韓國經營科學會誌
第26卷 第2號
2001年 6月

SDP의 개관 : 쌍대성, 계산복잡성 및 응용*

류춘호** · 명영수*** · 홍성필****

A Survey : SDP, its Duality, Complexity and Applications*

Choonho Ryu** · Young-Soo Myung*** · Sung-Pil Hong****

■ Abstract ■

SDP (Semidefinite Programming), as a sort of "cone-LP", optimizes a linear function over the intersection of an affine space and a cone that has the origin as its apex. SDP, however, has been developed in the process of searching for better solution methods for NP-hard combinatorial optimization problems. We surveyed the basic theories necessary to understand SDP researches. First, we examined SDP duality, comparing it to LP duality, which is essential for the interior point method. Second, we showed that SDP can be optimized from an interior solution in polynomial time with a desired error limit. Finally, we summarized several research papers that showed SDP can improve solution methods for some combinatorial optimization problems, and explained why SDP has become one of the most important research topics in optimization. We tried to integrate SDP theories, relatively diverse and complicated, to survey research papers with our own perspectives, and thus to help researchers to pursue their SDP researches in depth.

Keyword : SDP (Semidefinite Programming), combinatorial optimization, SDP duality, interior point method, survey

논문접수일 : 2000년 11월 15일 논문게재확정일 : 2001년 4월 12일
* 이 논문은 1999년 한국학술진흥재단의 학술연구비에 의하여 연구되었음(KRF-1999-042-C00134).
** 홍익대학교 경영학부
*** 단국대학교 경영·회계학부
**** 중앙대학교 상경학부

1. 배 경

90년대 중반 이후부터, SDP(Semidefinite Programming의 약자이다. 국내에는 아직 적절한 번역용어가 정착되지 않았으므로 본 논문에서는 'SDP'로 부르기로 한다)의 연구 결과들이 글자 그대로 쏟아져 나오게 된 동기는, 2000년 풀커슨상(Fulkerson Prize)을 받은 Goemans & Williamson의 최대절단면문제(Maximum cut problem)의 개선된 근사해법(approximation algorithm)연구[19]이다. 주어진 무방향 그래프의 절단면(cut)이란, 그 노드집합(node set)을 둘로 나눌 때, 두 집합에 걸치는 호(edge)들의 집합을 말한다. 최대절단면문제는 각 호에 가중치가 주어졌을 때, 가중치의 합이 최대가 되는 절단면([그림 6] 참조)을 구하는 대표적인 조합최적화문제 가운데 하나로, 풀기 어려운(NP-hard) 문제이다. 따라서 빠른 시간(다항시간) 안에 정확한 해를 구하는 알고리즘은 존재하지 않는 것으로 생각되는 문제이다. 기존의 최대절단면문제의 해법 중에 하나는 확률알고리즘(randomized algorithm)으로, 각 노드를 1/2의 확률로 양분된 노드집합에 속하도록 하는 것이다. 이 때 각 호가 절단면에 속할 확률은 1/2이 된다는 것을 증명할 수 있으며, 따라서 생성된 해의 가중치의 합의 기대값은 최소한 최적해의 1/2이 된다는 것을 보일 수 있다. 따라서 이러한 해법은 근사해법의 용어에 따르면 1/2-근사해법이 된다. 이 단순한 해법은 약 20년간 최대절단면문제의 대표적인 해법이었다. 물론 어느 정도의 개선은 있었으나 모든 기존 해법들의 근사치는 본질적으로 1/2을 넘지 못하였다. Goemans & Williamson은 기존에 알려진 최대절단면문제의 정수2차함수계획법 모형에서 출발하였다. 이 모형의 변수를 벡터로 완화하여 SDP를 얻었다. 이 때, SDP의 최적해를 확률적으로 활용하면, 가중치 합의 기대값이 최적해의 0.878배 이상인 절단면을 생성할 수 있음을 보였다. 따라서 새로운 해법은 0.878-근사해법으로 기존의 1/2-근사해법을 획기적으로 개선한 연구성과가 되었다. (자세한 설명은 본 논문의 5절에서 다룬다.)

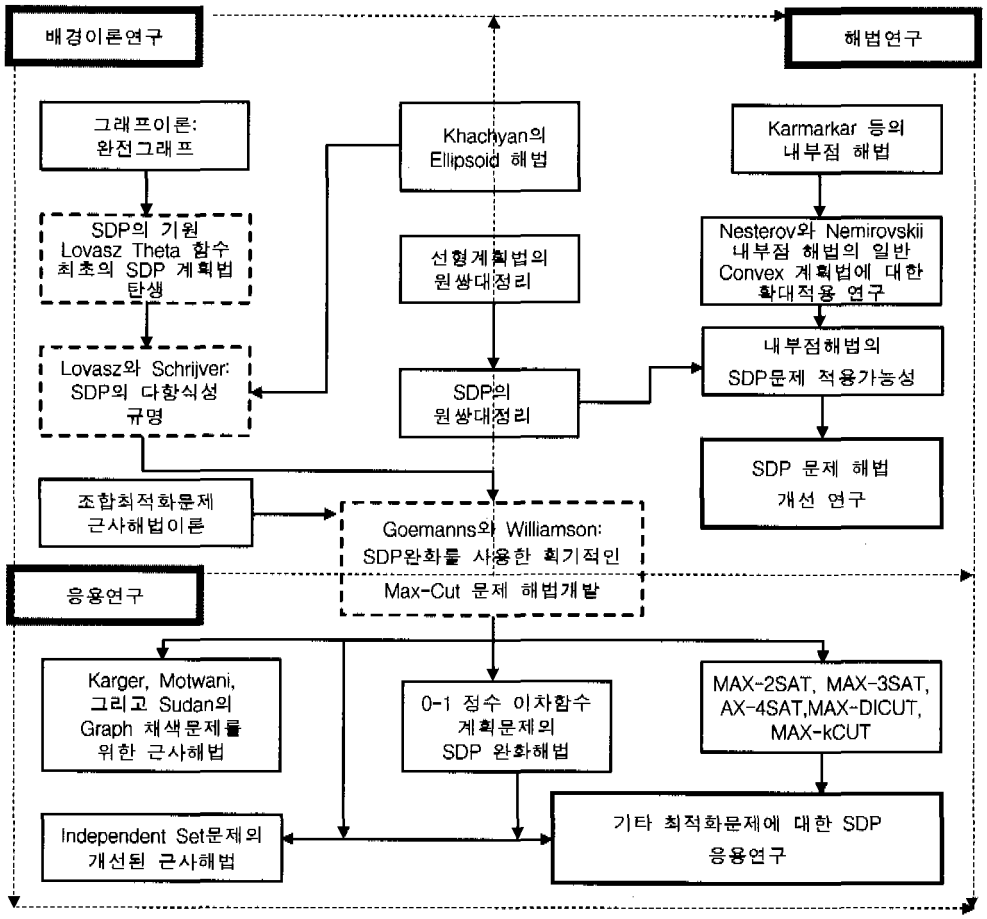
이렇게, SDP 연구를 본격적으로 촉발시킨 것은 Goemans & Williamson의 연구였지만, SDP의 다항식성과 같은 기본 이론을 정립하고, 풀기 어려운 조합최적화 문제의 적용가능성을 열어 보인 것은, 70년대 말부터 시작된 Lovasz등의 일련의 연구들([21, 22, 23, 30, 31])이다. Lovasz는 1979년 주어진 그래프의 최대안정집합(maximum stable node set) 크기의 상한을 구하기 위해, 그래프 노드에 벡터 변수를 대응시키는 새로운 발상을 제시하였는데, 이러한 벡터 변수들로 표현된 상한 제공 모형은 바로 최초의 SDP 문제가 되었다. 이어 1981년의 논문[21]에서는 주어진 볼록집합 위의 선형최적화문제(linear optimization over convex set)의 분리문제(separation problem)를 다항시간에 풀 수 있으면 이를 타원해법(Ellipsoid Method)의 서브루틴으로 사용하여 최적화문제를 다항시간에 풀 수 있음을 증명하였다. 앞에서 언급한 상한제공 SDP문제의 분리문제를 다항시간에 풀 수 있음을 증명하였으며, 이는 일반적인 SDP문제에도 적용되는 것으로 결국 일반적인 SDP의 다항식성을 규명한 논문이 되었다. 또한 완전그래프(perfect graph)의 경우, 상한제공 SDP의 최적값과 바로 최대안정집합(stable set)의 크기가 일치함을 보였다. 이러한 연구는 계속 발전되어 SDP는 상당히 광범위한 그래프들의 최대안정집합(maximum stable set)의 정확한 최적해를 구해내는 다항시간 해법을 제공할 수 있음이 밝혀졌다[22, 23, 31].

이렇게 SDP는 풀기 어려운 조합최적화문제들의 보다 나은 해법을 고안하기 위한 동기에서 탄생하였으며 현재에도 이러한 연구가 중요한 축을 이루고 있다. 또 하나의 중요한 축은 SDP 자체를 효율적으로 풀기 위한 해법 연구이다. SDP의 다항식성은 이미 앞에서 언급하였고, 본 논문에서 더 자세히 살펴보게 되겠지만, 타원해법을 사용하여 증명되었다. 그러나 이미 널리 알려진 바와 같이, 타원해법은 이론적인 가치는 높지만, 실용성

은 낮다. 따라서 SDP의 해법 연구는 주로 기존 선형계획의 내부점해법(interior point method)의 효율성을 SDP에 확장하려는 시도에 집중되어 있다. 내부점해법이 SDP에도 확장될 수 있음을 보인 연구는 크게 두 가지 방향으로 나눌 수 있다. 첫째는 "p-selfconcordant barrier"를 만들 수 있는 볼록집합(convex set) 위에서 선형함수를 극대화하는 볼록계획법(convex programming)은 모두 내부점해법으로 다항시간에 풀 수 있다는 Nesterov & Nemirovskii[35]의 연구이며, 분리문제를 다항시간에 풀 수 있는 문제는 다항시간에 최적화할 수 있다는 Grotschel, Lovasz, & Schrijver[21]의 연구와 비견되는 연구라고 하겠다. 이러한 일반적인 연구와는 상대적으로, 선형계획법을 위한 특정한 내부점해법을 SDP의 해법으로 확장시킨 구체적인 연구들을 들 수 있는데, Alizadeh[1]는 Ye[48]의 선형계획을 위한 내부점해법을 SDP을 위한 해법으로 확장하는 연구를 수행하였다. 이러한 확장은 SDP의 쌍대성에 기반을 두고 있다.

저자들의 견해에 따라, 이러한 SDP 중요연구들과 그 연관 관계를 편의적으로 정리하면 [그림 1]과 같이 나타낼 수 있겠다.

본 논문은 다음과 같이 구성되었다. 2절에서는 SDP의 정의와 몇 가지 특징들을 살핀다. 3절에서는 SDP의 쌍대성을 설명한다. 4절에서는 SDP의 계산적 복잡성을 규명한다. 첫째로 SDP의 최적화



[그림 1] SDP의 주요 연구들과 그 논리적 연관 관계

문제가 다항시간에 풀 수 있음을 논한다. 둘째로 계산복잡성 관점에서 볼 때 최적화문제와 가능성 문제(feasibility problem)가 동치인 선형계획법과는 달리 SDP의 가능성문제의 복잡성은 아직 규명되어 있지 않았다는 점을 지적한다. 5절에서는 SDP의 중요한 응용 연구로서 조합최적화에의 응용을 개관한다. 마지막으로 6절에서는 저자들의 견해로 중요하다고 생각하는 SDP의 연구 방향들을 지적하려고 한다.

SDP는 다양하면서도 깊이 있는 배경 이론의 이해를 요구하는, 최소한 저자들에게는, 상당히 난이도가 높은 주제라고 생각한다. 이 개관 논문의 목적 중의 하나는, 선택한 SDP의 이론들뿐 아니라, 이러한 배경 이론들도 함께 가능한 한 체계적으로 상술하여 넓은 층의 독자들이 SDP라는 주제에 접근하는데 도움이 되고자 하는 것이다. 예를 들어 부록에는 관련된 행렬, 다면체이론(polyhedral theory) 등을 정리하여 넣어 참고가 되도록 하였다. 그러나 상당한 기존의 이론들은 지면 한계로 인하여 직관적인 서술에 머물렀음을 밝힌다.

2. SDP의 정의

SDP는 다음과 같이 정의된다 :

$$\text{(SDP)} \quad \begin{array}{ll} \inf & c^T x \\ \text{s.t.} & x_1 A_1 + x_2 A_2 + \cdots + x_m A_m \geq B. \end{array}$$

여기서, inf란 최소극한값을 취한다는 의미로 min의 일반화라고 하겠다. 최적해의 목적 함수값이 유한할 때면 언제나 최소값이 존재하는 선형계획법과는 달리 SDP에서는 이러한 일반화가 필요하다는 점은 뒤에서 설명하겠다. 모든 A_i와 B는 $n \times n$ 대칭행렬이며 그 원소들은 주어진 상수이다. 어떤 $n \times n$ 대칭행렬 A에 대해 '$A \geq 0$'의 표기는 A가 PSD(Positive Semidefinite), 즉, 모든 n 차원 벡터 y에 대하여 $y^T A y \geq 0$가 성립함을 의미하며, $A \geq B$는 $A - B$가 PSD라는 의미이며 이 때, $\geq$는 대칭행렬 집합에 부분순서(partial order)를 정의하게 된다. 따라서 SDP의 해집합은, 모든 n 차원 벡터 y에 대하여 $y^T(x_1 A_1 + x_2 A_2 + \cdots + x_m A_m - B)y \geq 0$을 만족하는 m 차원 벡터 x의 집합으로 주어진다. 그러나, 하나의 고정된 y에 대하여, 앞의 조건을 만족하는 x의 집합은 닫힌(closed) 반공간(half-space)이며 따라서 볼록집합이 된다. 그러므로 SDP의 해집합은 닫힌 볼록집합들의 교집합이며 따라서 닫힌 볼록집합이 된다.

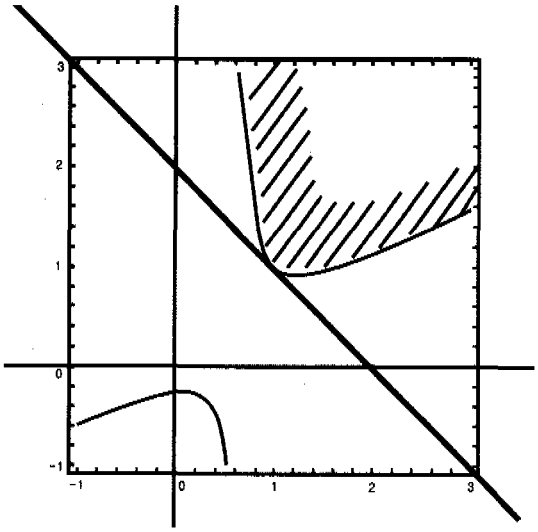
성질 2.1 : SDP의 가능해 집합은 닫힌 볼록집합이다. □

SDP가 선형계획을 특수한 경우로 포함하고 있음은, A_i들이 모두 대각행렬(diagonal matrix)일 때 SDP의 제약식들이 모두 선형부등식이 되는 것에서 쉽게 알 수 있다. 그러나 일반적인 SDP의 해집합은 앞에서 본 것처럼, 무수한 반공간의 교집합이다. 따라서 SDP의 해집합이 선형계획과는 달리 다면체(polyhedron)가 아니라는 것을 짐작할 수 있다. 실제로 다음의 간단한 예는 SDP의 해집합이 곡면을 포함할 수 있다는 것을 보여 준다.

예 2.2 : $x_1 \begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix} + x_2 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} -2 & 1 \\ 1 & 0 \end{bmatrix} \geq 0$라는 SDP 제약식을 간단히 표현하면 다음의 행렬이 PSD라는 조건이 된다 : $A(x) = \begin{bmatrix} x_1 + x_2 - 2 & -2x_1 + x_2 + 1 \\ -2x_1 + x_2 + 1 & x_1 + x_2 \end{bmatrix}$. 그런데 어떤 행렬이 PSD일 필요충분조건중의 하나는 주부분행렬(principal submatrix)의 행렬식(determinant)이 비음이라는 것이다(부록 A.1.3). 이 조건을 $A(x)$에 적용하면 위의 제약식은 다음의 세 개의 부등식으로 이루어진 조건과 동치임을 보일 수 있다 : $(x_1 + x_2 - 2)(x_1 + x_2) - (-2x_1 + x_2 + 1)^2 \geq 0$, $x_1 + x_2 - 2 \geq 0$, $x_1 + x_2 \geq 0$. 이 세 개의 조건을 모두 만족하는 영역은 [그림 2]의 빗금 친 부분으로 주어진다. □

위의 예에서는 SDP의 가능해집합이 하나의 곡선으로 정의되었지만, 행렬의 규모를 증가시키면 여러 개의 곡선과 직선으로 이루어진 예를 만들 수 있다.

SDP의 또 하나의 특징은 목적함수값의 하한(lower bound)이 있어 최소극한값(infimum)이 존재하여도 최소값(minimum)은 존재하지 않는, 그래서 그 극한값을 갖는 해가 존재하지 않는 "병적인(pathological)" 경우도 존재할 수 있다는 것이다. 이러한 특성 때문에 SDP의 경우, 최소값이 아닌 최소극한값을 추구하도록 문제를 일반화할 필요가 있다.



[그림 2] 예 2.2 SDP 제약식 해집합

예 2.3 : 다음의 SDP를 고려하자 :

$$\begin{aligned} \inf \quad & x_1 \\ \text{s.t.} \quad & x_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + x_2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ & = \begin{bmatrix} x_1 & 1 & 0 \\ 1 & x_2 & 0 \\ 0 & 0 & x_1 \end{bmatrix} \geq 0. \end{aligned}$$

예 2.2와 같은 방법으로 해집합은 다음의 부등식으로 주어짐을 알 수 있다 : $x_1 \geq 0$, $x_2 \geq 0$, $x_1 x_2 \geq 1$. 이 때, x_1은 0값에 무한히 접근할 수 있지만 0보다는 항상 커야한다. 따라서 위에 주어진 SDP의 최적해는 0을 극한값(infimum)으로 갖지만 그 극한값을 갖는 해는 존재하지 않는다는 것을 알 수 있다. □

이러한 예에서 볼 수 있듯이 SDP는 우리가 익숙한 선형계획의 일반화라는 관점에서 이해할 수 있지만, 한편으로는 그 해집합이 다면체가 아니기 때문에, 문제 구조의 조합적인 특성이 선형계획보다는 부족하여 특별한 고려가 필요하다. 예를 들어 해법 고안에 기반이 되는 쌍대성을 유용하게 사용하기 위해서는, 선형계획과는 달리, 비선형계획의 일반적인 맥락에서 요구되는 조건들을 SDP에서도 역시 가정할 필요가 있다. 이러한 특별한 조건을 사용하지 않는 쌍대문제를 정의할 수도 있지만[41] 그런 쌍대문제는 해법을 고안할 때는 그 유용성이 떨어지는 것으로 보인다. 실제로 지금까지 개발된 내부점해법들은 거의 모두, 부가적인 조건을 가정한 쌍대문제를 사용하였다. 이러한 경우, 다음의 절에서 보듯이 SDP도 선형계획과 비견할 만한 단순하고 아름다운 쌍대성을 지니게 된다.

