cut 알고리즘

여러 종류의 정수계획 문제를 해결하는 데 있어, branch-and-cut 알고리즘을 사용하여 계산 시간을 줄이는 효과를 기대할 수 있다. Branch-and-cut 알고리즘은 정수계획 문제를 해결하는 방법 중에서 branch-and-bound와 cutting-plane 알고리즘을 혼합한 형태이다.

6.1 정수계획 모형

일반적인 그래프 $G=(V, E)$ 에 가중치가 0인 호를 추가해서 완전호 그래프 $G'=(V, E')$ 을 만들 수 있다. G' 에서 최대 가중치 합 그래프 마디 분리 문제의 최적해를 구하고 추가했던 호들을 제거하면 G 에서의 최적해를 구할 수 있다.

완전호 그래프 $G=(V, E)$ 에서 최대 가중치 합 그래프 마디 분리 문제를 다음과 같은 정수계획 문제로 모형화할 수 있다.

$$\max \sum_{e \in E} w_e x_e \quad (6.1)$$

$$\begin{aligned} \text{s.t. } & x_{v_1v_2} - x_{v_1v_3} - x_{v_2v_3} \leq 0 \\ & x_{v_1v_3} - x_{v_1v_2} - x_{v_2v_3} \leq 0 \quad \forall v_1, v_2, v_3 \in V \\ & x_{v_2v_3} - x_{v_1v_2} - x_{v_1v_3} \leq 0 \end{aligned} \quad (6.2)$$

$$x_e = \{0, 1\} \quad (6.3)$$

이 모형의 제약식인 식 6.2는 식 5.3과 같다. 기본정리 5.2.3에 의해서 식 6.2가 $P_{nC}(G)$ 에 대해 타당한 부등식임을 알 수 있다. 따라서 최대 가중치 합 그래프 마디 분리 문제의 모든 가능해 x 는 식 6.2를 만족한다.

기본정리 6.1.1 완전호 그래프 $G=(V, E)$ 에서 식 6.2를 만족하는 $x \in \{0, 1\}^{|E|}$ 는 최대 가중치 합 그래프 마디 분리 문제의 가능해이다.

증명. 식 6.2를 만족하면서 최대 가중치 합 그래프 마디 분리 문제의 가능해가 되지 않는 $x \in \{0, 1\}^{|E|}$ 가 존재한다고 하자. 그렇다면 완전호 그래프 $G=(V, E)$ 에서 $x_e = 1$ 인 호를 제거한 그래프 G' 에는 경로 $P = \{uv_1, uv_2, \dots, v_k w\}$ 가 있지만 둘 사이에 호는 없는 마디 u, w 가 존재한다. x 는 식 6.2를 만족한다고 하였으므로

$x_{uv_2} = x_{uv_3} = \dots = x_{uv_k} = 0$ 이다. 그리고 마디 u, v_k, w 에 대해서도 위의 부등식을 만족해야 하므로 $x_{uw} = 0$ 이다. 따라서 호 uw 가 그래프 G' 에 존재해야 한다. 그러나 이것은 호 uw 가 G' 에 존재하지 않는다는 가정에 모순이 된다. \square

기본정리 5.2.3과 6.1.1에 의해서 식 6.2를 이용해 완전호 그래프 $G=(V, E)$ 에서 최대 가중치 합 그래프 마디 분리 문제를 정수계획으로 모형화할 수 있다.

6.2 Branch-and-Cut 알고리즘

6.2.1 Branch-and-Cut 알고리즘

Branch-and-cut 알고리즘은 일반적인 branch-and-bound 알고리즘의 각 분기점에서, 정수 조건을 완화하여 구한 유리수 최적해와, 정수 가능해의 블록덮개로 정의된 가능해 집합을 분리하는 제약식을 찾아 추가하는 방법이다.

6.2.2 사용할 추가 제약식들

문제의 계산 시간을 줄이는 효과를 기대하기 위해서는 n -절단면에 대응하는 벡터들의 블록덮개인 P_{nc} 에 대해 타당한 부등식으로 현재의 유리수 최적해와 P_{nc} 를 분리할 수 있어야 한다. 현재의 유리수 최적해가 P_{nc} 의 극대면을 정의하는 부등식을 만족하지 않는다면 그 부등식은 branch-and-cut 알고리즘에서 사용할 추가 제약식의 역할을 할 수 있다. 여기서는 식 5.5와 같은 형태의 부등식들을 추가 제약식으로 사용하기로 한다.

