Some Applications of Semidefinite Optimization from an Operations Research Viewpoint

Levent Tunçel*

February 2008 (revised: August 2008)

Abstract

This survey paper is intended for the graduate students and researchers who are interested in Operations Research, have solid understanding of linear optimization but are not familiar with Semidefinite Programming (SDP). Here, I provide a very gentle introduction to SDP, some entry points for further look into the SDP literature, and brief introductions to some selected well-known applications which may be attractive to such audience and in turn motivate them to learn more about semidefinite optimization.

Keywords: semidefinite programming, operations research

AMS Subject Classification: 90C22, 90-02

*(ltuncel@math.uwaterloo.ca) Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. Research of the author was supported in part by a Discovery Grant from NSERC.

1 Introduction, Motivation and Some Definitions

Linear optimization is one of the most fundamental tools in operations research. This is so, not only in the theory and mathematics of operations research but also in applications. *Semidefinite Optimization* can be seen as a very powerful and elegant generalization of linear optimization.

Linear optimization has had tremendous success since the birth of operations research. We have gotten very good at formulating real-world problems as Linear Programming (LP) problems when justified, and the improvements in software/hardware combinations (but especially in software due to the advances in LP theory) made it possible to solve very large LP problems routinely. LP problems of the largest size that we can solve on our lap-top computers today would have been considered completely out of reach three decades ago (on *any* computer available back then). Due to the fast advances in science and technology and rapid distribution of newly acquired knowledge as well as very wide accessibility of this knowledge, we see nowadays that those in charge of main decision making functions in many applications (manufacturing, service, government, etc.) are more and more willing to utilize sophisticated operations research techniques. This situation allows for not only LP but also other tools such as integer programming, mixed integer programming, nonlinear programming to find wider and wider acceptance by practitioners.

Since the late 1980's and the early 1990's, as the modern interior-point revolution was sweeping through the optimization community, Semidefinite Optimization began emerging as the next fundamental tool in operations research. At the time of this writing, there is a wide array of algorithms, freely available software and wide array of scientific papers on the theory and applications of Semidefinite Optimization. While there are many parallels between the "history of LP" and "history and expected future of SDP," there are also some very significant differences.

Consider a typical Linear Programming problem. Let $A \in \mathbb{R}^{m \times n}$ represent the given coefficient matrix and assume that the objective function vector $c \in \mathbb{R}^n$ and the right-hand-side vector $b \in \mathbb{R}^m$ are also given. Our primal problem, written in a standard equality form is (we will refer to it as (LP)):

$$\begin{array}{rrcl} \min & c^T x & & \\ & Ax & = & b, \\ & x & \geq & 0. \end{array}$$

Its dual (LD) is defined to be

$$\begin{array}{rrcl} \max & b^T y & & \\ & A^T y + s & = & c, \\ & s & \geq & 0. \end{array}$$

We denote by $\mathbb{S}^n$, the space of n-by-n symmetric matrices with real entries. $X \in \mathbb{S}^n$ is called *positive semidefinite* if

$$h^T X h \geq 0, \forall h \in \mathbb{R}^n.$$

We denote by $\lambda_j(X)$ the eigenvalues of X. Note that every eigenvalue of every $X \in \mathbb{S}^n$ is real. We index the eigenvalues so that

$$\lambda_1(X) \geq \lambda_2(X) \geq \cdots \geq \lambda_n(X).$$

We have

Proposition 1.1. *Let $X \in \mathbb{S}^n$. Then, the following are equivalent:*

(a) X is positive semidefinite;

(b) $\lambda_j(X) \geq 0, \forall j \in \{1, 2, \ldots, n\}$;

(c) there exists $B \in \mathbb{R}^{n \times n}$ such that $X = BB^T$;

(d) for every nonempty $J \subseteq \{1, 2, \ldots, n\}$, $\det(X_J) \geq 0$, where $X_J := [X_{ij} : i, j \in J]$.

We denote the set of positive semidefinite matrices by $\mathbb{S}^n_+$. We call the submatrices of X described in part (d) of the above proposition, *symmetric minors of X*. $X \in \mathbb{S}^n$ is called positive definite if

$$h^T X h > 0, \forall h \in \mathbb{R}^n \setminus \{0\}.$$

In the next result, we give characterizations of positive definiteness analogous to those in Proposition 1.1.