3. SDP의 기하학적인 성질 : 쌍대성을 중심으로

2절에서 정의된 문제를 편의상 원문제(Primal SDP, PSDP)라고 부르자 :

$$\text{(PSDP)} \quad \begin{aligned} \inf \quad & c^T x \\ \text{s.t.} \quad & x_1 A_1 + x_2 A_2 + \cdots + x_m A_m \geq B. \end{aligned}$$

이에 대하여 쌍대문제(Dual SDP, DSDP)를 다음과 같이 정의한다 :

$$\text{(DSDP)} \quad \begin{aligned} \sup \quad & B \cdot Y \\ \text{s.t.} \quad & A_i \cdot Y = c_i, \ i = 1, \cdots m \\ & Y \geq 0. \end{aligned}$$

여기서, B와 A_i, 그리고 c는 원문제로부터 주어진 것이며, Y는 $n \times n$ 대칭행렬로서의 변수를 의미한다. 기호 · 는 같은 크기를 가진 두 정방행렬의 내적(inner product), 즉 두 행렬의 같은 위치에 있는 두 원소를 곱한 값들을 모두 합하는 연산자로서, 벡터 내적의 일반화라고 하겠다.

이러한 원-쌍대문제의 정의에서, 선형계획의 원-쌍대문제와 유사한 대칭성을 찾아낼 수 있다. 예를 들어 원문제가 최소화일 때는 쌍대문제에서는 최대화가, 원문제의 PSD 제약식에 쌍대문제에서는 하나의 대칭 행렬 변수가, 원문제가 최소화문제이며 제약식의 "부등호" 방향이 '$\geq$'이기 때문에 쌍대문제의 대칭행렬변수는 PSD 조건이 대응되는 것 등이다. 또한 원문제의 목적함수의 계수는 쌍대문제의 우변 상수가 된다. 단, 쌍대 제약식 함수를 만들 때, Y와 A_i들이 $n \times n$ 대칭행렬임에 따라, 그 곱도 행렬의 내적으로 정의된다는 점이 다를 뿐이다.

물론 (PSDP)와 (DSDP)는 원-쌍대문제 정의를 위해 편의적으로 선택된 형태이며, 표면적인 차이점을 가질 뿐, 서로 같은 문제이다. 즉, (PSDP)는 (DSDP)의 형태로, 또한 (DSDP)는 (PSDP)의 형태로 전환할 수 있다. (예를 들어 (PSDP)를 (DSDP)의 형태로 전환하기 위해서는 $x = x^+ - x^-$, $x^+ \geq 0, x^- \geq 0$으로 치환하고 $x_1^+ A_1 + \cdots + x_m^+ - x_1^- A_1 - \cdots - x_m^- A_m - B = Y$, $Y \geq 0$으로 제약식을 전환한 다음, 행렬등식에서 각 행렬 원소에 대한 등식관계를 대각행렬의 내적으로 표현하면 된다.) 그러나, 무엇보다도 중요한 것은, 위에서 정의된 원-쌍대문제는 선형계획의 원-쌍대문제가 갖는 쌍대성을 거의 그대로 갖는다는 점이다.

정리 3.1 (약쌍대정리) x와 Y를 한 쌍의 원-쌍대 가능해라고 하자. 그러면 $c^T x \geq B \cdot Y$이다.

증명 :

$$
\begin{aligned}
& c^T x - B \cdot Y \\
& = x_1 A_1 \cdot Y + x_2 A_2 \cdot Y + \cdots + x_m A_m \cdot Y - B \cdot Y \\
& = (x_1 A_1 + x_2 A_2 + \cdots + x_m A_m - B) \cdot Y.
\end{aligned}
$$

x와 Y가 가능해이므로, 마지막 식은 한쌍의 PSD 행렬 간의 내적이 된다. 그러나, 두 개의 PSD 행렬의 내적은 항상 0보다 같거나 크다.(부록 따름정리 A.2.7) 따라서 $c^T x - B \cdot Y \geq 0$이다. ■

앞에서 언급한 것처럼, 어떤 부가적인 조건을 가정하면 (PSDP)와 (DSDP)간에 강쌍대성(Strong Duality)을 얻을 수 있다. 증명은 선형계획을 포함한 일반적인 볼록계획법의 강쌍대성 정리를 유도하는 과정의 한 경우로 이해하면 되겠다. 예를 들어 선형계획의 강쌍대성은 흔히 문헌에서 보듯이 심플렉스 해법의 부산물로 증명할 수도 있지만, 일반적인 볼록계획법의 강쌍대성을 유도하는 과정에 따라 증명할 수 있다. 이러한 일반적인 볼록계획법의 강쌍대성 유도과정은 다음과 같이 요약할 수 있겠다.

분리초평면정리(Separation Theorem) $\Rightarrow$ Farkas정리 $\Leftrightarrow$ 강쌍대성

즉, Farkas정리(Farkas Lemma)는 강쌍대성과 동치이며, 따라서 일단 Farkas정리가 증명이 되면 강쌍대성의 증명은 용이한 과정이다. 이러한 Farkas 정리는 일반 볼록계획법(convex programming) 맥락에서도 유도될 수 있지만, 본 논문에서는 선형계획법과 SDP 쌍대성간의 유사성을 강조하기 위해, 선형계획의 Farkas정리와 강쌍대성의 동치관계를 살펴보자. 선형계획법의 Farkas정리는 여러 가지 형태로 기술할 수 있으며 그 중의 하나는 다음과 같다.

정리 3.2 : (선형계획의 Farkas정리) $A \in R^{m \times n}$, $c \in R^n$일 때, $A^T y = c$가 비음해 $y \geq 0$을 가질 필요충분조건은 $Ax \geq 0$을 만족하는 모든 x에 대하여 $c^T x \geq 0$이 성립하는 것이다. □

선형계획문제, $\min\{c^T x \mid Ax \geq b\}$와 그 쌍대

문제 $\max\{b^Ty \mid A^Ty = c,\ y \geq 0\}$을 생각하자. 정리 3.2의 뒤의 조건에서 $Ax \geq 0$을 만족하는 x는 제차해(homogeneous solution)라고 부르며, 원문제가 가능해집합을 가지면, 하나의 무한반직선(unbounded ray)에 해당한다. 따라서 만약 $c^Tx < 0$이면 원문제의 목적함수값을 무한히 감소시킬 수 있다는 것이 된다. 그러므로 정리의 조건, $c^Tx \geq 0$이 성립한다는 의미는 원문제가 비가능(infeasible)이던지, 아니면 유한한 최적해를 갖는다는 것을 알 수 있다. 이것이 쌍대문제의 가능성(feasibility)과 동치라는 것은 바로 잘 알려진 선형계획법의 강쌍대정리로부터 유도할 수 있음을 알 수 있다. 따라서 이것은 정리 3.2의 앞의 조건에 해당한다. 그러므로 선형계획의 강쌍대성은 Farkas정리를 의미한다. 또한 Farkas정리를 사용하여 강쌍대성을 유도할 수 있다. 즉, Farkas정리와 강쌍대성은 동치임을 보일 수 있다.

이러한 선형계획의 Farkas정리는 분리초평면정리(separation hyperplane theorem)를 사용하여 얻을 수 있다. 분리초평면정리란 모든 볼록해석학(convex analysis)의 기본이 되는 정리이다.

정리 3.3 : (분리초평면정리) 만약 C가 R^n에 포함된 닫힌 볼록 집합이고 z가 C에 속하지 않는 벡터라면, z와 C를 분리하는 초평면이 존재한다. 즉, C에 속한 모든 w에 대해 $w^Tx \leq \beta$이고 $z^Tx > \beta$인, 벡터 x와 실수 β가 존재한다.

증명 : 부록 보조정리 A.2.1 참조. ■

이 정리를 사용하여 선형계획의 Farkas정리(정리3.2)를 얻기 위해서는 $C = \{A^Ty : y \geq 0\}$으로 정의하여 분리초평면정리를 사용하면 된다. 즉, $A^Ty = c$가 비음해 $y \geq 0$를 가지지 않는다고 하자. 이것은 c가 $C = \{A^Ty : y \geq 0\}$에 속하지 않는다는 것을 의미한다. 따라서 이것은 분리초평면정리에 의하여, 어떤 x와 β가 존재하여 모든 $y \geq 0$ 대하여 $(A^Ty)^Tx = y^TAx \leq \beta$을 만족하며 동시에 $c^Tx > \beta$를 만족함과 동치이다. 그런데 C는 원점에 위치한 뿔(cone)이므로 이 때, β는 0이 될 수 밖에 없음을 쉽게 보일 수 있다. 또한 모든 $y \geq 0$에 대하여 $y^TAx \geq 0$이 성립한다는 것은 $Ax \geq 0$과 동치이다. 따라서, $A^Ty = c$가 비음해 $y \geq 0$를 가지지 않는다는 것은, $Ax \geq 0$인 어떤 x가 존재하여 $c^Tx < 0$을 만족한다는 것과 동치임을 보인 것이다. 이것은 바로 선형계획의 Farkas정리(의 대우)가 됨을 알 수 있다.

이렇게, 선형계획의 경우에는 아무런 부가적인 조건의 가정이 없이, 분리초평면정리로부터, 강쌍대정리와 동치인 Farkas정리를 얻을 수 있다. 이것은 $C = \{A^Ty : y \geq 0\}$가 다면체가 선형변환된 집합으로 닫힌 볼록집합이어서 분리초평면정리의 가정을 만족하기 때문이다.

일반적인 볼록계획법의 경우에도 본질적으로 같은 원리로, C를 적절히 정의하여 분리초평면정리로부터 Farkas정리를 얻는다. 그러나 부가적인 조건이 필요하다. 그 이유는 선형계획의 경우와 같이 C의 닫힘성(closedness)을 항상 보장할 수 없기 때문이다. 따라서 부가적인 조건이 필요하며 여러 가지가 가능하다([7]). Slater[44]는 일반적으로 볼록계획법의 제약식에서 얻어진 제차방정식이 상대내부해(relative interior solution)를 가진다고 가정하면, 닫힌 C를 적절히 정의하여, 분리초평면정리를 사용하여 강쌍대성을 얻을 수 있음을 보였다. 본 논문에서는 이 Slater의 조건을 SDP에 단순적용한 Alizadeh[1]의 조건을 채택한다.

보조정리 3.4 : (Slater 조건) (PSDP) 제약식의 제차방정식이 내부해를 가진다고 하자. 즉, $\sum_{i=1}^{m} x_iA_i > 0$이 해를 갖는다고 하면 $C = \{(A_1 \cdot Y, A_2 \cdot Y, \cdots, A_n \cdot Y) : Y \geq 0\}$는 닫힌 집합이 된다. □

논의의 초점을 유지하기 위해 증명은 부록(정리 A.3.3)으로 미룬다. 본문에서는 C가 닫힌 집합이 될 때, 어떻게 SDP의 Farkas정리가 성립하는지에 논의를 국한하기로 한다.

정리 3.5 : (SDP의 Farkas정리) 보조정리 3.4와 같이 Slater 조건이 성립하여 $C=\{(A_1 \cdot Y, A_2 \cdot Y, \cdots, A_n \cdot Y) : Y \geq 0\}$가 닫힌 집합이라고 하자. 그러면 (DSDP)가 가능해, $A_i \cdot Y = c_i$, $i=1,\cdots m$, $Y \geq 0$가 존재할 필요충분조건은 $\sum_{i=i}^{m} x_i A_i \geq 0$를 만족하는 모든 x에 대하여 $c^T x \geq 0$이 성립하는 것이다.

증명 : (충분조건) 대우를 증명하자. $A_i \cdot Y = c_i$, $i=1,\cdots m$를 만족하는 $Y \geq 0$가 존재하지 않는다고 하자. 이것은 c가 $C=\{(A_1 \cdot Y, A_2 \cdot Y, \cdots, A_n \cdot Y) : Y \geq 0\}$에 속하지 않음을 의미한다. 가정에 의하여, C는 볼록집합이며, 닫혀있으므로 c와 C 사이에 분리초평면이 존재한다 : 어떤 $x \in R^m$가 존재하여 모든 $Y \geq 0$에 대하여, $x^T(A_1 \cdot Y, A_2 \cdot Y, \cdots, A_n \cdot Y) = (\sum_{i=1}^{n} x_i A_i) \cdot Y \geq 0$이며 $c^T x < 0$이 성립한다. 어떤 행렬이 모든 PSD 행렬과의 내적이 비음이면, 그 행렬은 PSD이며 그 역도 성립한다. (부록 따름정리 A.2.7). 따라서 어떤 $x \in R^m$이 존재하여 $(\sum_{i=1}^{n} x_i A_i) \geq 0$이며 $c^T x < 0$이 성립한다. 따라서 대우가 증명된다.

(필요조건) 어떤 x가 $\sum_{i=i}^{m} x_i A_i \geq 0$를 만족한다고 하자. $c^T x \geq 0$임을 보여야 한다. 전제에 의하여 $A_i \cdot Y = c_i$, $i=1,\cdots m$를 만족하는 $Y \geq 0$가 존재한다. 따라서 $c^T x = x_1 A_1 \cdot Y + \cdots + x_m A_m \cdot Y = (\sum_{i=i}^{m} x_i A_i) \cdot Y$이다. 또한 $\sum_{i=i}^{m} x_i A_i \geq 0$ 와 $Y \geq 0$이므로, 두 PSD 행렬의 내적은 항상 비음이라는 성질에 의하여 $c^T x \geq 0$이 된다. ■

지금까지 우리는 SDP의 강쌍대성을 유도하는 과정 중에서 Slater 조건을 가정하여 다음을 완성한 것이 된다.

분리초평면정리(Separation Theorem) ⇒ Farkas정리

이제 다음의 고리를 증명하여 강쌍대성의 유도를 완성하여 보자.

Farkas정리 ⇔ 강쌍대성

정리 3.6 : (강쌍대정리, [1]) 정리 3.5와 같이 Slater 조건이 성립하여 SDP의 Farkas정리가 성립한다고 하자. 그러면 이와 동치로서 다음의 강쌍대성이 성립한다 : (PSDP)의 최소극한값을 p^*, (DSDP)의 최대극한값을 d^*라고 하면, 두 값은 일치한다 : $p^* = d^*$. 단, $\inf \emptyset = +\infty$, $\sup \emptyset = -\infty$라고 표기한다.

증명 : 우선 Slater 조건은 원문제(PSDP)가 가능해를 갖는다는 것을 의미한다 (부록 정리 A.3.2). 따라서, $p^* = +\infty$인 경우는 배제되고, 유한하거나, $-\infty$인 경우가 남는다. 먼저 $p^* = -\infty$인 경우를 살펴보자. 이 경우는 약쌍대성에 의하여 쌍대문제인 (DSDP)가 가능해를 가질 수 없다. 즉, 우리의 표기법에 의하여 $d^* = -\infty$이며 따라서 정리는 참임을 알 수 있다. 역으로, $d^* = -\infty$ 즉, 쌍대문제가 비가능일때는, 앞의 SDP의 Farkas정리에 의하여, $\sum_{i=i}^{m} x_i A_i \geq 0$를 만족하며 $c^T x < 0$인 어떤 x가 존재한다. 이것은 곧, 원문제의 목적함수값을 무한히 감소시킬 수 있다는 의미이며, $p^* = -\infty$이 되고 정리가 성립한다. 따라서 p^*와 d^* 모두 유한한 경우만 남는다.

약쌍대정리에 의하여, $p^* \geq d^*$임을 이미 알고 있다. $p^* > d^*$라고 가정해보자. 이것은 다음의 시스템이 비가능이라는 것을 의미한다:

$$\begin{aligned} B \cdot Y &= p^* \\ A_i \cdot Y &= c_i, \quad i=1,\cdots m \\ Y &\geq 0. \end{aligned}$$

여기에 Farkas정리를 적용하면, 어떤 실수 x_0와 벡터 $x=(x_1,\cdots,x_n)$가 존재하여, $x_0 B + \sum_{i=1}^{m} x_i A_i$

≥ 0을 만족하며, $p^* x_0 + c^T x < 0$라는 것을 의미한다. 이때 x_0가 0이면, 앞에서와 같이 쌍대문제가 비가능이 되어 d^*가 유한하다는 가정에 맞지 않는다. 만약 x_0가 양수이면, $y=(1/x_0)x$로 치환하여 $\sum_{i=1}^{m} y_i A_i \geq -B$, $c^T y < -p^*$를 얻게된다. 따라서 $c^T y < -p^* - \varepsilon$인 어떤 $\varepsilon > 0$을 잡을 수 있다.

한편 p^*가 원문제의 최소극한값이므로 $\sum_{i=1}^{m} z_i A_i \geq B$이며 $c^T z < p^* + \varepsilon$인 가능해 z가 존재한다. 따라서 $\sum_{i=1}^{m} (y_i + z_i) A_i \geq 0$, $c^T(y+z) < 0$이 성립하여, 쌍대문제의 비가능성을 의미하게되어 모순이 된다. 따라서 x_0는 음수가 될 수밖에 없으며, 이 때 $y=(-1/x_0)x$로 치환하면, $\sum_{i=i}^{m} y_i A_i \geq B$, $c^T y < p^*$를 얻게된다. 그러나, 이 것은 p^*가 원문제의 최소극한값이라는 것에 모순된다. 따라서 모든 경우에 모순이 생기며, 이것은 $p^* > d^*$ 가정이 거짓이라는 것을 의미한다. 즉 $p^* \leq d^*$이며, 약쌍대정리, $p^* \geq d^*$와 결합하면 $p^* = d^*$가 되어 증명이 끝난다. ■

$X = \sum_{i=i}^{m} x_i A_i - B$로 표기하자. 그러면 정리 3.1의 증명과정에서 보았듯이, $c^T x - B \cdot Y = X \cdot Y$로 주어진다. 따라서 한 쌍의 원-쌍대 가능해가 최적해일 필요충분조건은 $X \cdot Y = 0$이다. 그런데, 두 개의 PSD 행렬의 내적이 0일 필요충분조건은 두 행렬의 곱이 0이다(부록 따름정리 A.2.8). 따라서 다음과 같이 선형계획의 상보여유정리(complementary slackness theorem)를 얻을 수 있다.