6.2.3 유리수 최적해 분리 알고리즘

임의의 $x_1, x_2, x_3, x_4, x_5 \in V$ 를 선택하여 다음과 같은 $P_{nc}(G)$ 의 극대면을 정의하는 부등식을 만들 수 있다.

$$x_{v_1 v_2} - x_{v_1 v_3} - x_{v_1 v_4} - x_{v_1 v_5} - x_{v_2 v_3} - x_{v_2 v_4} - x_{v_2 v_5} + x_{v_3 v_4} + x_{v_3 v_5} + x_{v_4 v_5} \leq 0 \quad (6.4)$$

앞에서 제시한 정수계획 문제의 정수 조건인 식 6.3을 $0 \leq x \leq 1$ 로 완화한 선형계획 문제를 풀어서 나온 최적해를 \hat{x} 이라 하자.

기본정리 6.2.1 선형완화 최적해 \hat{x} 은 식 6.4에 대해

$$\hat{x}_{v_3 v_4} + \hat{x}_{v_3 v_5} + \hat{x}_{v_4 v_5} \leq \hat{x}_{v_1 v_3} + \hat{x}_{v_1 v_4} + \hat{x}_{v_1 v_5} + \hat{x}_{v_2 v_3} + \hat{x}_{v_2 v_4} + \hat{x}_{v_2 v_5}$$

를 만족한다.

증명. 식 6.4를 다음 그림과 같이 표현해 보자.

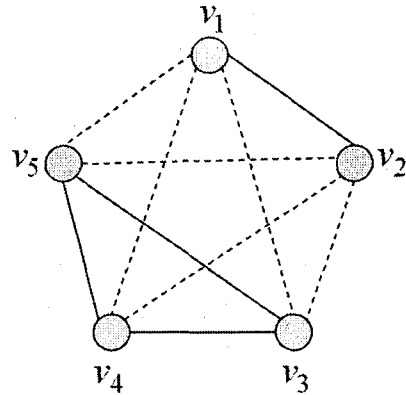


그림 6.1 마디가 5 개인 완전호 부분 그래프에서 만들 수 있는 부등식

그림 6.1에서 실선으로 표시한 호는 식 6.4에서 해당하는 변수의 계수가 1인 것이고, 점선으로 표시한 호는 식 6.4에서 해당하는 변수의 계수가 -1인 것이다. \hat{x} 은 식 6.2를 만족하기 때문에 그림 6.1 안의 삼각형 $v_1 v_3 v_4, v_2 v_3 v_4, v_1 v_3 v_5, v_2 v_3 v_5, v_1 v_4 v_5, v_2 v_4 v_5$ 에 대해서 각각 다음 부등식들이 성립한다.

$$\begin{aligned} \hat{x}_{v_3 v_4} &\leq \hat{x}_{v_1 v_3} + \hat{x}_{v_1 v_4} \\ \hat{x}_{v_3 v_4} &\leq \hat{x}_{v_2 v_3} + \hat{x}_{v_2 v_4} \\ \hat{x}_{v_3 v_5} &\leq \hat{x}_{v_1 v_3} + \hat{x}_{v_1 v_5} \\ \hat{x}_{v_3 v_5} &\leq \hat{x}_{v_2 v_3} + \hat{x}_{v_2 v_5} \\ \hat{x}_{v_4 v_5} &\leq \hat{x}_{v_1 v_4} + \hat{x}_{v_1 v_5} \\ \hat{x}_{v_4 v_5} &\leq \hat{x}_{v_2 v_4} + \hat{x}_{v_2 v_5} \end{aligned} \quad (6.5)$$

6.5에 있는 부등식들을 모두 더하면 된다. □

식 6.4를 $a^T x \leq 0$ 으로 표기하고 $|E| \times 1$ 벡터 b^T, c^T 를 다음과 같이 정의하자.

$$b_e = \begin{cases} 1 & a_e = 1, \text{ 단 } e \neq v_1 v_2 \\ 0 & \text{otherwise} \end{cases} \quad c_e = \begin{cases} 1 & a_e = -1 \\ 0 & \text{otherwise} \end{cases}$$

즉, 다음과 같이 표현할 수 있다.

$$a^T x = x_{v_1 v_2} + b^T x - c^T x$$

$\hat{x}_{v_1 v_2} = 0$ 이면 $a^T \hat{x} \leq 0$ 가 된다. 따라서 부등식 $a^T x \leq 0$ 는 \hat{x} 와 $P_{nc}(G)$ 를 분리할 수 없다. 또한 $0 \leq \hat{x}_{v_1 v_2} \leq 1$ 이기 때문에 \hat{x} 이 부등식 $a^T x \leq 0$ 에서 $b^T \hat{x} - c^T \hat{x} \leq -1$ 를 만족하게 되면 부등식 $a^T x \leq 0$ 는 \hat{x} 과 $P_{nc}(G)$ 를 분리할 수 없다.