Proposition 1.2. *Let $X \in \mathbb{S}^n$. Then, the following are equivalent:*

(a) X is positive definite;

(b) $\lambda_j(X) > 0, \forall j \in \{1, 2, \ldots, n\}$;

(c) there exists $B \in \mathbb{R}^{n \times n}$ nonsingular, such that $X = BB^T$;

(d) for every $J_k := \{1, 2, \ldots, k\}$, $k \in \{1, 2, \ldots, n\}$, $\det(X_J) > 0$.

We denote the set of symmetric positive definite matrices over reals by $\mathbb{S}^n_{++}$. Using Proposition 1.1 part (c) and Proposition 1.2 part (c), we deduce

$$\mathbb{S}^n_{++} = \left\{ X \in \mathbb{S}^n_+ : X \text{ is nonsingular} \right\}.$$

For $U, V \in \mathbb{S}^n$, we write $U \succeq V$ to mean $(U - V) \in \mathbb{S}^n_+$ and $U \succ V$ to mean $(U - V) \in \mathbb{S}^n_{++}$.

Another fact about the positive semidefinite matrices that is useful in modeling SDP problems is the *Schur Complement Lemma:*

Lemma 1.1. *Let $X \in \mathbb{S}^n$, $U \in \mathbb{R}^{n \times m}$ and $T \in \mathbb{S}^m_{++}$. Then*

$$M := \begin{bmatrix} T & U^T \\ U & X \end{bmatrix} \succeq 0, \iff \left(X - UT^{-1}U^T \right) \succeq 0.$$

We call the matrix $\left(X - UT^{-1}U^T\right)$, the *Schur Complement of T in M* in the above notation.

Note that in our standard form (LP) and (LD), the constraints $x \geq 0$, $s \geq 0$ mean that x and s lie in $\mathbb{R}^n_+$ which is a *convex cone* (that is, for every positive scalar α, and for every element x of the set, αx is also in the set and, for every pair of elements u, v of the set, $(u+v)$ also lies in the set). In particular, this convex cone is just a direct sum of nonnegative rays:

$$\mathbb{R}^n_+ = \mathbb{R}_+ \oplus \mathbb{R}_+ \oplus \cdots \oplus \mathbb{R}_+.$$

Replacing $\mathbb{R}_+$ by more general convex cones (which contain $\mathbb{R}_+$ as a special case) we can generalize (LP). In SDP, we replace $\mathbb{R}_+$ by $\mathbb{S}^{n_i}_+$ for some $n_i \geq 1$. (Note that for $n_i = 1$, we have $\mathbb{S}^{n_i}_+ = \mathbb{R}_+$.) Thus, in this more general optimization problem we write our variable x as a symmetric matrix (possibly with a block diagonal structure) and replace the constraint "$x \geq 0$" by

$$X \in \mathbb{S}^{n_1}_+ \oplus \mathbb{S}^{n_2}_+ \oplus \cdots \oplus \mathbb{S}^{n_r}_+.$$

Clearly, setting $n_1 := n_2 := \cdots := n_r := 1$ and $r := n$ takes us back to (LP) as a special case.

Another very interesting special case is given by the cone:

$$SO^n := \left\{ \begin{pmatrix} x_0 \\ x \end{pmatrix} \in \mathbb{R} \oplus \mathbb{R}^n : x_0 \geq \|x\|_2 \right\}.$$

This cone is known by many names in the literature: *second order cone*, *Lorentz cone*, and even sometimes as the *ice-cream cone*.

First notice that

$$SO^n := \operatorname{cl} \left\{ \begin{pmatrix} x_0 \\ x \end{pmatrix} \in \mathbb{R} \oplus \mathbb{R}^n : x_0 > \|x\|_2 \right\},$$

where $\operatorname{cl}(\cdot)$ denotes the closure. Secondly, using the Schur Complement Lemma, we see that

$$\begin{pmatrix} x_0 \\ x \end{pmatrix} \in SO^n \iff \begin{bmatrix} x_0 & x^T \\ x & x_0 I \end{bmatrix} \succeq 0.$$