따름정리 3.7 : 한 쌍의 원-쌍대 가능해가 최적해일 필요충분조건은 $XY = 0$이다. □

본 논문에서는 논의를 간편히 하기 위해, 제약식의 제차방정식이 내부해를 가진다는, 비교적 강한 Slater 조건을 사용하여 강쌍대정리를 유도하였다. 제약식의 제차방정식이 내부해가 존재하면 제약식자체도 내부가능해를 가진다. 그러나 그 역은 일반적으로 성립하지 않는다. 이러한 의미에서, 보다 약한 조건인 내부가능해의 존재를 사용하여도, 과정은 다소 길어지나, 동일한 원리에 의하여, 다음과 같이 좀더 구체적인 강쌍대성을 증명할 수 있다.

정리 3.8 : (i) 원문제가 내부가능해, 즉 $\sum_{i=i}^{m} x_i A_i > B$를 만족하는 해를 가진다고 하자. 그러면 $p^* = d^*$가 되며, 이러한 목적함수값을 가지는 쌍대최적해가 존재한다. (ii) 쌍대문제가 상대내부가능해(relative interior feasible solution), 즉 $A_i \cdot Y = c_i$, $i=1,\cdots m, Y > 0$를 만족하는 해를 가진다고 하자. 그러면 $p^* = d^*$가 성립하며, 이러한 목적함수값을 가지는 원문제의 최적해가 존재한다. □

정리 3.8을 증명하는 대신, 실제로 Slater류의 조건들이 원-쌍대간격(duality gap)이 0이 되게 하거나, 더 나아가 해가 존재하도록 하는데 필수적이라는 것을 보여주는 몇 가지 예를 들어보자.

예 3.9 : ($p^* > d^*$)

$$\begin{aligned} \inf \quad & x_1 \\ \text{s.t.} \quad & x_1 \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + x_2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ & = \begin{bmatrix} 0 & x_1 & 0 \\ x_1 & x_2 & 0 \\ 0 & 0 & x_1+1 \end{bmatrix} \geq 0. \end{aligned}$$

위의 문제의 해집합은 항상 $x_1 = 0$이라는 것을 알 수 있다. 따라서 $p^* = 0$이 된다. 쌍대문제는 다음과 같이 주어진다:

$$\begin{aligned} \sup \quad & -y_{33} \\ \text{s.t.} \quad & y_{12} + y_{21} + y_{33} = 1 \\ & y_{22} = 0 \\ & Y \geq 0. \end{aligned}$$

그런데, 일반적으로 PSD 행렬에서 대각선에 위치한 원소가 0이면, 같은 행과 열의 원소는 모두

0이 되므로, 쌍대문제의 가능해는 다음과 같은 형태가 됨을 알 수 있다 :

$$Y=\begin{bmatrix} a & 0 & b \\ 0 & 0 & 0 \\ b & 0 & 1 \end{bmatrix} \text{ (단, } a \geq b^2).$$

따라서, $d^*=-1$이 되어 쌍대간격이 존재한다. 실제로 위에서 원문제의 내부해나, 쌍대문제 상대내부해가 모두 존재하지 않는다는 것을 알 수 있다. □

예 3.10 : ($p^*=d^*$, 원문제는 최적해가 존재하지 않음, 쌍대문제는 최적해 존재함.)

$$\begin{aligned} \inf \quad & x_1 \\ \text{s.t.} \quad & x_1\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + x_2\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ & = \begin{bmatrix} x_1 & 1 & 0 \\ 1 & x_2 & 0 \\ 0 & 0 & x_1 \end{bmatrix} \geq 0. \end{aligned}$$

예 2.3과 동일한 문제이다. $p^*=0$이었지만 최적해는 존재하지 않았다. 쌍대문제는 다음과 같다 :

$$\begin{aligned} \sup \quad & -y_{12}-y_{21} \\ \text{s.t.} \quad & y_{11}+y_{33} = 1 \\ & y_{22} = 0 \\ & Y \geq 0. \end{aligned}$$

앞의 예제와 같은 이유로, 모든 쌍대문제의 가능해는 다음과 같은 형태가 됨을 알 수 있다 :

$$Y=\begin{bmatrix} a & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1-a \end{bmatrix} \text{ (단, } 0<a<1).$$

따라서, $d^*=0$이 되어 쌍대간격이 0이다. 실제로 예 2.3에서 보았듯이 원문제가 내부해를 가지기 때문에, 이렇게 쌍대간격이 존재할 수 없는 것이다. 그러나 쌍대문제의 상대내부해가, 즉 PD (positive definite)인 Y가 존재하지 않기 때문에 원문제에 $p^*=0$값을 갖는 최적해가 보장되지 않음을 알 수 있다. □

4. SDP의 복잡성

4.1 가능성문제(feasibility problem)

넓은 의미에서 하나의 최적화문제를 푼다는 것은 먼저 문제의 제약식을 모두 만족하는 가능해가 존재하는가를 판단하는 가능성문제(feasibility problem)를 포함한다. 우리가 익숙한 선형계획의 경우 가능성문제를 일단계문제(phase I problem)라는 최적화문제로 풀 수 있다. (일단계문제의 가능해는 인공변수들을 기저변수로 하는 기저가능해로 손쉽게 주어진다.) 또한 최적화문제는 원-쌍대 관계를 이용하여 가능성문제로 변환할 수 있다.

그러나, 일반적인 경우, 가능성문제와 최적화문제의 동등성은 성립하지 않으며, 최적해를 구하기 전에 가능해를 하나 확보하는 문제가 선결되어야 한다. 논의를 위하여, 가능성문제를 다음과 같이, "예" 또는 "아니오"라고 답하는 결정문제(decision problem) 형태로 국한하자.

정의 4.1 : (가능성문제) 문제가 가능해(feasible solution)를 가지는가? □

SDP의 경우 제한적인 의미의 최적화는 다항시간 안에 풀 수 있다 : 하나의 내부가능해와 가능해 집합의 상한이 주어진 경우는, 다항시간 안에 (절대오차 범위의) 최적해를 구할 수 있다. 그러나, 가능성문제를 다항시간에 답할 수 있는지는 아직 규명되어 있지 않다. 더 근본적으로 SDP가 NP라는 문제의 범주에 속하는지도 알려져 있지 않다.

NP문제란 직관적으로 말해, 그 답이 "예"일 때, 다항시간 안에 그 사실을 확인하게 해주는 증거(certificate)가 존재하는 결정문제를 총칭하며([15], [37]), 우리가 흔히 취급하는 선형 또는 조합최적화 문제는 대부분 NP에 속한다. 예를 들어 어떤 선형계획문제가 가능하면, 문제의 입력길이가 L일 때, 절대값이 $O(2^L)$인 기저해(따라서 유리수해)가 존재하며, 제약식에 대입하여 그 가능성을 확인해 볼

수 있어서 이것이 증거가 된다.

그러나, SDP의 경우 문제가 가능한 경우에도, 해 자체의 입력길이(input coding length) 가 문제 입력길이의 지수(exponential)함수가 될 정도로 큰 해만 존재하는 예를 쉽게 만들 수 있다.

예 4.1 : [1] 다음 문제의 해가 $x_i = 2^{2^{i-1}}$로 주어지는 것을 알 수 있다:

$$\begin{array}{ll} \inf & x_n \\ \text{s.t.} & x_1 \geq 2 \\ & \begin{bmatrix} x_i & x_{i-1} \\ x_{i-1} & 1 \end{bmatrix} \succeq 0, \ i = 1, \cdots, n. \end{array}$$

따라서, SDP가 NP의 범주에 속하는지를 증명하기 위해서는 (물론 NP에 속하지 않음을 증명할 수도 있지만), 선형계획과는 달리, 가능해를 증거로 사용할 수는 없다는 것을 말해준다. 따라서, 가능해가 아닌 다른 간접적인 증거를 사용하는 방법이 추구되어야 할 것이며 이러한 방법이 아직 알려지지 않았다.

SDP의 가능성문제라는 결정문제가 아직 NP의 범주에 속하는지 규명되지 않았다는 사실은, 이 문제를 다항시간에 푸는 알고리듬이 존재할 가능성이 희박하다는 부정적인 쪽으로만 해석되기 쉽다. 그러나, Ramana[41]는 Slater류의 조건을 사용하지 않는, 주어진 SDP 원문제에 대한 쌍대문제를 개발하는데 성공하였다. 이러한 쌍대문제를 (EDSDP) (exact dual SDP)라고 부르며 다음과 같은 성질을 가지고 있다[41].

첫째, (EDSDP)는 원문제 문제입력길이의 다항식들로 주어지는 변수 개수와 계수들의 입력크기를 가지고 있다. 즉, (EDSDP)의 문제입력길이는 원문제의 문제입력길이의 다항식함수가 된다. 둘째, 원문제(최소극한값문제)와 (EDSDP)(최대극한값문제) 간에는 약쌍대정리가 성립한다. 즉, 모든 원-쌍대 가능해 쌍에 대하여, 원문제의 목적함수값은 (EDSDP)의 목적함수값보다 같거나 크다. 셋째, 원문제가 가능해를 가지며, 최적목적함수값이 유한하면 (EDSDP)도 그러하며, 역도 성립한다. 그리고 양 문제의 최적목적함수값들은 일치한다.

이러한 사실은 SDP가 만일 NP의 범주에 속한다면, co-NP에도 속한다는 의미가 된다. (증명은 선형계획이 쌍대정리에 따라 NP와 co-NP에 모두 속한다는 잘 알려진 증명과 유사하므로 생략한다. 예를 들어 [15]를 참조할 수 있다.) 이러한 이유로, SDP의 가능성문제는, 학계에서 그렇게 믿고 있는 것처럼 NP와 co-NP가 일치하지 않는 한, NP-complete 문제는 아니라는 사실을 유도할 수 있게 된다. 따라서 SDP문제는 NP의 범주에서 제외될 정도로 아주 어려운 문제이거나, 쉽게 풀 수 있는 문제라는 추측을 할 수 있게 된다.

참고 4.2 : 계산모형(computation model)을 튜링기계모형(Turing machine model)이 아닌, 실수모형(real number model)[9]으로 확장하면, 어쩌면 당연한 결과겠지만, [41]에서 증명한 것과 같이 SDP가 NP의 범주에 속한다는 것을 증명할 수 있다. 실수모형이란, 간단히 말하면, 실수(real number)를 다룰 수 있는 계산모형을 의미하며, 따라서 주어진 수가 0-1 비트로 표현되는 것이 아니라, "다른 방법"으로 저장되고 연산될 수 있음을 가정하는 모형이다. 무리수 같은 실수들을 실제로 여러 가지 수단으로 다루고 있는 현실을 반영한 확장된 모형이라고 하겠다. 따라서 주어진 수들을 0-1 비트로 코딩할 때, 필요한 비트 수를 나타내는 입력크기는 문제입력크기에 포함시키지 않기 때문에, 예 4.1 등에서 언급한 문제들이 발생하지 않는다. 따라서, 이러한 확장된 모형을 가정한다면, 가능해 자체를 사용하여, 다항시간에 제시할 수 있는 가능성의 증거를 만들 수 있다. 이는 주어진 가능해, $\bar{x}$를 $A(x) = \sum_{i=1}^{m} x_i A_i - B$에 대입하여 가우스소거법으로 $A(\bar{x})$가 PSD인지를 확인할 수 있기 때문이다. 이점에 대해서는 다음 절에 자세히 언급한다. □

4.2 볼록집합 위에서의 선형최적화

우리는 앞 절에서 보편적인 튜링기계에 기반을

둔 복잡성이론에 의하면, 주어진 SDP문제가 가능한지를 "예" 또는 "아니오"로 대답하는 결정문제가, 선형계획과 같이 흔히 우리가 다루는 최적화문제들처럼 NP의 범주에 속하는지 아직 규명되지 않았음을 보았다. 따라서 가능해를 실제로 하나 구해내는 것은 더욱 어려운 문제가 될 것이다. 따라서 최적화문제의 의미를 넓게 보아서 가능성문제까지를 포함시킨다면, SDP의 최적화문제는 엄밀하게는 아직 그 복잡성이 규명되지 않았다고 봐야 할 것이다. 그러나, 내부 가능해 존재성과 같은 몇 가지 부가적인 조건이 만족되는 경우, SDP의 원하는 절대오차 범위의 최적해를 다항시간에 구할 수 있다. 따라서 본 절에서는 최적화문제를 좁은 의미로, 즉 최소한 내부 가능해가 주어진 경우로 국한시키기로 한다. 이러한 논의를 좀더 일반적인 맥락에서 진행하기 위해 우선 다음과 같은 일반적인 선형최적화문제를 고려하자:

$$\text{(P)} \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & x \in K. \end{array}$$

단, K는 닫힌 볼록집합이다.

예 4.1 : $x \in R^n$이고 K가 처음부터 m개의 반공간(halfspace), $a_i^T x \geq b_i (\, i = 1 \cdots m)$으로 주어지면, (P)는 선형계획문제가 된다. □

예 4.2 : 하나의 연결된 무향그래프 $G=(V,E)$가 주어졌을 때, 다음의 시스템을 만족하는 $x \in R^{|V|}$의 집합, 유한다면체(polytope) P를 생각하자:

$$\begin{aligned} x_u + x_v &\leq 1, \ \forall (u,v) \in E \\ x_v &\geq 0, \ \forall v \in V. \end{aligned}$$

x가 만일 P에 속하는 $0-1$ 벡터라면, $x_v = 1$을 만족하는 노드 v의 집합 S는, G에서 서로 이웃하지(adjacent) 않는 노드들의 집합인 안정집합(stable set)이 된다. P의 $0-1$ 벡터들을 모두 포함하는 가장 작은 볼록집합을 K라고 하고 $c = [1, 1, \cdots, 1]^T \in R^{|V|}$로 정의하자. 그러면 (P)는 가장 큰 안정집합을 구하는 문제 최대안정집합문제(maximum stable set)가 된다. □

예 4.3 : 하나의 연결된 무향그래프 $G=(V,E)$가 주어졌을 때, 다음의 시스템을 만족하는 $x \in R^{|E|}$의 집합, 유한다면체(polytope) P를 생각하자:

$$\begin{aligned} \sum_{e \in \delta(v)} x_e &\leq 1, \ \forall v \in V \\ x_e &\geq 0, \ \forall e \in E. \end{aligned}$$

단, $\delta(v)$는 노드 v에 달려있는 호의 집합을 의미한다. x가 만일 P에 속하는 $0-1$ 벡터라면, $x_e = 1$을 만족하는 호 e의 집합 M은, G의 (노드) 짝짓기(matching)가 된다는 것을 알 수 있다. 예 4.2에서 처럼, P의 $0-1$ 벡터들을 모두 포함하는 가장 작은 볼록집합을 K, 그리고 $c = [1, 1, \cdots, 1]^T \in R^{|E|}$로 놓자. 그러면 (P)는 최대짝짓기문제(maximum matching problem)가 된다. 그런데 Edmonds[12]는, K가 다음의 선형 시스템으로 나타낼 수 있음을 증명하였다 :

$$\begin{aligned} &\sum_{e \in \delta(v)} x_e \leq 1, \ \forall v \in V, \\ &\sum_{e \in E(S)} x_e \leq (|S|-1)/2, \ \forall |S| \text{가 홀수인 } S \subseteq V \\ &\qquad x_e \geq 0, \ \forall e \in E. \end{aligned}$$

단, $E(S)$는 집합 S에 양 끝 노드를 가지는 호의 집합을 의미한다. □

예 4.4 : $x \in R^m$이고 K가 $x_1 A_1 + x_2 A_2 + \cdots + x_m A_m \geq B$로 정의되는 볼록집합인 경우, (P)는 (PSDP)가 됨을 알 수 있다. □

위의 네 가지 예들은 모두 볼록집합 위에서 선형함수를 최적화하는 문제들임을 알 수 있다. 이 중에서 예 4.1의 선형계획은 다항시간 안에 풀 수 있는 문제이다. 예 4.2의 최대안정 집합문제가 NP-hard 문제임은 잘 알려져 있다. 이 경우, K는 유한다면체이기 때문에, 선형부등식으로 주어지는 반공간들의 교집합으로 표시할 수 있으며, 따라서

예 4.2는 선형계획문제가 된다. 그러나 이러한 반공간들이 모두 알려져 있지 않을 뿐더러, 필요한 반공간의 개수는 문제 입력크기의 지수함수로 주어진다는 어려움이 있어 실제로 선형계획문제로 예 4.2의 문제를 푼다는 것은 불가능한 것으로 학계에서 믿고 있다. 예 4.3의 최대짝짓기문제 또한, 설명되어 있는 것처럼, 크기가 홀수인 노드 부분집합 S에 대응하는 부등식들을 모두 포함하기 때문에, 필요한 반공간의 개수가 지수함수로 주어진다. 그러나, 예 4.3의 선형계획문제는 다항시간 안에 풀 수 있음을 증명할 수 있다. 2절에서 본 것처럼 예 4.4의 SDP 경우 또한, 가능해집합이, 셀 수 없이(uncountably) 많은 선형부등식들의 교집합으로 주어지기는 하나, 역시 선형계획문제로 볼 수 있으며, 실제로 몇 가지 조건이 만족되면, 본질적으로 선형계획을 푸는 것과 같은 원리로 다항시간에 최적해를 구할 수 있다. 이렇게, 선형최적화문제 (P)는 K의 특성에 따라, 문제의 계산복잡성이 다양하게 주어짐을 알 수 있다. 이러한 예들 중에서 다항시간 계산복잡성을 갖는 세 개의 예의 공통적인 특징은 K의 분리문제를 다항시간에 풀 수 있다는 것이며, 선형최적화문제 (P)의 경우, 최적화문제와 분리문제가 동치라는 놀라운 관계에서 그 다항성을 유추할 수 있게 된다.

분리문제란 닫힌 볼록집합과 한 점이 임의로 주어질 때, 점이 볼록집합에 속하는지를 규명하거나, 점과 볼록집합을 분리하는 분리초평면을 찾아내는 문제를 의미한다. 이러한 초평면이 존재하는 것은 3절에서 사용했던 분리초평면정리(부록 보조정리 A.2.1)에 따른다. 이러한 분리문제는 표면적으로 볼 때는 볼록집합 위에서 선형함수을 최적화하는 (P)보다는 쉬워 보인다. 그런데, 1981년, Grotschel, Lovasz, & Schrijver[21]는 K의 분리문제를 다항시간에 풀 수 있는 경우, 최적화문제 (P)를 다항시간에 풀 수 있음을 증명하였다. 이로 인하여, 그 이전에는 알려져 있지 않은 여러 가지 중요한 최적화문제들이 다항시간에 풀 수 있음이 증명되었다. 예를 들어 예 4.3의 최대짝짓기문제는 기존의 Edmonds의 알고리듬을 사용하여 다항시간에 풀 수 있을 뿐 아니라, 예 4.3에 설명된 선형계획법을 통해서 다항시간에 풀 수 있음이 밝혀졌다. SDP가 다항시간에 풀 수 있음이 최초로 증명된 것도, 바로 이러한 최적화문제와 분리문제의 연관성 때문이었다. 이러한 논의를 위하여 다음과 같은 정의를 도입하자.