기본정리 6.2.1에 의해서 $0 \leq b^T x \leq c^T x$ 이다. 그러므로 \hat{x} 에 대해 임의의 마디 v_1, v_2 가 정해졌을 때, $c^T \hat{x}$ 이 가장 작은 부등식 $a^T x \leq 0$ 는 식 5.5와 같은 형태의 다른 부등식들에 비해 $b^T \hat{x} - c^T \hat{x} (\leq 0)$ 값이 비교적 0에 가까울 것이라는 기대를 할 수 있다. 따라서 부등식 $a^T x \leq 0$ 는 \hat{x} 에 대해 $a^T \hat{x} = \hat{x}_{v_1 v_2} + b^T \hat{x} - c^T \hat{x} > 0$ 가 되어 \hat{x} 과 $P_{nc}(G)$ 를 분리할 수 있을 것이라는 기대를 할 수 있다.

지금까지 발견한 사실들을 활용하여, 식 5.5와 같은 형태의 부등식 중에서 유리수 최적해 x 와 P_{nc} 를 분리하는 것을 찾기 위해 다음 알고리즘을 사용한다.

알고리즘 6.2.2 유리수 최적해 분리 알고리즘
입력 완전호 그래프 $G=(V, E)$, 유리수 최적해 $x, S, U \subset V, F \subseteq E, i \in Z$.

출력 추가 제약식 $a^T x \leq 0$.

초기화 $F \leftarrow \{e | x_e > 0\}$.

단계 1 $f=(u, w) \in F$ 선택, $F \leftarrow F \setminus \{f\}$,
 $S \leftarrow \emptyset, U \leftarrow V \setminus \{u, w\}, i \leftarrow 0$.

단계 2 $S \leftarrow S \cup \{v \in U | \min(x_{uv} + x_{vw})\}$,
 $U \leftarrow U \setminus S, i \leftarrow i+1$.

$i < 3$ 이면 단계 2 로, $i = 3$ 이면 단계 3 으로.

단계 3 $x_f + \sum_{l, m \in S} x_{lm} - \sum_{v \in S} (x_{uv} + x_{vw}) > 0$
이면

$$x_f + \sum_{l, m \in S} x_{lm} - \sum_{v \in S} (x_{uv} + x_{vw}) \leq 0$$

출력하고 종료,

아니면 단계 4 로.

단계 4 $F \neq \emptyset$ 이면 단계 1로, 아니면 종료.

7. 실험 결과

이 장에서는 6 장에서 제시한 branch-and-cut 알고리즘을 사용하여 임의로 생성한 최대가중치 합 그래프 마디 분리 문제의 최적해를 구해 보았다. 실험 환경은 3.1GHz의 처리 속도를 가진 CPU를 탑재한 컴퓨터와 Microsoft Windows XP 운영체제이며, Microsoft Visual C++ 6.0을 이용해 구현하였다. Branch-and-cut 알고리즘의 매 분기마다 풀어야 하는 선형계획 문제는 ILOG사의 최적화 프로그램인 CPLEX 9.0의 라이브러리를 사용하여 해결하였다.

7.1 실험 조건

대상이 되는 그래프는 완전호 그래프이며 각 호에는 $[-99, 99]$ 구간의 정수 가중치를 임의로 할당하였다. 15 개와 20 개의 마디를 가지는 그래프에 대해서 각각 10 회씩 실험하였다. 6 장에서 제시한 추가 제약식이 문제의 계산 시간을 줄이는 데 도움이 되는 지를 알아보기 위해 동일한 문제에 대해 추가 제약식을 넣지 않을 때에 대해서도 계산 시간을 측정하였다.

7.2 실험 결과

마디의 수가 늘어날수록 계산 시간이 더 걸릴 것이라는 예상은 쉽게 확인할 수 있다. 마디 수가 같은 경우에도 계산 시간의 편차가 크다는 것 또한 확인할 수 있다. 이를 통해 호의 가중치 값이 계산 시간에 많은 영향을 준다고 추측할 수 있다.

7.2.1 마디의 수가 15 개일 때

마디의 수가 15 개일 때, 추가 제약식 투입 여부와 계산 시간을 정리한 결과는 표 7.1과 같다. 표에서 B&B는 추가 제약식을 투입하지 않은 경우, B&C는 추가 제약식을 투입한

경우를 의미한다.

실험 번호	1	2	3	4	5	6
B&B (sec)	0.17	0.14	0.06	1.09	2.45	2.28
B&C (sec)	0.14	0.08	0.05	0.52	1.16	0.2
실험 번호	7	8	9	10	평균	표준 편차
B&B (sec)	2.38	9.27	4.27	1.74	2.39	2.76
B&C (sec)	1.22	1.97	0.94	1.34	0.76	0.66

표 7.1 $n=15$ 인 경우 실험 결과

추가 제약식을 투입했을 때 계산 시간이 줄어들었음을 확인할 수 있다.