Let $\mathcal{A}_n : \mathbb{R}^{n+1} \to \mathbb{S}^{n+1}$ denote the linear operator satisfying

$$\mathcal{A}_n \begin{pmatrix} x_0 \\ x \end{pmatrix} = \begin{bmatrix} x_0 & x^T \\ x & x_0 I \end{bmatrix}.$$

Then,

$$\left(x^{(1)}, x^{(2)}, \ldots, x^{(r)}\right) \in SO^{n_1} \oplus SO^{n_2} \oplus \cdots \oplus SO^{n_r}$$

iff

$$\left(\mathcal{A}_{n_1}\left(x^{(1)}\right), \mathcal{A}_{n_2}\left(x^{(2)}\right), \ldots, \mathcal{A}_{n_r}\left(x^{(r)}\right)\right) \in \mathbb{S}^{n_1+1}_+ \oplus \mathbb{S}^{n_2+1}_+ \oplus \cdots \oplus \mathbb{S}^{n_r+1}_+.$$

Thus, we arrived at the *Second Order Cone Programming (SOCP)* problems and showed how SOCP is a special case of SDP. The cones SO^n and $\mathbb{S}^n_+$ belong to a family of extremely well-behaved convex cones called *symmetric cones*. (From this family, SO^n and $\mathbb{S}_+^n$ are the only two with wide applications in Operations Research so far, the next one in line is the cone of hermitian positive semidefinite matrices with complex entries.) For further information on the algebraic and analytic structure of symmetric cones, see Faraut and Korányi [20].

From now on, we will usually write $X \in \mathbb{S}_+^n$, $X \succeq 0$, etc.; however, depending on the context we might mean

$$X \in \mathbb{S}_+^{n_1} \oplus \mathbb{S}_+^{n_2} \oplus \cdots \oplus \mathbb{S}_+^{n_r}, \text{ where } n_1 + n_2 + \cdots + n_r = n.$$

For our general discussions, we will represent the data for SDP by $C, A_1, A_2, \ldots, A_m \in \mathbb{S}^n$ and $b \in \mathbb{R}^m$ such that our primal SDP is

$$\begin{array}{llrcl} (P) & \inf & \langle C, X \rangle & & \\ & & \langle A_i, X \rangle & = & b_i, \quad \forall i \in \{1, 2, \ldots, m\} \\ & & X & \succeq & 0. \end{array}$$

In the above optimization problem, the inner product is the *trace inner product*. I.e., for $U, V \in \mathbb{R}^{n \times n}$,

$$\langle U, V \rangle := \mathrm{Tr}\left(U^T V\right).$$

We define the dual of (P) as

$$\begin{array}{llrcl} (D) & \sup & b^T y & & \\ & & \sum_{i=1}^{m} y_i A_i & \preceq & C. \end{array}$$

Or, equivalently

$$\begin{array}{llrcl} (D) & \sup & b^T y & & \\ & & \sum_{i=1}^{m} y_i A_i + S & = & C, \\ & & S & \succeq & 0. \end{array}$$

In Section 2, we mention some generalizations of duality theorems from LP to SDP setting. In Section 3, we briefly explain a basic complexity result for SDP based on interior-point methods. The remaining sections (Sections 4–12) are geared towards making connection to SDP from various application areas. For a more detailed introduction to the applications of SDP, the reader might want to start with the following references: Alizadeh [2], Ben-Tal and Nemirovskii [7], Boyd et al. [13], Boyd and Vandenberghe [14], Goemans [21, 22], Nesterov and Nemirovskii [48], Todd [69], Tunçel [70] and the Handbook of SDP [75].

2 A Whiff of Duality Theory

Well-known *weak duality relation* easily generalizes to the SDP setting under our definitions.

Theorem 2.1. *(Weak Duality Theorem) Let $A_1, A_2, \ldots, A_m, C \in \mathbb{S}^n$ and $b \in \mathbb{R}^m$ be given.*

1. *For every X feasible in (P) and every (y, S) feasible in (D), we have*
$$\langle C, X\rangle - b^T y = \langle X, S\rangle \geq 0.$$
2. *If X is feasible in (P), (y, S) is feasible in (D) and $\langle C, X\rangle = b^T y$, then X is optimal in (P) and (y, S) is optimal in (D).*

Unlike linear programming, (P) having an optimal solution does *not* guarantee the same for its dual (D). In particular,

- (P) and (D) both having optimal solutions does *not* guarantee that their optimum objective values are the same.
- Optimum objective value of (P) being finite does *not* guarantee that it is attained.