정의 4.5 : (최적화문제) 임의의 유리수 벡터 $c \in Q^n$에 대하여, 다음과 같은 y를 구한다 : $c^T y \leq c^T x \ \forall x \in K$. □

정의 4.6 : (분리문제) 임의의 유리수 벡터 $y \in Q^n$가 주어졌을 때, y가 K에 속함을 보이거나, 그렇지 않은 경우, y를 K로부터 분리하는 분리초평면을 구한다. 즉, 다음을 만족하는 c를 구한다 : $\| c \| \geq 1$이며 $c^T y < c^T x \ \forall x \in K$. □

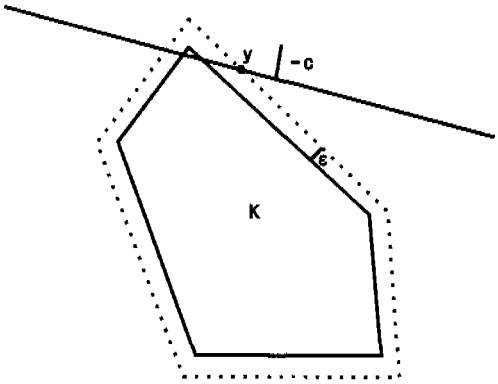
위에 주어진 최적화문제와 분리문제의 정의는 문제의 입력 계수 뿐 아니라 최적해의 유리수성(rationality)이 보장되는, 선형계획문제를 비롯한 조합최적화문제에서 보편적으로 사용되는 정의이다. 그러나, SDP를 비롯한 일반적인 선형최적화문제의 경우, 위의 정의는 부적합할 수 있다. 왜냐하면, K 위에서 선형함수를 최적화하는 점은 무리수를 좌표값으로 가질 수 있기 때문이다. 그러나, 보편적인 튜링기계나 RAM(random access machine)의 경우, 무리수를 어떻게 나타낼 수 있는지 알려진 바가 없다. 따라서 무리수 최적해를 정확히 구해야 한다는 무리한 정의는 다음과 같이 약화된 정의로 대체되는 것이 더 적절하다고 하겠다.

정의 4.7 : (ε-최적화문제) 임의의 유리수 벡터 $c \in Q^n$와 오차 $\varepsilon > 0$가 주어졌다고 하자. 다음과 같이 선형함수 $c^T x$를 "거의" 최적화하는 y를 구한다 : i) $d(y, K) \leq \varepsilon$; ii) $c^T y \leq c^T x + \varepsilon \ \forall x \in K$. 단, $d(y, K)$는 y와 K 사이의 거리를 의미한다. ([그림 3] 참조.) □

이에 맞추어 분리문제도 다음과 같이 약화된 정

의로 교체한다.

정의 4.8 : (ε-분리문제) 임의의 유리수 벡터 $y \in Q^n$와 오차 $\varepsilon > 0$가 주어졌다고 하자. 다음의 두 가지 중 하나를 수행한다. 첫째로 y가 K에 "거의" 속함을 보이거나, 즉 $d(y, K) \le \varepsilon$임을 밝히거나, 아니면 둘째로 y를 K로부터 "거의" 분리하는 분리초평면을 구한다. 즉, 다음을 만족하는 c를 구한다 : $\| c \| \ge 1$이며 $c^T y \le c^T x + \varepsilon \ \forall x \in K$. □



[그림 3] ε-최적화문제

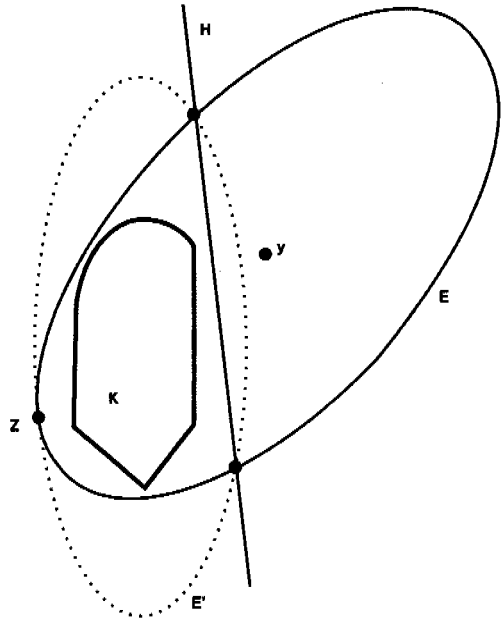
참고 4.9 : 이러한 ε-정의는 선형계획문제나 조합최적화문제의 경우, 원래 정의를 포함함을 보일 수 있다. 선형계획의 ε-최적화문제를 다항시간에 풀 수 있다고 하자. 이 때 다항시간이란, 문제입력크기 L과 오차의 입력크기 $|\log \varepsilon|$의 다항식으로 주어지는 시간을 의미한다. 그러면, 선형최적화문제의 정확한 해를 L의 다항시간에 풀 수 있음을 보일 수 있다. 이러한 이유는 첫째로 선형계획의 기저해는 각 좌표값의 절대값이나 분모의 값이 모두 2^L보다 크지 않기 때문이다. 이러한 사실로부터, 최적값과의 오차가 $O(2^{-2L})$인 가능해를 구하면 이로부터 최적기저가능해를 구할 수 있음을 보일 수 있다. 또한 같은 원리로 원래 제약식, $a_i^T x \le b_i$와 근사제약식, $a_i^T x \le b_i + 2^{-2L}$은 서로 동치임을 보일 수 있다. (이러한 사실들은 선형계획문제를 위한 타원해법의 기본성질이 된다. 예를 들어 [37]을 참조.) 따라서, $\varepsilon = O(2^{-2L})$이면 ε-최적해는 정확한 최적해임을 알 수 있다. 그러나, $|\log \varepsilon| = O(L)$임으로 L과 $|\log \varepsilon|$의 다항시간은 L의 다항시간이 된다. □

본 소절의 본론을 논의하기로 하자. (P)의 ε-분리문제를 다항시간에 풀 수 있으면 ε-최적화문제 역시 다항시간에 풀 수 있음을 보이고, SDP의 ε-분리문제를 다항시간에 풀 수 있음을 보이자. 그러면 SDP의 ε-최적화문제의 다항성을 보이는 것이다.

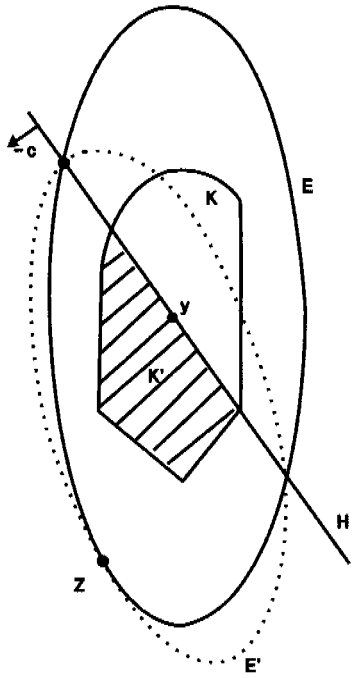
(P)의 ε-분리문제와 ε-최적화문제의 다항식성의 관점에서의 동등함은 바로 타원해법의 특성에서 비롯된다. 간략히 말해, 타원해법의 적용을 위해서는 매 반복 단계마다, 현재 해 x와 볼록집합 K 사이의 분리문제만 해결하면 된다. 예를 들어, 선형계획법에 타원해법을 적용하는 경우, 현재해가 가능하지 않을 때, 모든 제약식을 다 고려할 필요 없이, 현재 해가 만족하지 않는 제약식을 하나만 찾으면 일정한 비율 이하로 부피가 감소한 새로운 타원을 생성하여 다음 반복단계로 넘어 갈 수 있기 때문이다 ([그림 4] 참고). 이러한 관찰을 더 일반적인 문제 (P)에 적용하여, K의 분리문제를 다항시간에 풀 수 있다면, K를 정의하는 제약식들을 모두 명시적으로 알지 못하여도, (P)를 다항시간에 최적화할 수 있을 것이라는 직관적으로 추측할 수 있다.

실제로 (P)의 분리문제의 다항해법이 주어졌을 때, 이를 서브루틴으로 사용하는 타원해법을 어떻게 고안할 수 있는지를 논의하자. 우선, 선형계획을 위한 타원해법은 두 가지 방식으로 기술할 수 있다. 첫째는 선형계획문제를 가능성문제로 전환한 다음, 가능해를 찾도록 타원해법을 사용하는 것이다.

둘째는 목적함수를 함께 고려하여, 매 반복단계에서 현재해가 비가능일때는 가능성문제를 풀 때와 같이 현재해와 가능해집합을 분리하는 분리초평면을 사용하고 ([그림 4] 참고), 현재해가 가능일



[그림 4] 타원해법의 반복단계 : 현재해가 비가능인 경우



[그림 5] 타원해법의 반복단계 : 현재해가 가능인 경우

때는 목적함수가 현재해의 목적함수값과 같은 점들로 정의된 초평면을 사용하여, 목적함수가 감소하는 쪽으로 탐색 가능해 집합을 줄여 최적화문제를 직접 푸는 방식이다. 물론, 최적화문제를 가능성문제로 바꿀 수 없는 일반적인 선형최적화문제, (P)의 경우는 후자를 사용할 수밖에 없다.

이제 K의 분리문제를 풀 수 있는 알고리듬이 주어졌을 때, 이를 서브루틴으로 사용하여 후자의 타원해법이 최적화문제를 어떻게 푸는지 살펴보자.

정리 4.10 : [21] (P)에서 $K \subseteq R^n$이며, 다음의 조건을 만족하는 내부가능해 a_0, 실수 $0 < r \leq R$ 이 주어져있다고 가정하자:

$$B(a_0, r) \subseteq K \subseteq B(a_0, R).$$

여기서 $B(a_0, r)$는 a_0를 중심으로 하고 반지름이 r인 유클리드구(Euclidean ball)를 의미한다. 또한, 임의의 오차 $\varepsilon > 0$에 대하여 ε-분리문제를 푸는 알고리듬이 주어졌다고 하자. 그러면 n, $|\log r|$, $|\log R|$, 그리고 $|\log \varepsilon|$의 다항식으로 표시되는 횟수의, 기본연산 또는 ε-분리문제 알고리듬을 사용하여 ε-최적화문제를 풀 수 있다.

증명(스케치) : 먼저 논의의 간편성을 위하여 무리수 때문에 발생하는 오차가 없으며, 분리문제도 정의 4.6과 같이 정확히 풀 수 있다고 가정하고 타원해법의 한 반복단계를 설명하자.

초기해를 $x_0 = a_0$, 초기타원을 $E_0 = B(x_0, R)$로 놓는다. 일반적으로 k번째 반복단계를 시작할 때, 다음과 같은 타원 E_k가 주어진다 : $k-1$번째 반복단계까지 구해진 어떤 가능해보다 목적함수값이 같거나 작은 해들의 집합을 K_k라고 할 때, E_k는 K_k를 포함한다. E_k의 중심을 x_k라고 하자. k번째 반복단계는 다음과 같이 진행된다.

$E = E_k$, 그리고 $y = x_k$ 라고 하자. 주어진 분리문제 알고리듬을 사용하여, y와 K의 분리문제를 푼다. 다음의 두 가지 경우가 발생할 수 있을 것이

다. 첫째, y가 K에 속하지 않는 경우이다. 이 경우, 분리문제 알고리듬으로부터 y를 K로부터 분리하는 초평면 H도 함께 주어질 것이다. ([그림 4] 참조.) 이 경우, H와 평행한 분리초평면이 E와 접하는 점 z에서 E와 접하며, $E \cap H$에서 E와 교차하는 최소타원 E'을 구할 수 있다. (이 때, K에 대한 직접적인 정보가 필요 없음에 주목하자.) $E_{k+1} \leftarrow E'$로 하고 k번째 반복단계가 끝난다. 둘째, y가 K에 속하는 경우이다. 이 경우에는 $c^T x = c^T y$로 정의되는 초평면을 H로 정의하고 첫째 경우와 동일한 방법으로 새로운 타원 E'을 구하여 E_{k+1}으로 놓는다.

오차를 고려하는 경우, 위의 반복단계는 다음과 같이 수정된다. 우선 다음과 같은 파라미터들을 정의하자:

$$N = 4n^2 \lceil \log \frac{2R^2 \| c \|}{r\varepsilon} \rceil, \quad \delta = \frac{R^2 4^{-N}}{300n},$$

$$p = 5N.$$

분리문제는 δ-분리문제로 수정하며, 무리수의 경우, 소수점 이하 p자리까지만 저장한다. 이 정도의 오차만을 허용하며 반복단계를 진행하면 다음과 같은 성질들을 증명할 수 있다.

첫째, k번째 반복단계에서 생성되는 타원을 $E_k = \{x \in R^n : (x - x_k)^T A_k^{-1} (x - x_k) \le 1\}$라고 할 때, $\| x_k \| \le \| a_0 \| + R2^k$, $\| A_k \| \le R^2 2^k$, 그리고 $\| A_k^{-1} \| \le R^{-2} 2^{2k}$라는 상한들을 갖는다.

둘째, $\dfrac{E_{k+1}\text{의 부피}}{E_k\text{의 부피}} < e^{-1/5n}$을 만족한다.

셋째, $k-1$번째 반복단계까지 구해진 어떤 가능해보다 목적함수값이 같거나 작은 해들의 집합을 K_k라고 할 때, E_k는 K_k를 포함한다.

넷째, 둘째와 셋째 사실로부터, N번의 반복단계를 수행하기 전에, 최적목적함수값과의 차이가 ε을 넘지 않는 가능해를 구할 수 있음을 증명할 수 있다. 또한 반복단계를 N번 이상 수행할 필요가 없기 때문에, 첫째 사실과 결합하면, 알고리듬이 생성하는 숫자는 (n, $|\log r|$, $|\log R|$, 그리고 $|\log \varepsilon|$의) 다항식 입력길이를 가지며, 더 나아가 타원해법이 전체가 수행하는 연산이나 분리 알고리듬 역시 다항식 횟수만큼 사용된다는 결론에 이르게 된다. 따라서, 분리알고리듬이 다항시간을 갖는다면, 전체 타원해법도 다항시간 해법이 된다. ■

정리 4.10에 의하여, SDP의 ε-최적화문제의 다항식성을 보이기 위해서는, ε-분리문제를 다항시간에 풀 수 있음을 보이면 된다. 물론, SDP의 경우, 내부가능해가 주어지는 경우 정리의 가정들이 어떻게 만족되는지를 논의해야 할 것이다. 이러한 자세한 사항은 [21, 23]을 참조할 수 있으며, 본 논문에서는 이러한 기술적인 사항에 대한 논의를 생략하며, 단지 SDP의 ε-분리문제를 어떻게 다항시간에 풀 수 있는지를 구체적인 예를 통하여 서술함으로써, 직관적인 이해를 도모하며 증명에 대신한다.

예 4.11 : 예 2.2에서 보았던 SDP 제약식, $x_1 \begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix} + x_2 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} -2 & 1 \\ 1 & 0 \end{bmatrix} \ge 0$, 또는 $A(x) = \begin{bmatrix} x_1 + x_2 - 2 & -2x_1 + x_2 + 1 \\ -2x_1 + x_2 + 1 & x_1 + x_2 \end{bmatrix} \ge 0$을 고려하자. 주어진 벡터, $\bar{x} = [3, 1]^T$과 $K = \{x : A(x) \ge 0\}$로 정의된 분리문제를 풀어보자. 즉, $\bar{x} \in K$인지 또는 $A(\bar{x}) \ge 0$인지를 밝히거나 아니면 $c^T \bar{x} < c^T x \ \forall \ x \in K$인 c를 구해보자. (SDP의 경우, ε-분리문제가 아닌 정확한 분리문제를 풀 수 있다.)

"대칭적" 가우스소거법을 사용하면, 이 문제를 동시에 해결할 수 있다. $A(\bar{x}) \ge 0$일 필요충분조건은 행렬 $A(\bar{x})$에 가우스소거법을 적용할 때, 피봇원소들이 모두 비음인 것이다. (부록 정리 A.1.3 참조.) 이 때, 행들만을 소거하지 않고 대칭적으로 열들도 함께 소거한다. 이때, 행렬 $A(\bar{x}) = \begin{bmatrix} 2 & -4 \\ -4 & 4 \end{bmatrix}$는 $\begin{bmatrix} 2 & -4 \\ -4 & 4 \end{bmatrix} \to \begin{bmatrix} 2 & -4 \\ 0 & -4 \end{bmatrix} \to \begin{bmatrix} 2 & 0 \\ 0 & -4 \end{bmatrix}$와 같이 소거된다 :

따라서 $\overline{x}$는 K에 속하지 않음을 알 수 있다. 그리고 위에 적용한 소거는 다음과 같이 기본행연산(elementary row operation)과 기본열연산에 대응되는 행렬을 $A(\overline{x})$의 좌우에 곱한 것과 같은 의미이다 :

$$\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 2 & -4 \\ -4 & 4 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & -4 \end{bmatrix}.$$

따라서 다음의 관계가 성립한다 :

$$[\,0 \;\; 1\,]\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 2 & -4 \\ -4 & 4 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = -4.$$

이것은 다음의 식이 K의 원소들에는 만족되지만, $\overline{x}$에는 만족되지 않는다는 의미이며, 따라서 분리초평면을 정의하게 되는 것이다 :

$$[\,0 \;\; 1\,]\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} x_1+x_2-2 & -2x_1+x_2+1 \\ -2x_1+x_2+1 & x_1+x_2 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = -3x_1+9x_2-4 \geq 0. \quad \square$$

5. SDP의 조합최적화 응용

5.1 개요

지금까지 SDP의 기본 이론들을 살펴보았다. 5절에서는 SDP의 중요한 응용연구 중에서 조합최적화에 SDP를 응용한 연구들에 대하여 개관하려고 한다. 1 절에서 언급한 바와 같이 엄청난 SDP 관련 연구가 이루어지게 된 중요한 이유 중 하나는 SDP가 조합최적화문제를 푸는데 유용하게 이용될 수 있다는 점이었다. SDP와 조합최적화의 관계의 기원은 고유값(eigenvalue)과 조합최적화의 관계에 대한 연구까지 거슬러 올라 갈 수 있다. 이미 60년대 말에 고유값을 이용하여 조합최적화문제의 하한(최소화문제 가정)을 구할 수 있다는 연구가 있었고, 후에 Alizadeh[1]에 의해서 고유값을 구하는 문제가 SDP로 표현될 수 있다는 것이 알려졌기 때문이다. 그러나 조합최적화문제에 명시적으로 SDP가 이용된 것은 Lovasz[30]의 θ-함수에 대한 연구였다. 그리고 조합최적화문제에 SDP를 응용하는 연구를 본격적으로 촉발시킨 것은 Goemans & Williamson[19]의 연구이다. Goemans & Williamson은 최대절단면문제를 SDP로 완화하여 풀고, 이 해를 이용하여 가능해를 만들면 이 때 구해진 상한과 하한은 매우 근접한 값을 갖게 된다는 것을 보였다. 이 연구를 통하여 조합최적화문제 중에는 SDP 완화문제가 다른 완화문제보다 문제의 가능해집합을 더 밀접하게 표현할 수 있고, 따라서 SDP 완화문제의 해를 이용하면 기존의 해법들에 비해 개선된 근사해를 만들 수도 있다는 기대감이 생겼으며, 이에 따라 많은 조합최적화문제에 SDP를 활용하는 연구결과를 가져왔다.