7.2.2 마디의 수가 20 개일 때

마디의 수가 20 개일 때, 추가 제약식 투입 여부와 계산 시간을 정리한 결과는 표 7.2 와 같다. 표에서 B&B는 추가 제약식을 투입하지 않은 경우, B&C는 추가 제약식을 투입한 경우를 의미한다.

실험 번호	1	2	3	4	5	6
B&B (sec)	1.69	6.06	32.66	30.01	97.17	49.92
B&C (sec)	1.61	4.89	17.06	33.25	74.95	41.73
실험 번호	7	8	9	10	평균	표준 편차
B&B (sec)	190.08	129.2	25.66	8.8	63	49.7
B&C (sec)	155.24	94.38	17.31	5.77	49.45	54.97

표 7.2 $n=20$ 인 경우 실험 결과

4 번 실험을 제외한 모든 경우에서 추가 제약식을 투입했을 때 계산 시간이 줄어들었음을 확인할 수 있다. 4 번 실험의 경우는 추가 제약식을 찾는 시간이 추가 제약식으로 인해 줄어든 시간보다 큰 경우라고 생각할 수 있다.

8. 결론

8.1 결론

이 논문에서는 호의 가중치가 음수가 될 수 있는 무향 그래프에서 마디집합들을 선택하여 각 마디집합 사이에 걸쳐 있는 호의 가중치 합을 최대화하는 최대 가중치 합 그래프 마디 분리 문제에 대해 연구하였다. 논문에서 얻은 결과는 다음과 같다.

- 최대 가중치 합 그래프 마디 분리 문제는 다항시간 안에 최적해를 찾을 수 없다고 믿고 있는, 이른바 NP -hard 문제이다.

- $P \neq NP$ 라는 가정 하에서, 이 문제에 대한 근사해법은 존재하지 않는다.

- 기존에 연구가 많이 진행된 절단면 유계다면체와 이 문제의 가능해 공간이 가지는 관계를 설명하고, 문제를 해결할 때 도움이 될 수 있는 몇 가지 추가 제약식들을 제시하였다.

- 발견한 추가 제약식을 활용한 branch-and-cut 알고리즘을 제시하고, 실험을 통해 추가 제약식이 계산 시간을 줄이는 데 도움이 된다는 사실을 확인하였다.

8.2 추후 연구 방향

이 논문에 이은 추후 연구 방향은 다음과 같다.

- 3 개 이상의 마디집합을 선택하여 만들어지는 절단면의 가능해 공간에 대한 연구가 필요하다.

• 극대면을 정의하는 부등식들을 더 많이 찾아야 한다. 이는 가능해 집합에 대한 연구뿐만 아니라, 문제의 계산시간을 줄이는 데에도 도움이 될 것이다.

• 계산 시간을 줄이기 위해, branch-and-cut 알고리즘에서 지금보다 더 빠른 유리수 최적해 분리 기법을 개발해야 한다.

참고문헌

- [1] F.O. Hadlock, Finding a maximum cut of planar graph in polynomial time, *SIAM J. on Comput.*, 4, pages 221-225, 1975.
- [2] M.R. Garey, D.S. Johnson, *Computers and Intractability: A guide to the theory of NP-completeness*, page 210, Freeman, 1979.
- [3] M. Grotschel, W.R. Pulleyblank, Weakly bipartite graphs and the Max-Cut Problem, *O.R. Letters*, 1 pages 23-27, 1982.
- [4] F. Barahona, M. Grotschel, A.R. Mahjoub, Facets of the Bipartite Subgraph Polytope., *Mathematics of Operations Research*, 10 pages 340-358, 1985.
- [5] F. Barahona, A.R. Mahjoub, On the Cut Polytope, *Mathematical Programming*, 36 pages 157-173, 1986.
- [6] D.W. Matula, F. Shahrokhi, Sparseset cuts and bottlenecks of graphs, *Discrete Applied Mathematics*, 27, 113-123, 1990.
- [7] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P.D. Seymour and M.Yannakakis, The complexity of multiway cuts, *SIAM J. on Comput.*, 23 pages 864-894, 1994.
- [8] M.X. Goemans and D.P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *J. of ACM*, 42, pages 1115-1145, 1995.
- [9] N. Garg, V.V. Vazirani and M. Yannakakis, Approximate max-flow min-(multi)cut theorems and their applications, *SIAM J. on Comput.*, 25 pages 235-251, 1996.
- [10] Edited by D.S. Hochbaum, *Approximation Algorithms for NP-hard Problems*, page 207, PWS, 1997.
- [11] S. Arora, S. Rao, and U. Vazirani, Expander flows, geometric embeddings, and graph partitionings, In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 222-231, 2004.
- [12] S. Arora, J. R. Lee, and A. Naor, Euclidean distortion and the sparsest cut, In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, 2005.
- [13] 홍성필, 선형계획법, page 75, 한경사, 2005.