There are many approaches to remedy these situations. See Borwein and Wolkowicz [12], Ramana [56], Ramana, Tunçel and Wolkowicz [57], Tunçel [71], Pataki [51], Polik and Terlaky [55], Tunçel and Wolkowicz [72], Wolkowicz [74].

We will cover a relatively simple and elegant approach involving *interior points* with respect to the cone constraints (or *strictly feasible points*). $\bar{X} \in \mathbb{S}^n_{++}$ such that $\mathcal{A}(\bar{X}) = b$ is called *a Slater point of* (P). Similarly, feasible solutions $(\bar{y}, \bar{S})$ of (D) with the property that $\bar{S} \in \mathbb{S}^n_{++}$ (or equivalently $\sum_{i=1}^m y_i A_i \prec C$) are called *Slater points of* (D).

If such feasible points exist for both (P) and its dual (D) then as far as the statements of the main duality theorems are concerned, we are back to an LP-like situation:

Theorem 2.2. *(Strong Duality Theorem) Suppose (P) and (D) both have Slater points. Then (P) and (D) both have optimal solutions and their optimum objective values coincide.*

Now, let us consider Farkas' Lemma for the LP setting:

Lemma 2.1. *Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ be given. Then exactly one of the following systems has a solution:*

(I) $Ax = b, \quad x \geq 0;$

(II) $A^T y \leq 0, \quad b^T y > 0.$

In many ways, Farkas' Lemma is essentially equivalent to LP (strong) duality theorems. Therefore, it should not be a big surprise to us that generalizations of Farkas' Lemma to the SDP setting involve some additional work and/or modifications of the original statements (more than a straight-forward translations of the terms).

Let $A_1, A_2, \ldots, A_m \in \mathbb{S}^n$ and $b \in \mathbb{R}^m$ be given. We say that the system

$$\{\langle A_i, X\rangle = b_i, \forall i \in \{1, 2, \ldots, m\}; \quad X \succeq 0\}$$

is *almost feasible*, if for every $\epsilon > 0$, there exists $\bar{b} \in \mathbb{R}^m$ such that $\|b - \bar{b}\|_2 < \epsilon$ and $\left\{\langle A_i, X\rangle = \bar{b}_i, \forall i \in \{1, 2, \ldots, m\}; \quad X \succeq 0\right\}$ is feasible.

Now, we are ready to describe a generalization of Farkas' Lemma to the SDP setting.

Lemma 2.2. *Let $A_1, A_2, \ldots, A_m \in \mathbb{S}^n$, $b \in \mathbb{R}^m$ be given. Then exactly one of the following two statements holds:*

(I) $\langle A_i, X\rangle = b_i,\ \forall i \in \{1, 2, \ldots, m\},\quad X \succeq 0$ is almost feasible;

(II) there exists $y \in \mathbb{R}^m$ such that $\sum_{i=1}^{m} y_i A_i \preceq 0, \quad b^T y > 0$.

3 A Glimpse of Theory of Algorithms for Convex Optimization

Currently the most popular algorithms to solve SDPs are interior-point algorithms. These algorithms start with $X^{(0)} \succ 0$, $S^{(0)} \succ 0$ and some suitable $y^{(0)} \in \mathbb{R}^m$ and generate a sequence $\left\{X^{(k)}, y^{(k)}, S^{(k)}\right\}$ approaching the set of optimal solutions of (P) and (D) (if they exist, and if there is no duality gap). Note that neither $X^{(0)}$ nor $\left(y^{(0)}, S^{(0)}\right)$ is required to be feasible.

Next, we give a sample convergence theorem to give the reader some understanding of what it means to "solve" SDPs in the nicer case that $X^{(0)}$ and $\left(y^{(0)}, S^{(0)}\right)$ are feasible in their respective problems. See for instance, Nesterov and Todd [49], Tunçel [70], Ye [76].