SDP가 조합최적화문제의 해결에 활용되는 방법은 다음의 두 가지로 요약될 수 있다. 첫째는 조합최적화문제를 SDP로 완화시키고, 이 완화문제(relaxaton)를 풀어서 문제의 하한(lower bound, 최소화문제인 경우)을 구하는 것이다. 둘째는 SDP 완화문제의 해를 이용하여 조합최적화문제의 가능해를 구하는 것이다. 이 때 구해진 해가 항상 최적해의 α배 이하(최소화문제인 경우) 또는 이상(최대화문제인 경우)이 된다면, 이 해법을 α-근사해법이라고 부른다. 이러한 근사해는 최적목적함수값의 상한(최소화문제인 경우) 또는 하한(최대화문제인 경우)을 제공하게 된다. 특수한 조합최적화문제의 경우는 SDP 완화문제가 최적해를 제공할 수도 있으나, 많은 경우는 구해진 상한과 하한을 이용하여 최적해를 가늠해 볼 수 있을 뿐이다. 물론 분지한계법(branch and bound method)을 이용한다면 이러한 상한과 하한이 최적해 탐색과정에 효율적으로 사용될 수 있을 것이다.

이러한 관찰에 근거하여 본 절에서도 SDP의 응용연구들을 완화문제 연구와 근사해법 연구로 구분하였다. 이러한 구분 기준에 따라 5.2절에서는 조합최적화문제에 SDP 완화문제가 이용된 대표적인 예를 살펴보고, 5.3절에서는 SDP 완화문제의

해를 이용하여 조합최적화문제의 α-근사해를 구한 방법을 소개하며, 5.4절에서는 5.2절과 5.3절에서 소개된 것 이외에 SDP가 조합최적화에 응용된 예를 간략하게 나열한다. SDP의 조합최적화문제 응용 연구에 관한 대표적인 개관 논문으로는 Alizadeh[1], Goemans[16, 17], Goemans & Rendl[18] 등이 있다. 본 절에서는 Goemans & Williamson은 최대절단면문제 연구를 비롯하여, 앞에 언급한 개관논문들의 내용을 중심으로 조합최적화의 대표적 SDP응용사례를 다루고, 아울러 최근에 새로이 발표된 논문들을 추가하여 소개하기로 한다. 마지막으로 5.5절에서는 조합최적화에 대한 SDP완화의 장단점에 대해서 논의하기로 한다.

5.2 SDP를 이용한 조합최적화문제의 완화

이 절에서는 조합최적화문제에 SDP완화문제가 이용된 대표적인 예를 살펴보기로 한다. 이를 위해서 0-1 이차함수계획법과 최대절단면문제, 0-1 정수계획법, 최대안정집합문제(maximum stable set problem) 등에 SDP가 어떻게 활용되었는지 살펴본다. 5.4절에는 이 이외의 응용 예가 추가로 소개되어 있다.

5.2.1 0-1 이차함수계획법/최대절단면문제

우선 0-1 변수에 2차함수를 목적함수로 갖는 일반적인 형태인 0-1 이차함수계획법에 사용된 SDP 완화문제를 소개하기로 한다. 0-1 이차함수계획법은 최대절단면문제와 동등한(equivalent) 문제이다. 다시 말해서 임의의 0-1 이차함수계획법은 특정 최대절단면문제로 변환할 수 있고 반대로의 변환도 가능하다. 이러한 관계를 좀더 상세히 설명하여 보자. Q를 $(n-1)\times(n-1)$ 대칭행렬, c를 $n-1$ 차원 벡터라고 하면 0-1 정수계획법문제는 다음과 같이 표현할 수 있다 :

$$\text{(QP)} \quad \begin{array}{ll} \min & z^T Q z + c^T z \\ \text{s.t.} & z \in \{0,1\}^{(n-1)}. \end{array}$$

여기서 z를 모든 i에 대해서 $x_i = 2z_i - 1$ 인 x 벡터로 치환하면, 적절한 행렬 $\widehat{L}$과 벡터 q에 대해서 (QP)는 다음과 같이 표시된다 :

$$\begin{array}{ll} \min & x^T \widehat{L} x + 2q^T x \\ \text{s.t.} & x \in \{-1,1\}^{(n-1)}. \end{array}$$

선형식으로 표시된 항, $2q^T x$를 처리하기 위하여 추가변수 x_n을 도입하고, $L = \begin{bmatrix} \widehat{L} & q \\ q^T & 0 \end{bmatrix}$을 이용하면 최종적으로 다음과 같이 표현할 수 있다:

$$\begin{array}{ll} \min & x^T L x \\ \text{s.t.} & x \in \{-1,1\}^n, x_n = 1. \end{array}$$

이 문제에서는 제약식을 만족하는 해의 부호를 바꾸어도 목적함수의 값이 변하지 않기 때문에 $x_n = 1$라는 조건은 명시적으로 표시하지 않아도 최적해에 영향을 미치지 않으므로 생략할 수 있다. 따라서 우리가 고려하는 최종적인 모형은 다음과 같다 :

$$\text{(MC)} \quad \begin{array}{ll} \min & x^T L x \\ \text{s.t.} & x \in \{-1,1\}^n. \end{array}$$

이제 (MC)와 최대절단면문제가 동등한 문제임을 보이기로 하자. 두 문제의 관계에 대해서는 여러 학자의 연구내용이 있으나, 여기서는 Mohar & Poljak[34]의 표현을 근간으로 소개하기로 한다. 1절에서 소개한 바와 같이, 무방향 그래프에서 절단면(cut)이란, 노드집합(node set)을 양분했을 때, 두 노드집합에 걸치는 호(edge)들의 집합을 말한다. 최대절단면문제는 각 호에 가중치가 주어졌을 때, 호의 가중치의 합이 최대가 되는 절단면을 구하는 문제이다([그림 6] 참조). 최대절단면문제는 생산 및 제조분야와 회로설계문제(circuit layout design) 등에 응용성을 가진 조합최적화문제이다.

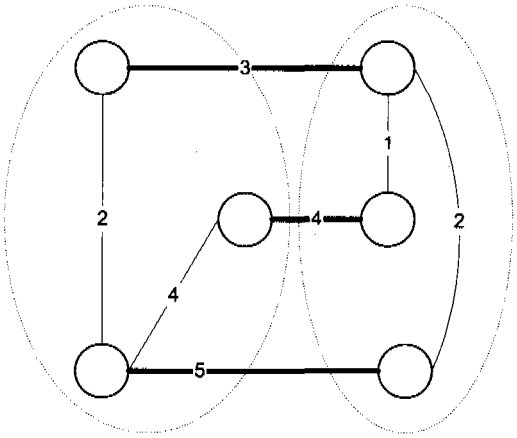
주어진 무방향의 그래프를 $G=(V,E)$라고 하고, 호의 가중치를 w_e, $e \in E$ 로 표시하기로 하자. 또한 노드의 부분집합 $S \subset V$ 에 대해서 $\delta(S)$

를 한쪽 노드는 S 에 다른 한쪽 노드는 $V \setminus S$ 에 속한 호들의 집합이라고 정의하자. 즉 $\delta(S)$ 는 절단면이 된다. 따라서 최대절단면문제는 $\sum_{e \in \delta(S)} w_e$ 를 최대로 하는 절단면 $\delta(S)$ 를 구하는 것이다. 변수 x_i 를 노드 i 가 S 에 속하면 1의 값을 갖고 아니면 -1의 값을 갖도록 설정하면 절단면 $\delta(S)$ 의 가중치 합 $w(S)$ 는

$$w(S) = \sum_{(i,j) \in E} w_{ij} \frac{1}{2}(1 - x_i x_j)$$

로 표시할 수 있다. 그래프 G 의 라플라시안(Laplacian), $L(G) = [l_{ij}]$ 를 고려하자. 라플라시안이란 $|V| \times |V|$ 행렬로, 주대각선 상의 원소에 대해서는 $l_{ii} = \sum_j w_{ij}$ 이고 기타 $i \neq j$ 인 원소에 대해서는 $l_{ij} = -w_{ij}$ 로 정의되는 행렬이다. 따라서 $w(S) = (1/4)x^T L(G) x$ 가 되어서 최대절단면문제를 (MC)의 형태로 나타낼 수 있음을 알 수 있다. 역으로 모든 0-1 2차함수계획법이 최대절단면문제로 표시될 수 있다는 것은, 모든 정방행렬은 라플라시안 행렬과 주대각행렬의 합으로 표시될 수 있다는 사실로부터 쉽게 유도된다. 즉 임의의 정방행렬 M은 적당한 그래프 G의 라플라시안 $L(G)$와 대각행렬(diagonal matrix) D의 합으로 표시되고, 임의의 벡터 $x \in \{-1, 1\}^n$에 대해서 $x^T D x = tr\, D$($tr\, D$는 행렬 D의 대각원소(diagonal elements)의 합을 표시한다)이므로 G의 최대절단면을 구함으로써 (MC)의 최적해를 구할 수 있다.

최대절단면문제는 (따라서 0-1 2차함수계획법도) NP-hard 문제이어서 빠른 시간(다항시간) 안에 정확한 해를 구하는 알고리듬은 존재하지 않는 것으로 믿어지는 문제이다. 따라서 최적해를 구하기 위해서 분지해법과 같은 탐색해법을 이용하거나 아니면 근사해를 구하는데 만족해야 할 것이다. 어느 경우이건 원 문제의 효율적인 상한은 귀중한 요소이다. 여기서는 SDP 완화문제를 이용해서 최대절단면문제, 즉 (MC)의 상한을 구하는 방법에 대해서 설명하기로 하자. $x^T L x = L \cdot (xx^T)$라는 사실에 주목하여 새로운 $n \times n$ 대칭행렬 $Y = [y_{ij}]$를 변수로 고려한다. 여기서 의도하는 것은 $y_{ij} = x_i x_j$의 관계가 성립되도록 하는 것이므로 xx^T 가 PSD이고 $x_i^2 = 1$ 이라는 사실을 행렬변수의 제약조건으로 부여한다. 우리가 xx^T 의 계수(rank)가 1이라는 조건까지 Y에 부가하면 $Y = xx^T$ 를 정확하게 만족시킬 수 있으나 행렬의 계수에 대한 조건이 가해지면 Y의 가능해집합이 볼록집합이 된다는 성질을 잃게 되므로 계수조건은 생략한다. 이러한 관찰을 종합하면 다음의 SDP문제가 원 문제의 완화문제가 됨을 알 수 있다:



[그림 6] 최대절단면의 예

$$\text{(SDP-MC)} \quad \begin{array}{lll} \max & \frac{1}{2} \sum_{(i,j) \in E} w_{ij}(1 - y_{ij}) & \\ \text{s.t.} & y_{ii} = 1, & i \in V \\ & Y = [y_{ij}] \geq 0. & \end{array}$$

최대절단면문제의 상한을 라플라시안 행렬의 고유값을 이용하여 구할 수 있음은 Mohar & Poljak[34]에 의해서 처음 제시되었고 위에서 기술한 SDP완화문제 형태의 표현은 Poljak & Rendl[40]에 의한 것이다.

5.2.2 0-1 정수계획법의 비선형완화

다음과 같은 유한다면체 P를 고려해 보자 :

$$P=\{\ x\in R^n \mid \sum_j a_{kj}x_j \le b_k\ \forall k \text{ and } 0\le x_j\le 1\ \forall j\}.$$

$P_I=P\cap\{0,1\}^n$라 하면 P_I는 일반적인 0-1 정수계획법의 가능해집합을 표현하게 되고 P는 P_I의 선형완화(linear relaxation)가 된다. 한편으로 Lovasz & Schrijver[31]는 각 제약식에 x_j와 $(1-x_j)$를 곱해서 새로운 부등식을 만드는 비선형완화방법을 고려하였다. 이러한 아이디어는 Sherali & Adams[43]에 의해서도 제시되었었다. P_I를 구성하는 임의의 부등식 $\sum_{j=1}^{n} a_{kj}x_j \le b_k$의 양변에 x_j 또는 $(1-x_j)$를 곱한 뒤에 0-1변수의 성질을 이용하여 ($x_j^2=x_j$), $y_{ij}=x_ix_j$, $y_{ii}=x_i^2=x_i$로 치환하면 P는 다음과 같은 행렬변수 $Y=[y_{ij}]$에 대한 가능해집합으로 표현할 수 있다 :

$$M(P)=\{\ Y\in R^{n\times n} \mid \sum_{i,j} c_{ijk}y_{ij}\le d_k\ \forall k,\ 0\le y_{ij}\le 1\ \forall i,j\ \}.$$

앞의 최대절단면문제에 대한 SDP완화문제처럼 $Y=xx^T$임에 근거하여 Y가 PSD라는 조건을 추가하고 계수 조건은 생략함으로써 P_I는 다음과 같은 SDP가능해집합으로 완화시킬 수 있다:

$$M_+(P)=\{\ Y\in R^{n\times n} \mid \sum_{i,j} c_{ijk}y_{ij}\le d_k\ \forall k,\ 0\le y_{ij}\le 1\ \forall i,j,\ Y=[y_{ij}]\ge 0\}.$$

뒤에 살펴보겠지만 $M(P)$와 $M_+(P)$는 P보다 P_I를 더 밀접하게 표현한다. 하지만 P의 제약식의 $2n$배 많은 제약식을 갖고 있음을 아울러 인식할 필요가 있다. $M(P)$, $M_+(P)$, P 및 P_I의 관계를 나타내기 위하여 $M(P)$와 $M_+(P)$를 P_I나 P처럼 x-공간에 표현하기로 하자. 이를 위하여 $M(P)$와 $M_+(P)$를 x-space에 투영(projection)한 결과인 $N(P)=\{x\in R^n \mid Y\in M(P), x_i=y_{ii}\ \forall i\}$와 $N_+(P)=\{x\in R^n \mid Y\in M^+(P), x_i=y_{ii}\ \forall i\}$를 정의하면 다음과 같은 관계가 성립함을 $N_+(P)$와 $N(P)$의 정의로부터 쉽게 알 수 있다.

성질 5.1 : $P_I\subseteq N_+(P)\subseteq N(P)\subseteq P$. □

한편으로 P에서 $N(P)$와 $N_+(P)$를 만드는 과정을 반복적으로 수행하는 경우를 고려해 보자. $N^r(P)=N(N^{r-1}(P))$, $N_+^r(P)=N_+(N_+^{r-1}(P))$라고 정의하면 다음과 같은 관계가 성립한다.

정리 5.2 : (Lovasz & Schrijver[31])

(i) $P_I=N^n(P)\subseteq\cdots\subseteq N(P)\subseteq P$,

(ii) $P_I=N_+^n(P)\subseteq\cdots\subseteq N_+(P)\subseteq P$. □

성질 5.1에서 알 수 있듯이 $N_+(P)$가 $N(P)$보다는 P_I의 강력한 완화이지만 얼마나 더 강력한지는 P_I의 구조에 따라 차이가 난다. 또한 반복적으로 $N(P)$와 $N_+(P)$를 수행하는 경우에 적어도 변수의 개수 이내의 횟수 안에 P_I와 일치하는 영역을 표시하게 되지만 P_I의 구조에 따라서는 훨씬 더 적은 횟수의 반복과정만으로도 P_I를 표시할 수 있게 된다. 현재 특정 0-1 정수계획법에 대해서 정수가능해집합을 표현하기 위하여 필요한 반복과정의 횟수에 대한 분석이 흥미로운 연구 분야로 부상하고 있다. 결론적으로 이 절에서 우리가 전하고자 하는 내용은 $N_+(P)$ 즉 $M_+(P)$를 이용하여 0-1 정수계획법의 하한(최소화문제 가정)을 구할 때 SDP가 사용된다는 것이다.

5.2.3 최대안정집합문제와 Lovasz의 θ함수

4.2절의 예 4.2에 소개되었던 최대안정집합문제(maximum stable set problem)는 조합최적화분야에서는 과거부터 잘 알려진 문제이다. 안정 집합(stable set)이란 주어진 무방향 그래프 $G=$

(V,E)에서 서로 인접하지 않는 - 두 노드를 잇는 호가 존재하지 않는 - 노드들의 집합이며, 안정집합 중 원소의 수가 최대가 되는 안정집합을 최대안정집합(maximum stable set)이라고 한다. 그래프 G의 최대안정집합의 원소의 수를 $\alpha(G)$로 표시하기로 하면 최대안정집합문제는 주어진 그래프 G의 $\alpha(G)$를 구하는 문제이다. 노드에 가중치가 주어지는 경우에는 포함된 원소의 가중치의 합이 최대가 되는 안정집합을 구하는 문제도 생각할 수 있다. 여기서 노드의 수는 n으로 가정하자. 즉 $n=|V|$이다.

Lovasz[30]는 $\alpha(G)$의 상한값을 제공하는 θ함수를 정의하였다. θ함수는 다양한 형태로 표현할 수 있는데, 이 절에서는 그 중의 하나를 소개하고 θ함수의 값을 구하는데 SDP가 어떻게 이용되는지 보이기로 한다. 그러나 이에 앞서 θ함수와 관련된 흥미로운 사실을 몇 가지 더 살펴보기로 하자. 우선 $\alpha(G)$를 구하는 0-1 정수계획법모형을 고려해 보자. 변수 x_i를 노드 i가 최대안정집합에 속하면 1의 값을 갖고 아니면 0의 값을 갖도록 설정하면 두 노드 i와 j사이에 호가 존재하면 두 노드는 동시에 안정집합에 속할 수 없으므로 $x_i+x_j \le 1$ 라는 부등식은 최대안정집합문제의 가능해들이 만족시켜야 하는 조건이 되며, 문제의 선형완화에 추가할 수 있다.

이러한 원리를 확장하기 위하여 집합 내의 모든 노드 쌍이 이웃하는 성질을 갖는 노드의 집합을 생각해 보자. 임의의 무방향그래프에서 이러한 성질을 만족시키는 노드의 부분집합 중 그 집합 밖의 임의의 노드를 추가하는 경우에는 이러한 성질을 만족시킬 수 없게 되는 노드의 부분집합을 클릭(clique)이라고 정의한다. 따라서 클릭의 노드 중에, 안정집합에 포함될 수 있는 노드의 수는 하나를 넘지 못하게 된다. 즉, 모든 클릭 C에 대해서 $\sum_{i\in C} x_i \le 1$ 이 성립한다. 다음과 같은 다면체를 생각해 보자 :

$$QSTAB(G) = \{x \ge 0 \mid \sum_{i\in C} x_i \le 1 \ \forall \text{ 클릭 } C \text{ of } G\}.$$

그러면 $\alpha(G)$를 구하는 0-1 정수계획법모형은 다음과 같이 나타낼 수 있다 :

$$\alpha(G) = \max\{\sum_{i\in V} x_i \mid x \in QSTAB(G) \cap \{0,1\}^n\}.$$

또한 $k(G)=\max\{\sum_{i\in V} x_i \mid x \in QSTAB(G)\}$ 라고 정의하면 $\alpha(G) \le k(G)$ 임을 쉽게 알 수 있다. 흥미로운 것은 $QSTAB(G)$를 구성하는 제약식 $\sum_{i\in C} x_i \le 1$ 는 무수히 많고 (문제의 입력길이에 대해서 지수식으로 표시된다.), 이 제약식에 대한 분리문제는 NP-hard이므로 $k(G)$를 다항시간에 풀 수 있는 해법은 알려져 있지 않았는데 반해서, Lovasz[30]에 의해서 제시된 θ 함수는 $\alpha(G) \le \theta(G) \le k(G)$를 만족하면서도 (즉 $\theta(G)$가 $k(G)$보다 $\alpha(G)$의 강력한 상한이다) $\theta(G)$는 타원해법이나 SDP의 해법에 의해서 다항시간 내에 풀 수 있다는 점이다.

이제 $\theta(G)$를 구하는 문제가 어떻게 SDP로 표현되는지 살펴보기로 하자. 주어진 그래프 G에 대해서 다음과 같은 정방행렬의 집합을 생각해 보자 :

$$P=\{A \in S_n \mid a_{ij}=1 \ \text{ if } \ (i,j) \notin E \ \text{ or } \ i=j\}.$$

그러면 원소의 수가 k개인 안정집합이 존재한다면 안정집합 내의 임의의 두 노드 i와 j에 대해서 호가 존재하지 않아서 $a_{ij}=1$ 이므로 이 안정집합에 해당하는 주부분행렬(principal sub-matrix)은 모든 원소가 1의 값을 갖는 행렬이 될 것이다. 이런 행렬을 J_k로 표시하자. 일반성을 손상하지 않으면서도 안정집합 내의 노드가 1에서 k까지의 번호를 갖고 있다고 가정할 수 있다. 그러면 주어진 그래프 G에 대해서 다음과 같은 정

방행렬 $\hat{A} \in P$ 가 존재할 것이다.

$$\hat{A} = \begin{pmatrix} J_k & \vdots & B_k \\ \cdots & \cdots & \cdots \\ B_k & \vdots & C_k \end{pmatrix}.$$

임의의 정방행렬 A 에 대하여 $\lambda_{\max}(A)$ 가 A 의 최대 고유값을 표시하는 것으로 정의하자. 고유값에 대해서는 다음과 같은 사실이 성립한다.

정리 5.4 : $n \times n$ 정방행렬 A 와 벡터 $y \in R^n$, 및 실수 a 에 대해서 $(n+1) \times (n+1)$ 정방행렬 $\hat{A}$ 가 $\hat{A} = \begin{pmatrix} A & \vdots & y \\ \cdots & \cdots & \cdots \\ y^T & \vdots & a \end{pmatrix}$ 와 같이 만들어졌다고 가정하자. 또한 A 와 $\hat{A}$ 의 고유치 $\{\lambda_i\}$ 와 $\{\hat{\lambda}_i\}$ 가 각각 크기 순으로 $\lambda_1 \le \cdots \le \lambda_n$, $\hat{\lambda}_1 \le \cdots \le \hat{\lambda}_{n+1}$ 과 같이 배열되었다고 가정하면 $\hat{\lambda}_1 \le \lambda_1 \le \hat{\lambda}_2 \le \lambda_2 \le \cdots \le \hat{\lambda}_n \le \lambda_n \le \hat{\lambda}_{n+1}$ 이 성립한다. □

위의 정리를 이용하면 $\lambda_{\max}(\hat{A}) \ge \lambda_{\max}(J_k) = k$ 임을 쉽게 알 수 있다. 따라서 모든 $A \in P$ 에 대하여 $\alpha(G) \le \lambda_{\max}(A)$가 성립된다. Lovasz에 의해서 제시된 θ함수의 여러 가지 표현 방법 중 $\theta(G) = \min_{A \in P} \lambda_{\max}(A)$ 의 정의는 위와 같은 관찰에 근거하고 있다. 그리고 $\lambda_{\max}(A) = \min\{t \mid tI - A \ge 0\}$ 가 된다. 왜냐하면, A 의 고유값이 $\{\lambda_i\}$ 일 때 $tI - A$ 의 고유값은 $\{t - \lambda_i\}$ 이므로 $tI - A$ 가 PSD라는 조건은 $tI - A$ 의 고유값 $\{t - \lambda_i\}$ 는 모두 0보다 같다는 조건과 동일하기 때문이다. 이러한 사실로부터 다음과 같이 최대고유값의 최소화문제를 SDP문제로 표현할 수 있다 :

$$\theta(G) = \min\{t \mid tI + \sum_{(i,j) \in E} x_{ij} E_{ij} \ge J\}.$$

여기서, J는 모든 원소가 1인 $n \times n$ 행렬이고 E_{ij} 는 (i,j) 원소와 (j,i) 원소를 제외한 모든 원소가 0인 $n \times n$ 행렬이다. 어떠한 $A \in P$ 에 대해서도 $A - J$ 는 E_{ij} 들의 결합으로 표시될 수 있다는 사실을 생각하면 위의 SDP모형에 대해서 쉽게 이해할 수 있다.

5.3 SDP완화를 이용한 근사해법의 개발

5.3 절에서는 SDP완화문제의 해를 이용하여 조합최적화문제의 α-근사해를 구하는 연구를 논의한다. 여기서는 2절에서 정의한 최대절단면문제의 근사해법 연구를 토의하려고 한다. 기타 유사한 응용 예들이 5.4절에서 추가로 소개되어 있다.

1절에서 언급한대로 Goemans & Williamson[19]의 최대절단면문제는 SDP의 가장 중요한 응용 연구로, 2000년 수리계획학회(mathematical programming society)에서 풀커슨상을 수상하였고 Pulleybank에 의해서 조합최적화분야의 대표적 연구업적 10건 중의 하나로 선정되기도 하였다. Goemans & Williamson은, 5.2.1절에서 소개한 최대절단면문제의 SDP완화문제의 해를 이용하여 가능해를 만들면 구해진 상한과 하한은 매우 근접한 값을 갖게 된다는 것을 보였다. 5.2.1절의 최대절단면문제의 SDP 완화문제를 다시 고려하자 :

$$\text{(SDP-MC)} \quad \begin{aligned} \max \quad & \frac{1}{2} \sum_{(i,j) \in E} w_{ij}(1 - y_{ij}) \\ \text{s.t.} \quad & y_{ii} = 1 \quad i \in V \\ & Y = [y_{ij}] \ge 0. \end{aligned}$$

PSD 행렬은 정방행렬의 곱으로 표시될 수 있으므로(부록 정리 A.1.3) $Y = V^T V$ 에서 v_i를 V 의 i번째 열로 이루어진 벡터라고 하면 (SDP-MC)는 다음과 같이 표현할 수 있다 :

$$\begin{aligned} \max \quad & \frac{1}{2} \sum_{(i,j) \in E} w_{ij}(1 - \mathrm{v}_i \cdot \mathrm{v}_j) \\ \text{s.t.} \quad & \| \mathrm{v}_i \|^2 = 1 \quad i \in V \\ & \mathrm{v}_i \in R^n \quad i \in V. \end{aligned}$$

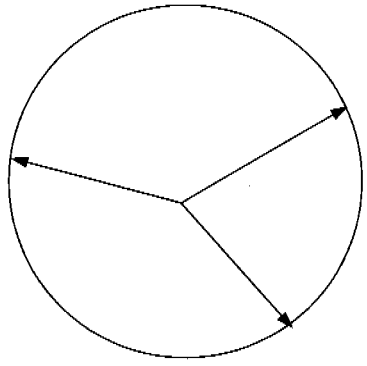
위의 모형에서 알 수 있듯이 노드 하나에 벡터가 하나씩 대응되게 되는데 Goemans & William

son[19]은 절단면, 즉, $\delta(S)$ 를 선택하기 위하여 이 벡터 값에 따라 노드를 S 에 포함시킬지 아닐지를 결정하였다. 이것은 마치 선형완화식의 해를 이용하여 정수해를 구성하는 것에 비교될 수 있다. $\| v_i \|^2 = 1$ 라는 조건에서 우리는 구해지는 벡터가 n 차원의 ($|V| = n$ 가정) 단위구 안에 [그림 7]에서와 같은 모습을 하고 있음을 쉽게 알 수 있다. 우리가 무작위로 [그림 8]에서처럼 단위구의 중심을 지나는 초평면(hyperplane)을 생성한 다음 이를 이용하여 벡터들을 두 부분 즉 S 와 $V \setminus S$ 에 속할 집단으로 나눌 수 있다. 이를 무작위 초평면해법(randomized hyperplane technique)이라고 부른다. 우리의 관심은 "이렇게 만들어진 절단면의 가중치의 합은 최대절단면문제의 최적해에 대해서 얼마나 근사하게 접근할 수 있을까?"이다. 초평면은 무작위로 선택되므로 우리는 구해진 해의 가중치의 합에 대한 기대값만을 계산할 수 있다. 그러나 복잡한 과정을 거치면 다항시간 내에 기대값 만큼의 목적함수를 보장하는 해를 구할 수 있기 때문에 이 해법은 확정적해법(deterministic algorithm)으로 변형시킬 수 있다 (Mahajan & Ramesh[33]).

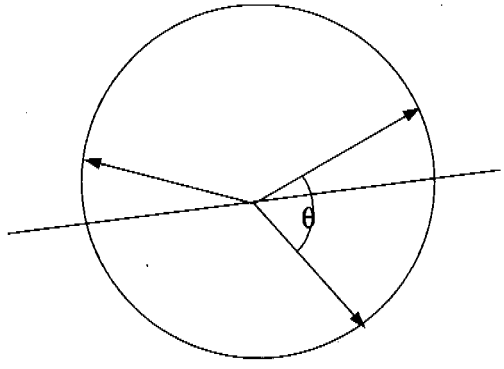
무작위 초평면해법에 의해 구해진 해의 기대치는 다음과 같이 분석할 수 있다:

$$
\begin{aligned}
& E\Big(\sum_{e \in \delta(S)} w_e\Big) \\
&= \sum_{(i,j) \in E} w_{ij} \times [\text{초평면이 } v_i \text{와 } v_j \text{를 분리할 확률}] \\
&= \sum_{(i,j) \in E} w_{ij} \frac{\theta}{\pi} \\
&= \sum_{(i,j) \in E} w_{ij} \frac{\arccos(v_i \cdot v_j)}{\pi} \\
&\geq 0.878 \times \frac{1}{2} \sum_{(i,j) \in E} w_{ij}(1 - v_i \cdot v_j) \\
&\geq 0.878 \times \text{최적해}.
\end{aligned}
$$

마지막에서 두 번째 부등식은 아크코사인(arccos) 값에 대한 간단한 분석에서 도출할 수 있으며 마지막 부등식은 SDP 완화문제의 정의로부터 자명하다.



[그림 7] 무작위벡터의 생성



[그림 8] 무작위벡터의 분류

5.4. 기타 조합최적화문제에의 SDP의 활용

이 절에서는 앞에서 다루어진 내용 이외에 조합최적화에 SDP를 활용한 최근까지의 연구 결과를 소개한다. 대부분의 연구는 SDP완화와 함께 α-근사해법을 제시하였으나 SDP완화만을 제시한 연구도 있다. 가능성문제(feasibility problem)인 SAT을 제외하고는 모두 최적화문제(optimization problem)의 형태이며 가능성문제인 경우는 NP-complete, 최적화문제인 경우는 NP-hard 문제들이다.

• MAX 2 SAT/ MAX SAT 문제

정의 : 참(true) 또는 거짓(false)으로 판명할 수 있는 명제(literal)들과 적당한 개수의 명제들의 이접(disjunction)으로 이루어진 절(clause)의 집합이 주

어졌을 때, 주어진 절들 중 참이 되는 절의 수가 최대가 되도록 명제에 참 또는 거짓을 할당하는 방법을 찾는 문제를 MAX SAT문제라고 한다. 모든 절을 구성하는 명제의 수가 k 개 이하인 MAX SAT문제를 MAX k SAT문제라고 부른다.

연구결과 : Goemans & Williamson[19]은 앞서 언급한 MAX CUT의 해법을 이용하여 MAX 2 SAT에 대해서는 0.878-근사해법을 MAX SAT문제에 대해서는 0.758-근사해법을 제시하였다. Feige & Goemans[13]는 MAX 2 SAT에 대해서 개선된 0.931-근사해법을 개발하였다.

• MAX DICUT 문제

정의 : MAX DICUT문제는 방향성 있는 그래프에 정의된 MAX CUT문제이다. 방향성이 있는 그래프 $G=(V,A)$와 방향성이 있는 호의 가중치 w_{ij}, $(i,j)\in A$ 가 주어져 있다고 가정하자. 이때 i를 호 (i,j)의 꼬리, j를 머리로 부른다. MAX DICUT문제는 꼬리가 S 에 머리는 $V\setminus S$ 에 속한 호들의 가중치의 합이 최대가 되도록 노드의 부분집합 $S\subset V$ 를 선택하는 문제이다.

연구결과 : Goemans & Williamson[19]은 역시 앞서 언급한 MAX CUT의 해법을 이용하여 0.796-근사해법을 제시하였고, Feige & Goemans[13]가 개선된 0.859-근사해법을 발표하였다.

• MAX k CUT/ MAX BISECTION 문제

정의 : MAX k CUT문제는 최대절단면문제의 확장이다. 무방향의 그래프 $G=(V,E)$와 각 호에 가중치가 주어져 있는 경우, MAX k CUT문제는 노드집합(node set) V 를 $(V_1,\cdots,V_k)$ 의 k 개의 부분집합으로 분할했을 때, 서로 다른 부분집합의 노드간에 걸쳐있는 호(edge)들의 가중치의 합이 최대가 되도록 하는 $(V_1,\cdots,V_k)$ 를 결정하는 문제이다. $k=2$ 이고 $|V_1|=|V_2|=\frac{|V|}{2}$ 인 MAX k CUT문제를 MAX BISECTION문제라고 부른다.

연구결과 : Frieze & Jerrum[14]은 MAX k CUT 문제에 대해서 k값에 따라 서로 다른 근사해를 제공하는 근사해법을 제시하였고 MAX BISECTION 문제에 대해서는 0.651-근사해법을 제시하였는데, 후자의 문제에 대해서 Ye[49]는 개선된 0.699-근사해법을 제시하였다.

• GRAPH PARTITION문제

정의 : GRAPH PARTITION문제는 다양한 유형의 문제를 포함하는 포괄적인 의미로 사용되기도 하는데 여기서는 특정한 문제로 좁혀서 정의된 경우이다. GRAPH PARTITION문제는 MAX k CUT문제처럼 노드집합(node set) V 를 $(V_1,\cdots,V_k)$ 의 k 개의 부분집합으로 분할하는 방법을 결정하는 문제이다. 차이는 각 부분집합에 속한 노드 수가 주어진 갯수 이상이 되어야 하고 서로 다른 부분집합의 노드간에 걸쳐있는 호(edge)들의 가중치의 합이 최소가 되도록 하는 $(V_1,\cdots,V_k)$ 를 결정한다는 점이다.

연구결과 : Wolkowicz & Zhao[47]는 SDP완화를 이용하여 하한을 구하는 방법과 함께 계산결과를 제시하였다.

• DENSE k SUBGRAPH/ DENSE $\frac{n}{2}$ SUBGRAPH 문제

정의 : DENSE k SUBGRAPH문제는 무방향의 그래프 $G=(V,E)$, 양의 정수 k, 그리고 각 호에 가중치가 주어져 있을 때, S 에 속한 노드 사이의 호들의 가중치의 합이 최대가 되도록 k 개의 노드로 구성된 부분집합 $S\subset V$ 를 선택하는 문제이다.

연구결과 : 임의의 k 에 대해서는 Srivastav & Wolf [45]에 의해서 근사해법이 제시되었다. 이들의 해법은 $k=\frac{n}{2}$ 인 경우에는 0.48-근사해법이 되는데 이러한 특수한 경우에 대해서 Ye & Zhang[50]은 0.586-근사해법을 제시하였다.

• MAX VERTEX COVER 문제

정의 : MAX VERTEX COVER문제는 무방향의 그래프 $G=(V,E)$와 양의 정수 k가 주어져 있을 때, S에 속한 노드에 연결되어 있는 호들의 수가 최대가 되도록 k 개의 노드로 구성된 부분집합 $S \subset V$를 선택하는 문제이다.

연구결과 : Han, Ye, Zhang, & Zhang[27]은 과거의 선형계획완화를 이용한 최선의 0.75-근사해법에 대해서 $k \geq \frac{n}{2}$ 인 경우에 k값에 따라 달라지나 적어도 0.8-근사해를 보장하는 개선된 해법을 제시하였다.

• QUADRATIC ASSIGNMENT 문제

정의 : QUADRATIC ASSIGNMENT문제는 n개의 시설을 n개의 장소에 최소의 비용으로 배치하는 문제이다. 이 때 비용함수는 시설 사이에 물동량과 배치되는 장소의 거리의 곱에 비례한다.