Theorem 3.1. *Let $X^{(0)}$ feasible in (P) and $\left(y^{(0)}, S^{(0)}\right)$ feasible in (D) be given such that $X^{(0)} \in \mathbb{S}^n_{++}$ and $S^{(0)} \in \mathbb{S}^n_{++}$ and*

$$n \ln\left(\frac{\langle X^{(0)}, S^{(0)}\rangle}{n}\right) - \ln\left(\det\left(X^{(0)}\right)\right) - \ln\left(\det\left(S^{(0)}\right)\right) \leq \sqrt{n}\ln\left(\frac{1}{\epsilon}\right), \text{ for some } \epsilon \in (0,1).$$

Then, there exist many interior-point algorithms which deliver in $O\left(\sqrt{n}\ln\left(\frac{1}{\epsilon}\right)\right)$ iterations $\bar{X}$ feasible in (P), $\left(\bar{y}, \bar{S}\right)$ feasible in (D) such that

$$\langle \bar{X}, \bar{S}\rangle \leq \epsilon\langle X^{(0)}, S^{(0)}\rangle.$$

In the above, ϵ can be (and should be) chosen by the user. The technical condition in the statement of the theorem measures the *distance* (roughly speaking) from the initial point $\left(X^{(0)}, y^{(0)}, S^{(0)}\right)$ to the boundary of the corresponding feasible regions. It really measures the distance to the so-called *central path* (or more precisely the distance to the most central point with the same duality gap—at this most central point the measure takes the value zero, and is positive elsewhere; as we approach the boundary while keeping the duality gap constant, the measure tends to $+\infty$).

We can use different epsilons, e.g., ϵ_1 for the "proximity-to-the-central-path measure" above and ϵ_2 for the duality gap condition on the final solution $\left(\bar{X}, \bar{y}, \bar{S}\right)$. Then the iteration complexity of the algorithms can be upperbounded by

$$O\left(\sqrt{n}\left(\ln\left(\frac{1}{\epsilon_1}\right) + \ln\left(\frac{1}{\epsilon_2}\right)\right)\right).$$

Each iteration involves solution of linear systems of equations with dimensions $O(n^2)$. Forming the linear systems usually require nontrivial linear algebra which should be done very carefully balancing the issues of sparsity, speed, memory requirements and accuracy. (These can vary wildly from one application to the next.)

Theoretically speaking, convergence theorems based on ellipsoid method and some first-order methods are generally stronger in the sense that they are based on *black box* type models (e.g., *polynomial-time separation oracles*). See for instance Nemirovskii and Yudin [43] and Tunçel [70].

Generally speaking, interior-point methods are second order methods, and in practice, for a given instance, if we can perform one iteration of the interior-point algorithm in a reasonable time, then it usually means that we can solve the problem at hand to a decent accuracy in a reasonable time.

If an iteration of the interior-point algorithm is too expensive for the given application and we do not require a lot of accuracy, then first-order methods might be a perfect fit for the application at hand. See Nesterov [44, 45, 46, 47], Nemirovskii [42].

4 Objective Function: Maximize a Linear Function or the Determinant?

We defined the SDP problem as optimizing a linear function of a matrix variable subject to finitely many linear equations, inequalities and semidefiniteness constraints. However, in many applications, we would like to find the most *central* solution. Here, the term *central* is usually defined by a strictly convex barrier function. By *barrier*, we mean it is the kind of function which takes finite values in the interior of the cone and whose values tend to $+\infty$ along every sequence in the interior of the cone converging to a point on the boundary of the cone (for instance the function $-\ln(\det(X))$ for the cone of positive semidefinite matrices). A typical example is the problem of the form:

$$\begin{array}{rrcl} \min & -\ln(\det(X)) & & \\ & \mathcal{A}(X) & = & b, \\ & X & \succ & 0, \end{array}$$

or of the form:

$$\begin{array}{rrcl} \min & -\ln(\det(S)) & & \\ & \mathcal{A}^*(y) + S & = & C, \\ & S & \succ & 0. \end{array}$$

In this survey paper, we also mention convex optimization problems of the above type. Even though strictly speaking they are not SDPs, they can be written as SDPs (the epigraph of the objective function, $-\ln(\det(\cdot))$, can be represented as the feasible region of an SDP). Moreover, the most popular algorithms used to solve such problems are closely related to those that solve SDPs.