연구결과 : Zhao, Karisch, Rendl, & Wolkowicz [51]는 SDP완화를 이용하여 하한을 구하는 방법과 함께 계산결과를 제시하였다.

• QUADRATIC KNAPSACK 문제

정의 : 유한집합 $N=\{1,\cdots,n\}$과 각 원소 $i \in N$에 양의 정수 a_i, 임의의 양의 정수 B와 $n \times n$비용 행렬 $C=[c_{ij}]$가 주어졌을 때, QUADRATIC KNAPSACK문제는 다음과 같이 정의된다:

$$\text{(QKP)} \quad \begin{array}{ll} \max & x^T C x \\ \text{s.t.} & a^T x \leq B \\ & x \in \{0,1\}^n. \end{array}$$

연구결과 : Helmberg, Rendl & Weismantel[26]은 SDP완화를 이용하여 하한을 구하는 방법과 함께 계산결과를 제시하였다.

• GRAPH COLORING 문제

정의 : GRAPH COLORING문제는 무방향의 그래프 $G=(V,E)$에서 호로 연결된 두 노드는 서로 다른 색을 할당해야 되는 제약을 만족시키면서 최소 종류의 색을 이용하여 노드를 채색하는 방법을 구하는 문제이다.

연구결과 : Karger, Motwani, & Sudan[28]은 SDP 완화와 무작위 초평면해법을 이용하여 근사해법을 제시하였다.

• SAT 문제

정의 : MAX SAT문제가 가능성(feasiblity)문제로 표시된 형태이다. 참(true) 또는 거짓(false)으로 판명할 수 있는 명제(literal)들과 적당한 갯수의 명제들의 이접(disjunction)으로 이루어진 절(clause)의 집합이 주어졌을 때, 주어진 절들 모두 참이 되도록 명제에 참 또는 거짓을 할당하는 방법이 존재하는지를 결정하는 문제이다.

연구결과 : Klerk, Maaren, & Warners[11]은 SDP 완화가 가능성(feasiblity) 문제에도 유용하게 응용될 수 있음을 보였다.

• BETWEENESS 문제

정의 : BETWEENESS문제는 유한집합 V와 V에 속한 서로 다른 3원소로 이루어진 순서쌍 (x_i, x_j, x_k)들이 주어져 있을 때 주어진 순서쌍 중에서 x_j가 x_i와 x_k사이에 오는 순서쌍의 수가 최대가 되도록 V의 원소의 순서를 정하는 문제이다.

연구결과 : Chor & Sudan[10]은 SDP완화와 무작위 초평면해법을 이용하여 순서를 정하는 근사해법을 제시하였다.

• 기 타

Anjos & Wolkowicz([4], [5])는 MAX CUT문제에 대해서 기존의 SDP완화보다 강화된 SDP완화 모형을 제시하였고, Alon & Kahale[3]은 최대안정집합의 원소의 수 $\alpha(G)$에 대한 개선된 근사해법을 제시하였다.

5.5 SDP완화의 장단점

이 절에서는 SDP완화가 전통적인 선형계획완화에 비해 조합최적화문제를 푸는데 과연 유리한 것인지, SDP완화를 이용한 해법으로 조합최적화문제를 효율적으로 해결하기 위해서는 어떤 점에 유의해야 되는지 등에 대해서 고려해보기로 한다.

조합최적화문제를 푸는데 완화문제의 용도는 완화문제를 풀어서 원 문제의 하한(최소화문제가정)을 구하는 것과 완화문제의 해를 이용하여 조합최적화문제의 실행가능해를 구하는 것이다. 어느 경우이든 원 문제의 최적해에 가까운 해를 제공하는 강력한 완화문제가 유용할 것이다. 하지만 분지해법 등을 이용하여 최적해를 구하는 경우까지를 고려한다면 완화문제가 제공하는 해의 질(quality)과 이러한 해를 구하는데 필요한 계산부담과의 상호관계에 따라 실제 조합최적화문제의 최적해를 구하는데 효율적인 완화문제가 어떤 것인지 결정될 것이다. 이러한 관점에서 SDP완화와 전통적인 선형계획완화를 비교해 보기로 하자.

완화문제가 제공하는 해의 질적인 면에서는 적어도 SDP완화가 더 우수하다고 말할 수 있다. 왜냐하면 SDP완화에는 선형계획완화에 포함되는 모든 제약식을 포함시킬 수 있기 때문이다. 특히 2절의 안정집합문제에서처럼 어떤 선형계획완화모형은 다항시간 내에 풀 수 없으나 그 선형계획완화모형보다 강력한 SDP완화는 다항시간 내에 풀 수 있는 경우도 있다. 앞에서 소개한 많은 SDP에 기반을 둔 효율적 근사해법이 선형계획완화를 이용한 근사해법에 보다 개선된 결과를 가져올 수 있었던 것도 SDP완화해가 선형계획완화해보다 질적으로 우수했기 때문이었다. 한편으로는 Goemans & Rendl[18]의 연구에 나타난 것처럼 SDP완화에 사용된 행렬이 일정 조건을 만족하는 경우는 SDP완화를 동일한 선형계획완화로 변형할 수 있는 경우도 있다.

SDP완화와 선형계획완화를 푸는데 드는 노력은 단순하게 비교하기는 불가능하게 생각된다. 왜냐하면 같은 조합최적화문제에 대해서 동일한 수준의 해를 제공하는 SDP모형과 선형계획모형을 대상으로 비교해야 하는데 이러한 모형을 찾는 것이 어렵기 때문이다. 다만 그동안 계산경험을 제시한 연구결과를 통해서 알 수 있는 것은 아직까지는 기존의 조합최적화문제의 최적해를 구하는데 SDP완화를 이용하여 계산시간을 획기적으로 단축했다는 결과는 없다는 것이다. 이는 아직은 SDP완화에서 얻는 해의 질적인 우월성이 내부점해법 형태의 SDP해법을 푸는데 걸리는 시간을 상쇄하기 힘들기 때문으로 추정된다. 특히 Helmberg & Rendl[24]의 MAX CUT문제의 계산결과에서 볼 수 있듯이 SDP완화를 사용하는 경우에도 더 나은 완화해를 얻기 위해서는 유효부등식(valid inequality)을 추가하여야 하고 이러한 유효부등식을 첨가해서 반복적으로 SDP를 풀 때 계산상의 부담이 큰 것으로 나타난다. 또한 근사해법의 경우도 제시된 해법들이 대부분 확률적인 해법인데 이를 확정적 해법으로 변형시킬 수 있다고 했으나 이는 이론적으로만 의미가 있고 현실적으로는 여러 번 시행하여 기대치에 가까운 해를 구하여야 한다. 이 경우에 SDP문제를 푸는데 드는 시간이 부담이 된다면 근사해를 이용해서 실행가능해를 구하는 것도 실용성이 떨어질 수 있다.

이처럼 현실에서 발생하는 실제 조합최적화문제를 풀 때 SDP가 효율적으로 활용되기 위해서는 SDP완화를 이용한 해법을 개발할 때 계산부담을 줄이기 위한 노력이 이루어져야 한다. 이를 위해서 고려되어야 할 항목들에 대해서는 6.3절에서 다루기로 한다.

6. 추후 연구 방향

6.1 SDP 가능성문제 복잡성 연구

본론에서 개관하였듯이 SDP의 가능성문제가 NP 범주에 속하는지 아직 규명되어 있지 않다. 이것은 단순히 기술적인 문제가 아니라, SDP문제가 가능할 때, 해를 직접 사용하지 않고, 다항시간 내에 문제의 가능성을 확인할 수 있는 간접적인 증거

를 어떻게 나타내느냐하는 보다 근본적인 문제이다.

6.2 SDP의 해집합(spectrahedron) 구조 연구

현재 SDP를 위한 해법은 내부점해법이 주종을 이루고 있다. Nesterov & Nemirovskii[35]는 이론적인 효율성과 실용적인 효율성을 모두 갖춘 내부점 해법이, 소위 self-concordant barrier function이 존재하는 SDP를 포함한 볼록계획법에 적용될 수 있음을 보임으로써 본격적인 해법연구를 촉발시켰다. 현재 가장 활발한 연구방향으로 보인다. 그러나, 이러한 연구량에 비해, 선형계획법을 위하여 내부점해법이 보여 주었던 효율성을 SDP 경우에도 성취하기 위해서는 본질적인 장애가 존재하는 것으로 보인다. 그것은 내부점해법이 SDP에 적용되는 경우, 규격이 제곱정도로 커진 행렬의 역행렬을 구하는 과정이 불가피하며, 이러한 이유 때문에 현재로서는 문제의 크기가 중형 이상인 경우, 내부점해법의 효율성은 부정적으로 생각되고 있다. 실제로, 문제의 특수한 구조를 이용한 개별적인 해법을 개발하여 사용하는 경우가 많다.

이러한 내부점해법의 대안으로 생각할 수 있는 것이 심플렉스 타입 해법이다. 심플렉스해법은 그 고유한 특성상, 각 반복단계에서 현재 기저를 "rank one" 변환만이 필요하기 때문에 내부점해법이 가지는 단점을 피할 수 있다. 그러나, SDP를 위한 심플렉스해법은 아직 어려움을 갖고 있다. 그것은 해집합이 다면체로 주어지는 선형계획과는 달리, SDP의 해집합은 꼭지점 사이를 이동하는 심플렉스 타입의 반복단계가 잘 정의되어 있지 않기 때문이다. 이를 위해서는 SDP의 해집합, "spectrahedron"의 기하학적인 성질을 해법 개발의 관점에서 더 연구가 필요하다고 본다.

6.3 SDP의 조합최적화 응용 연구

5.5절에서 언급한대로 현실에서 발생하는 실제 조합최적화문제를 풀 때 SDP가 효율적으로 활용되기 위해서는 SDP완화를 이용한 해법을 개발할 때 계산부담을 줄이기 위한 노력이 이루어져야 한다. 이를 위해서 다음의 사항들이 고려되어야 한다. 우선 조합최적화문제는 특수한 구조를 갖는 경우가 대부분이므로 적절한 SDP완화모형을 선택하는 것이 중요하다. 또한 SDP완화모형의 크기가 대규모인 것이 일반적이므로 계산 시 모든 제약조건을 다 포함할 것인지 일부만을 교대로 첨가하여 반복적으로 풀 것인지 결정하는 전략이 필요하다. 둘째로, 내부해법 형태의 SDP해법을 적용하는 경우에 Benson, Ye, & Zhang[8]의 연구에서처럼 대상 조합최적화문제의 구조에 특화된 SDP해법을 고려할 수 있다면 향상된 결과를 얻을 수 있을 것이다. 마지막으로 더 나은 완화해를 얻기 위해서 유효부등식 추가해서 반복적으로 SDP를 풀 때 고려해야 할 사항이 있다. SDP해법의 특성상 최적해가 아닌 근사해를 얻는 것이므로 첨가할 유효부등식을 찾기 위해서는 근사해를 구하는 과정을 어느 시점에 중단할지 결정하여야 하고, 유효부등식이 첨가된 후의 SDP문제의 내부점을 어떻게 구할 것인가를 고려하여야 한다.

부 록

A.1 행렬의 성질

PSD 행렬을 포함하는 여러 가지 행렬의 유용한 성질들을 도출하는 수단 중에 하나는 행렬을 대각화(diagonalization)하는 것이다. 행렬을 대각화의 기본이 되는 정리 중에 하나가 Schur의 정리이다. Schur의 정리는 서로 수직(orthogonal)인 단위(unit)벡터들로 열들을 구성하는 일원행렬(unitary matrix)를 사용한다. 즉, 일원행렬이란 $U^H U = I$인 복소행렬 U를 모두 일컫는 말이다. 여기서 U^H는, 예를 들어, $\begin{bmatrix} 2-i & 1+i \\ -i & 3 \end{bmatrix}^H = \begin{bmatrix} 2+i & i \\ 1-i & 3 \end{bmatrix}$가 되는 행렬연산을 의미하며, 실수행렬의 전치

(transpose) 연산의 복소행렬로의 확장임을 쉽게 알 수 있다. 따라서 특별히 U가 실수행렬인 경우는, $U^TU=I$가 되며, 이 때 U를 단위수직행렬(orthogonal matrix)라고 부른다.

보조정리 A.1.1 (Schur의 정리) 모든 복소정방행렬 A에 대해, $U^{-1}AU=T$가 역삼각행렬이 되도록 하는 일원(unitary) 행렬 U가 존재한다. 그리고 A와 T는 동일한 고유값(characteristic value)들을 가지며, 이러한 고유값들은 T의 대각선에 나타난다. □

증명은 생략하며, 여기서는 이 정리를 사용하여, 도출할 수 있는 유용한 성질들을 살펴보자. 정방행렬 A가 $A=A^H$를 만족할 때, 허미션(Hermition) 행렬이라고 부른다. (허미션 행렬의 대각원소들은 모두 실수가 될 수밖에 없음을 쉽게 알 수 있다.) 원소가 모두 실수인, 실수행렬의 경우, 허미션 행렬은 곧 대칭행렬임을 의미한다. 또한 허미션 행렬이 역삼각형이면 곧, 대각선을 제외한 원소는 모두 0인 대각행렬이 됨을 알 수 있다. 따라서, Schur의 정리는 다음과 같은 행렬의 중요한 성질을 내포한다.

정리 A.1.2 (스펙트럼 정리) 모든 (복소) 허미션 행렬 A는 일원행렬을 이용하여 대각화시킬 수 있다: $U^{-1}AU(=U^HAU)=\Lambda$. 이때, 대각행렬 Λ의 원소는 A의 고유값이며 모두 실수가 된다. 따라서, 모든 (실수) 대칭행렬은 직교 행렬을 이용하여 대각화시킬 수 있다: $Q^{-1}AQ=Q^TAQ=\Lambda$. □

같은 규격의 행렬로 이루어진 벡터공간을 생각하자. 만약 A, B가 같은 규격의 행렬이라면, 그 내적은 다음과 같이 정의된다 : $A\cdot B=\sum_{i,j}A_{ij}B_{ij}$ $\mathrm{tr}A^TB$. 일반 벡터 내적의 일반화라고 하겠다. SDP문제에서 사용하는 행렬은 $n\times n$ 행렬 벡터공간 중, 대칭행렬 부분공간에 국한된다. 실제로 $n\times n$ 대칭 행렬은 $n(n+1)/2$ 차원 벡터공간과 대응("homeomorphic")된다. 예를 들어, $n=3$인 경우, 두 공간사이에 다음과 같은 사상을 정의하면 덧셈, 스칼라곱, 그리고 내적을 보존하는 것을 알 수 있다.

$$(a,b,c,d,e,f)\leftrightarrow\begin{bmatrix} a & d/\sqrt{2} & e/\sqrt{2} \\ d/\sqrt{2} & b & f/\sqrt{2} \\ e/\sqrt{2} & f/\sqrt{2} & c \end{bmatrix}$$

이러한 대칭행렬 중에서 A가, 모든 벡터 x에 대해 $x^TAx\geq 0$이면, A를 PSD(positive semidefinite) 행렬이라고 한다. 그리고 만약 $A-B$가 PSD이면, $A>B$라고 표시하는데, 이 때, "$\geq$"를 "Löwner 순서"라고 부르며, PSD 행렬 위에 부분순서(partial order)를 정의하게 된다.

정리 A.1.3 다음 문장들은 모두 동치이다.

i) 모든 벡터 y에 대해, $y^TAy\geq 0$이다.
ii) 행렬 A의 고유값들은 모두 비음이다.
iii) A의 모든 주부분행렬(principal submatrix)의 행렬식(determinant)은 비음이다.
iv) A에 가우스 소거법을 적용할 때, 모든 피봇원소는 비음이다.
v) $A=W^TW$로 표현할 수 있다. □

A.2 다면체 집합(Polyhedral sets)

보조정리 A.2.1 (분리초평면정리) 만약 C가 닫힌 볼록집합이고, 벡터 x가 C에 속하지 않는다면, x와 C를 분리하는 초평면(hyperplane)이 존재한다. 즉, 모든 $c(\in C)$에 대해 $c\cdot y\leq b$이고 $x\cdot y>b$인, 벡터 y와 실수 b가 존재한다.

증명 : x와 C를 분리시키는 초평면을 찾아야한다. d를 $\|x-c\|^2$의 최소극한값이라고 하자.(단, $c\in C$) 그리고 $\|x-c_0\|^2=d$인 c_0를 잡자.(C는 닫힌 집합이므로 그러한 $c_0\in C$이 존재한다.) 그리고 $y=x-c_0$, $b=c_0\cdot y$ 라고 하자. 그러면 명백하게 $x\cdot y=y\cdot y+b>b$ 이다.

$$x\cdot y > b \qquad (*)$$

그리고 $c \in C$인 점을 하나 잡자. 그러면 C가 볼록하므로 $\varepsilon > 0$가 충분히 작을 때, $\varepsilon c + (1-\varepsilon)c_0 \in C$이다. 따라서 다음 관계들이 성립함을 알 수 있다.

$$\begin{aligned}
&\|x-(\varepsilon c+(1-\varepsilon)c_0)\|^2 \geq d \\
&\quad \Leftrightarrow \|x-c_0-\varepsilon(c-c_0)\|^2 \geq d \\
&\quad \Leftrightarrow \|x-c_0\|^2 - 2\varepsilon(c-c_0)^T(x-c_0) + \\
&\qquad \varepsilon^2\|c-c_0\|^2 \geq d \\
&\quad \Leftrightarrow -2\varepsilon(c-c_0)^T(x-c_0)+\varepsilon^2\|c-c_0\|^2 \geq 0 \\
&\qquad (\|x-c_0\|^2 = d \text{ 이므로}) \qquad (**)
\end{aligned}$$

그런데 $\varepsilon^2 \ll \varepsilon$ 이므로, 식(**)이 항상 성립하기 위해서는 $-2\varepsilon(c-c_0)^T(x-c_0) \geq 0$ 이어야 한다. 그런데 $y = x - c_0$, $b = c_0 \cdot y$를 대입하면, 다음 식이 나온다.

$$c \cdot y \leq b \qquad (***)$$

따라서 (*)와 (***)로부터 정리가 성립한다. ■

n 차원에서의 뿔(cone)은 덧셈과, 비음인 스칼라와의 곱에 닫혀있는 집합이다. 예를 들어, PSD 행렬에 비음수를 곱하거나, 두 개의 PSD 행렬을 더하면 역시 PSD행렬이 된다. 이로부터 PSD 행렬의 집합은 뿔(cone)이 됨을 알 수 있다. 또한 뿔은 볼록(convex)하다 : 만약 c 와 c'가 뿔 K에 속하고 $0 < t < 1$이라면, tc와 $(1-t)c'$도 K에 속하고, 따라서 $tc+(1-t)c'$도 K에 속한다. 만약 A가 어떤 벡터들의 집합이라면, A의 대극체(polar)는 $A^o = \{x \mid x \cdot y \geq 0 \ \forall y \in A\}$라고 정의한다. 대극체는 볼록해석학의 유용한 개념 중에 하나이다.

보조정리 A.2.2

i) $A \subseteq B$ 이면 $A^o \supseteq B^o$이다.

ii) $A \subseteq A^{oo}$이다.

iii) A^o는 닫힌 뿔이다.

보조정리 A.2.3 만약 K가 닫힌 뿔이라면, $K = K^{oo}$이다.

증명 : $K \subseteq K^{oo}$이므로, $x \in K^{oo}$이고 $x \notin K$인 x가 존재하지 않음을 보이면 된다. 보조정리 A.2.1에 의해서 x와 K를 분리시키는 초평면 (y, b)가 존재한다. 그런데 K는 뿔이므로 초평면을 $(y, 0)$으로 이동할 수 있다. 그래서 다음 식이 성립한다:

$$\begin{aligned}
&y^T x > 0, \\
&y^T c \leq 0, \ \forall c \in K.
\end{aligned} \qquad (*)$$

따라서, $(-y)^T c \geq 0 \Leftrightarrow -y \in K^o$이 성립한다. 그러나 $x \in K^{oo}$이므로 $(-y)^T x \geq 0$이며 이것은 식 (*)에 모순된다. ■

정리 A.2.4 만약 K와 K'가 닫힌 뿔이라면, $(K \cap K')^o = K^o + K'^o$이다.(A와 B가 벡터들의 집합이라면, $A+B = \{a+b \mid a \in A, \ b \in B\}$라 정의한다.)

증명 : A와 B를 임의 집합이라고 하자. 만약 $x \in A^o + B^o$이고 $y \in A \cap B$이면, $x^T y \geq 0$이다. 그러므로 $A^o + B^o \subseteq (A \cap B)^o$이다. 한편, $A+B \supseteq A$와 $A+B \supseteq B$와 그리고 보조정리 A.2.2에 의해 $(A+B)^o \subseteq A^o$ 이고 $(A+B)^o \subseteq B^o$이다. 그러므로 $(A+B)^o \subseteq A^o \cap B^o$이다. 따라서, $(A^o+B^o)^o \subseteq A^{oo} \cap B^{oo}$도 성립한다. 그러므로 $(A^o+B^o)^{oo} \supseteq (A^{oo} \cap B^{oo})^o$를 얻을 수 있다. 만약 A와 B가 닫힌 뿔이라면 보조정리 A.2.3에 의해 $A^{oo} = A$이고 $B^{oo} = B$이 되어 $A^o + B^o \supseteq (A \cap B)^o$가 된다. ■

정리 A.2.5 만약 K와 K'가 닫힌 뿔이라면, $(K+K')^o = K^o \cap K'^o$이다.

증명 : 보조정리 A.2.4에 의해 유도된다. ■

앞에서 PSD 행렬집합은 뿔이 됨을 관찰하였다. 이 뿔을 P라고 하면 다음을 만족한다.

정리 A.2.6 $P^o = P$

증명 : i) $P \subseteq P^o$: 만약 A와 B가 집합 P에 속한다면, $A = X^T X$, $B = Y^T Y$이다. 또한 $A \cdot B = tr(X^T X Y^T Y) = tr(XY^T YX^T) = (YX^T) \cdot (YX^T) \geq 0$ 이므로 $P \subseteq P^o$가 성립한다.

ii) $P \supseteq P^o$: A가 대칭이나 P에 속하지 않는다고 하자. 그러면, 정리 A.1.2와 A.1.3에 의하여 $A = Q\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)Q^T$로 표시할 수 있으며, A의 고유값들 중 적어도 하나는 음수가 된다. 이 고유값을 λ_1이라고 하자.

한편, $B = Q\mathrm{diag}(1, 0, \cdots, 0)Q^T$로 정의하면, B는 PSD, 즉 P에 속한다. 그러나, $A \cdot B = = \mathrm{tr}A^T B\ \mathrm{tr}Q\mathrm{diag}(\lambda_1, 0, \cdots, 0)Q^T = \mathrm{tr}\mathrm{diag}(\lambda_1, 0, \cdots, 0)Q^T Q$ (왜냐하면, $\mathrm{tr}AB = \mathrm{tr}BA$가 성립) $= \lambda_1 < 0$. 즉 $A \notin P^o$가 성립한다. 따라서 $P \supseteq P^o$이다. ■

따름정리 A.2.7 $A \geq 0 \Leftrightarrow A \cdot B \geq 0\ \forall B \geq 0$. □

따름정리 A.2.8 $A \geq 0, B \geq 0 \Rightarrow (A \cdot B = 0 \Leftrightarrow AB = 0)$

증명 : $(A \cdot B = 0 \Rightarrow AB = 0)$ $A = X^T X$, $B = Y^T Y$로 놓자. 그러면, $A \cdot B = \mathrm{tr}(A^T B) = \mathrm{tr}((YX^T)^T YX^T) = (YX^T) \cdot (YX^T) = 0$. 따라서 $YX^T = 0$이되며, $AB = X^T(XY^T)Y = 0$이 성립한다.

ii) $(AB = 0 \Rightarrow A \cdot B = 0)$ $AB = 0$은 $\mathrm{tr}(AB) = 0$을, 따라서 $\mathrm{tr}(AB) = \mathrm{tr}(A^T B) = A \cdot B = 0$을 의미한다. ■

A.3 Slater 조건

보조정리 A.3.1 S를 임의의 거리가 정의되는 벡터공간(normed linear space), 그리고 L을 S의 선형부분공간(linear subspace)라고 하자. 또한 P를 S의 원점에 꼭지점이 위치한 뿔(cone)이라고 하자. 이 때, P가 내부(interior)를 갖고 있으며 (P의 차원이 S의 차원과 같으며), L과 P의 내부가 교집합을 가지고 있으면, $S = L + P$가 성립한다.

증명 임의의 벡터 $s \in S$를 생각하자. 가정에 의하여 $a \in L \cap \mathrm{int}P$인 벡터 a가 존재한다. a가 P의 내부점에 존재하며, S가 거리가 정의되는 벡터공간이므로 어떤 $\varepsilon > 0$이 존재하여, 다음이 성립한다 : $a + \varepsilon s / \| s \| \in P$. P가 뿔이므로 모든 $\lambda \geq 0$에 대하여 $\lambda(a + \varepsilon s / \| s \|) \in P$가 성립한다. 이때 $\lambda_\varepsilon = \| s \| / \varepsilon$으로 잡으면, $\lambda_\varepsilon(a + \varepsilon s / \| s \|) = \lambda_\varepsilon a + s \in P$가 된다. 따라서 $\lambda_\varepsilon a + s = p$로 표기하면, $s = p + (-\lambda_\varepsilon)a$. $p \in P$이고 $\lambda_\varepsilon a \in L$(왜냐하면 a가 선형부분공간 L에 속하므로)이므로, $s \in L + P$가 성립하며 증명이 끝난다. ■

보조정리 A.3.2 (PSDP)에서 부등식 $\sum_{i=i}^{m} x_i A_i > 0$가 해를 갖는다고 하자. 그러면 임의의 대칭행렬 B에 대하여, $\sum_{i=i}^{m} x_i A_i \geq B$는 가능해를 갖는다.

증명 S를 모든 $n \times n$ 대칭행렬로 이루어진 벡터공간, 그리고 $L = \{ \sum_{i=i}^{m} x_i A_i : x \in R^m \}$이라고 하자. 이때 L은 S의 선형부분공간이 된다. 가정에서 $\sum_{i=i}^{m} x_i A_i > 0$가 해를 가지므로, 또한 S의 선형부분공간 L은 모든 $n \times n$ PSD행렬로 이루어진 뿔 P의 내부점과 만난다. 따라서 보조정리 A.3.1에 의해, 임의의 대칭행렬 $-B$는 L의 원소와 P의 원소 즉 어떤 PSD 행렬 C와의 합으로 표시된다. 다시 말하면, $-B = -\sum_{i=i}^{m} x_i A_i + C$, 또는 $\sum_{i=i}^{m} x_i A_i - B = C \geq 0$가 성립하는 어떤 x가 존재한다는 의미이며, 이것은 곧 (PSDP)의 가능해가

존재함을 의미한다. ■

정리 A.3.3 (Slater 조건) (PDSP) 제약식의 제차 방정식이 내부해를 갖는다고 하자. 즉, $\sum_{i=1}^{m} x_i A_i > 0$이 해를 갖는다고 하자. 그러면, $C=\{(A_1 \cdot Y, A_2 \cdot Y, \cdots, A_n \cdot Y) : Y \geq 0\}$는 닫힌 집합이 된다.

증명 : (스케치) 조건은 $L = \{ \sum_{i=1}^{m} x_i A_i : x \in R^m \}$이 PSD 행렬 뿐, P의 내부와 교차하는 것을 의미한다. 따라서 $L+P$는 $n \times n$ 대칭행렬 전체공간, S이 된다. 따라서 $S^o = (L+P)^o$이 된다. 그러나 $S^o = \{0\}$이며, 정리 A.2.4에 의하여 $(L+P)^o = L^o \cap P^o = L^{\perp} \cap P$(정리 A.2.6)가 된다. 그러나, $L^{\perp} = \{Y : A_1 \cdot Y = 0, A_2 \cdot Y = 0, \cdots, A_n \cdot Y = 0\}$이다. 따라서 P의 원소 중에서 0행렬이 $T(Y) = (A_1 \cdot Y, A_2 \cdot Y, \cdots, A_n \cdot Y)$라는 선형사상(linear transformation)에 의하여 R^m의 0으로 사상되는 유일한 원소가 된다. 이것은 $C = T(Y)$가 닫힌 집합이 되는 충분조건이 된다. ■

참 고 문 헌

[1] Alizadeh, F., "Interior point methods in semidefinite programming with applications to combinatorial optimizations," *SIAM J. Optimization*, 5(1) (1995), pp.13-51.

[2] Alizadeh, F., J.-P. Haeberly and M. Overton, "Primal-dual interior-point methods for semidefinite programming," Technical Report 659, Computer Science Dept, Courant Institute of Math. Sci., New York University, (1994).

[3] Alon, N. and N. Kahale "Approximating the independence number via the θ-function," *Mathematical Programming*, 80 (1998), pp. 253-264.

[4] Anjos, M. and H. Wolkowicz, "A strengthened SDP relaxation via a second lifting for the Max-Cut problem," Research report, University of Waterloo, Waterloo, Ontario, (1999).

[5] Anjos, M. and H. Wolkowicz, "A tight semidefinite relaxation of the cut polytope," Research report, University of Waterloo, Waterloo, Ontario, (2000).

[6] Anstreicher, K., X. Chen, H. Wolkowicz, and Y. Yuan, "Strong duality for a trust-region type relaxation of the quadratic assignment problem," Research Report CORR 98-31, Dept. of Combinatorics and Optimization, University of Waterloo, (1998).

[7] Bazaraa, M.S. and C.M. Shetty, *Nonlinear Programming, Theory and Algorithms*, Wiley, 1979.

[8] Benson, S., Y. Ye, and X. Zhang, "Solving large-scale sparse semidefinite programs for combinatorial optimization," Technical Report, Department of Management Science, University of Iowa, (1997).

[9] Blum, L., M. Shub, and S. Smale "On a theory of computation and complexity over real numbers : NP-completeness, recursive functions and universal machines," *Bulletin of American Mathematical Society*, 21(1) (1989), pp.1-46.

[10] Chor, B. and M. Sudan, "A geometric approach to betweenness," in Proc. of 3rd European Symp. on Algs., volume LNCS 979 of Lecture Notes in Computer Science, (1995), pp.227-237.

[11] de Klerk, E., H.V. Maaren, J.P. Warners, "Relaxations of the satisfiability problem

using semidefinite programming," CWI report, CWI, (1999).

[12] Edmonds, J., "Maximum matching and a polyhedron with 0,1-vertices," *J. Res. Nat. Bureau of Standards*, 69B (1965), pp.125- 130.

[13] Feige, U. and M.X. Goemans, "Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT," in Proc. of 3rd Israel Symp. on Theory of Computing and systems, (1995), pp.182-189.

[14] Frieze, A. and M. Jerrum "Improved approximation algorithms for MAX k-CUT and MAX BISECTION," in Proc. of 4th Integer Programming and Combinatorial Optimization Conference, (1995), pp.1-13.

[15] Garey, M. R. and D. S. Johnson, *Computers and Intractability : A guide to the theory of NP-completeness*, Freeman, (1979).

[16] Goemans, M.X., "Semidefinite programming in combinatorial optimization," *Mathematical Programming*, 79 (1997), pp.143- 162.

[17] Goemans, M.X., "Semidefinite programming and combinatorial optimization," *Documenta Mathematica*, Extra Volume ICM III (1998), pp.657-666.

[18] Goemans, M.X. and F. Rendl, "Semidefinite programs and association schemes," *Computing*, (1999), to appear.

[19] Goemans, M.X. and D.P. Williamson, "0.878 -approximation algorithms for MAX CUT and MAX-2SAT," In Proc. 26th Annual ACM Symposium on Theory of Computing, (1994), pp.422-433.

[20] Goemans, M.X. and D.P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, 42 (1995), pp.1115-1145.

[21] Grotschel, M., L. Lovasz, and A. Schrijver. "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica*, 1(2) (1981), pp.169-197.

[22] Grotschel, M., L. Lovasz, and A. Schrijver. "Polynomial algorithms for perfect graphs," *Annals of Discrete Mathematics*, 21 (1984), pp.325-356.

[23] Grotschel, M., L. Lovasz, and S. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer Verlag, 1988.

[24] Helmberg, C. and F. Rendl, "Solving quadratic (0,1)-problems by semidefinite programs and cutting planes," Preprint, Konrad-Zuse-Zentrum Berlin, (1995).

[25] Helmberg, C., F. Rendl, R.J. Vanderbei, and H. Wolkowicz, "An interior-point method for semidefinite programming," *SIAM J. Optimi.*, 6 (1996), pp.342-361.

[26] Helmberg, C., F. Rendl, and R. Weismantel, "A semidefinite programming approach to the quadratic knapsack problem," Preprint, Konrad-Zuse-Zentrum Berlin, (1996).

[27] Han, Q., Y. Ye, H. Zhang, and J. Zhang, "An improved rounding method and semidefinite programming relaxation for graph partition," Technical Report, Dep't of Management Science, University of Iowa, (2000).

[28] Karger, D., R. Motwani, and M. Sudan. "Approximate graph coloring by Semidefinite programming," *Journal of ACM*, 45 (1998), pp.246-265.

[29] Karmarkar, N., "A new polynomial-time algorithm for linear programming," *Combinatorica*, 4 (1984), pp.373-395.

[30] Lovasz, L., "On the Shannon capacity of a

graph," *IEEE Transactions on Information Theory*, 25(1) (1979).

[31] Lovasz, L. and A. Schrijver, "Cones of matrices and set-functions and 0-1 optimization," *SIAM J. Optimi.*, 1(2) (1991), pp.166-190.

[32] Luo, Z.-Q., J.F. Sturm, and S. Zhang, "Duality results for conic convex programming," Report 9719/A, Econometric Institute EUR, P.O. Box 1738, 3000 DR, The Netherlands, April, (1997).

[33] Mahajan, S. and H. Ramesh, "Derandomizing semidefinite programming based approximation algorithms," In Proc. 36th Symp. on Foundations of Computer Science, (1995), pp.162-169.

[34] Mohar, B. and S. Poljak, "Eigenvalue methods in combinatorial optimization", In Brualdi, R., S. Friedland, and V. Klee eds, *Combinatorial and Graph Thoeretic Problems in Linear Algebra*, volume 50 of The IMA Volumes in Mathematics and its Applications, Springer-Verlag, (1993), pp. 107-151.

[35] Nesterov, Y. and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*, SIAM Studies in Applied Mathematics, Philadelphia, (1993).

[36] Nesterov, Y. and A. Nemirovskii, *Interior point polynomial methods in convex programming : theory and applications*, Society for Industrial and Applied Mathematics, Philadelphia, (1994).

[37] Papadimitriou, C.H. and K. Steiglitz, *Combinatorial Optimization : Algorithms and Complexity*, Prentice Hall, 1982.

[38] Pataki, G., "Cone-LP's and semidefinite programs : Geometry and a simplex-type method," In Proc. IPCO V, Lec. Note on Computer Sciences 1084, Springer, (1996), pp.162-174.

[39] Pataki, G. and L. Tuncel, "On the generic properties of convex optimization problems in conic form," Research Report 97-16, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, September, (1997).

[40] Poljak, S. and F. Rendl, "Nonpolyhedral relaxations of graph-bisection problems," *SIAM J. Optimization*, 5 (1995), pp.467-487.

[41] Ramana, M., "An exact duality theory for semidefinite programming and its complexity implications," DIMACS Tech. Report 95-02R, (1995).

[42] Ramana, M., L. Tuncel, and H. Wolkowicz, "Strong duality for semidefinite programming," *SIAM J. Optimization*, 7 (1997), pp.641-662.

[43] Sherali, H.D. and W.P. Adams, "A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems," *SIAM J. Discrete Mathematics*, 3 (1990), pp.411-430.

[44] Slater, M., "Lagrange Multipliers Revisited : A Contribution to Nonlinear Programming," Cowles Commission Paper, Mathematics 403, (1950).

[45] Srivastav, A. and K. Wolf, "Finding dense subgraphs with semidefinite programming," in *Approximation Algorithms for Combinatorial Optimization*, Jansen, K. and J. Rolim eds., (1998), pp.181-191.

[46] Vandenberghe, L. and S. Boyd, "Semidefinte programming," *SIAM Review*, (1996), pp. 49-95.

[47] Wolkowicz, H. and Q. Zhao, "Semidefinite relaxations for the graph partitioning problem," *Discrete Applied Mathematics*, 96-

97 (1999), pp.461-479.

[48] Ye, Y., "A class of projective transformations for linear programming," *SIAM J. Computing*, 19 (1990), pp.457-466.

[49] Ye, Y., "A .699-approximation algorithm for max-bisection," Technical Report, Department of Management Science, University of Iowa, (1999).

[50] Ye, Y. and J. Zhang, "Approximation of dense-n/2-subgraph and the complement of min-bisection," Technical Report, Department of Management Science, University of Iowa, (1999).

[51] Zhao, Q., S.E. Karisch, F. Rendl, and H. Wolkowicz, "Semidefinite programming relaxations for the quadratic assignment problem," *Journal of Combinatorial Optimization*, 2 (1998), pp.71-109.