We do not rewrite the above convex optimization problems as SDPs, because currently, in almost all applications, it is better to solve them directly in the above form without using the SDP representation of the epigraph of $-\ln(\det(\cdot))$.

5 Scheduling

Scheduling is such an important part of Operations Research that almost all leading university programs in Operations Research have a whole course dedicated solely to it. Here, we will mention an application to one family of parallel machine scheduling problems (many others exist). Namely, $Rm\ /\ \ /\ F_w$ that is, minimizing the total weighted flow time in a very basic, parallel, unrelated machine environment. We have m parallel machines and n independent jobs that can be performed on any of the machines. The given data (which is assumed to be deterministic) are

$$p_{ij} \text{ the processing time of job } j \text{ on machine } i, \forall i \in \{1, 2, \ldots, m\}, \forall j \in \{1, 2, \ldots, n\}$$

and the weights

$$w_j,\ j \in \{1, 2, \ldots, n\}.$$

All jobs are available for processing at time 0. Preemption is not allowed and we assume that the machines do not fail during the course of the schedule. Given a feasible schedule, let C_j denote the completion time of job j in the given schedule. Then the objective function value of this schedule is given by

$$F_w = \sum_{j=1}^{n} w_j C_j.$$

Among all feasible schedules, we want to find the one with minimum flow time (F_w).

The first fact I would like to mention about this problem is that it is $\mathcal{NP}$-hard even when there are only two machines (i.e., $m = 2$). See for instance, Bruno, Coffman and Sethi [15], Lenstra, Rinooy Kan and Brucker [32], as well as Pinedo [53] and the references therein. The single machine special case ($m = 1$) has a very well-known $O(n \log(n))$ algorithm (known as *Smith's rule* [66]), sort the jobs with respect to WSPT:

$$\frac{w_{[1]}}{p_{[1]}} \geq \frac{w_{[2]}}{p_{[2]}} \geq \cdots \geq \frac{w_{[n]}}{p_{[n]}}.$$

Since the problem is $\mathcal{NP}$-hard, we will focus on polynomial-time approximation algorithms. First, let us consider a special case of $Rm\ /\ \ /\ F_w$ in terms of the machine environment. Instead of working with m unrelated machines, suppose that all of our m machines are identical. That is, we have the problem: $Pm\ /\ \ /\ F_w$. This special case is still $\mathcal{NP}$-hard. However, a very simple and extremely fast heuristic, *list scheduling with a list in WSPT order* yields a $\frac{1}{2}\left(1+\sqrt{2}\right)$-approximation algorithm, a result of Kawaguchi and Kyan [26].

Now, we are ready to go over an algorithm proposed by Skutella [65] which uses Semidefinite Optimization (more precisely speaking, it is a very special case of it, only one second order cone constraint away from linear programming). Let us split the construction of an optimal schedule for $Rm\ /\ \ /\ F_w$ into two stages:

1. Find an optimal assignment of the jobs to the machines;
2. given some optimal assignment of the jobs to the machines, construct an optimal schedule.

By Smith's rule, the second stage can be easily implemented in $O(n\log(n))$ time. Therefore, it suffices to focus on finding an optimal assignment. Moreover, it is not hard to imagine that "approximately optimal" assignments would yield "near optimal" schedules under the above scheme.

Let

$$x_{ij} := \begin{cases} 1, & \text{if job } j \text{ is assigned to machine } i; \\ 0, & \text{otherwise.} \end{cases}$$

We write x as a mn vector as follows. For each machine $i \in \{1,2,\ldots,m\}$, order the elements of x_{ij} with respect to WSPT, break the ties by the initial indexing of the jobs. This gives a total order. Let

$$h_{ij} := w_j p_{ij},\ \forall i \in \{1,2,\ldots,m\}, \forall j \in \{1,2,\ldots,n\}.$$

We order the elements of vector $h \in \mathbb{R}^{mn}$ with respect to the total order described above. We denote the underlying order for machine i by

$$\cdot \rightarrow_i \cdot$$

so that job j precedes job k on machine i, in this order iff

$$j \rightarrow_i k.$$

We also define $H \in \mathbb{S}^{mn}$ as follows

$$H_{(i,j),(k,\ell)} := \begin{cases} w_\ell p_{ij}, & \text{if } i=k \text{ and } j \rightarrow_i \ell; \\ w_j p_{i\ell}, & \text{if } i=k \text{ and } \ell \rightarrow_i j; \\ 0, & \text{otherwise.} \end{cases}$$

It is clear that the matrix $(H + \text{Diag}(h))$ is block diagonal with m blocks (one for each machine). Moreover, it is easy to establish (using elementary row/column operations) that the determinant of the ith block is

$$w_n p_{i1} \prod_{i=1}^{m} (w_1 p_{i2} - w_2 p_{i1})(w_2 p_{i3} - w_3 p_{i2}) \cdots (w_{n-1} p_{in} - w_n p_{i,n-1}).$$

Note that the symmetric minors of these blocks also have their determinants expressed by this formula. By construction, for each i, we have

$$(w_1 p_{i2} - w_2 p_{i1}) \geq 0, (w_2 p_{i3} - w_3 p_{i2}) \geq 0, \cdots, (w_{n-1} p_{in} - w_n p_{i,n-1}) \geq 0.$$

Therefore, (recalling Proposition 1.1) we conclude the following observation of Skutella:

Lemma 5.1. *Let h and H be as given above. Then*

$$H + \text{Diag}(h) \succeq 0.$$

Let $B \in \mathbb{R}^{n \times n}$ be such that $BB^T = H + \text{Diag}(h)$ (since $(H + \text{Diag}(h)) \succeq 0$, such B exists—use for instance a Cholesky or LDL^T decomposition). Now, consider the following convex optimization problem:

$$\begin{array}{lrcll} \min & z & & & \\ & \|B^T x\|_2^2 + h^T x - 2z & \leq & 0, & \\ & h^T x - z & \leq & 0, & \\ & \bar{e}^T x_{:j} & = & 1, & \forall j \in \{1, 2, \ldots, n\}, \\ & x & \in & \mathbb{R}^{nm}_+. & \end{array}$$

In the above, we used MATLAB-like notation to refer to the parts of x vector corresponding to different jobs and we denoted by $\bar{e}$ the vector of all ones of appropriate size. Note that the above optimization problem can be expressed as a conic convex optimization problem in the form of our generic primal problem where all the cone constraints are nonnegativity constraints except one which is a second order cone constraint.

In many applications of SDP, the real application problem we are dealing with is approximated by a SDP. For instance, we have a combinatorial optimization problem and we work with an SDP relaxation. So, let (P) be such a relaxation and let z denote the optimum value of the real application. Then any feasible solution $\bar{X}$ of (P) gives us

$$\langle C, \bar{X} \rangle \leq z.$$

If we manage to use $\bar{X}$ to generate a feasible solution of the real problem, denoting the objective function value of the generated feasible solution by $\hat{z}$, we have

$$\langle C, \bar{X} \rangle \leq z \leq \hat{z}.$$

Skutella proves that an approximately optimal solution of the above second order cone optimization problem can be used in a simple randomized rounding scheme to obtain a polynomial-time approximation algorithm with a good approximation ratio. In particular, de-randomizing such an algorithm Skutella obtains:

Theorem 5.1. *(Skutella [65]) Solving the above second order cone optimization problem approximately and using the derandomized rounding algorithm on the approximately optimal solution, we can obtain in polynomial time, a feasible schedule for the given instance of Rm / / F_w whose objective function value is within a factor of $\frac{3}{2}$ of the optimal objective value.*

6 Facility Location and Data Mining

Facility Location has been one of the fundamental topics in operations research since the birth of the discipline. More recently, another area *Data Mining* emerged as a very important and fundamental topic. The latter is due to the tremendous increase in information gathering-storing-processing capabilities.

From an optimization viewpoint these two fundamental topics (even though one is classical and the other is modern) share the same structure at a very basic level. Let us consider the following basic problems.

Basic k-Facility Location: We have customers (or main sale-stores/warehouses) located at the points $a^{(1)}, a^{(2)}, \ldots, a^{(n)} \in \mathbb{R}^d$. We would like to optimally determine the location of k factories (to be opened) to minimize the total costs (or maximize the total profit).

Basic Clustering: We have data points $a^{(1)}, a^{(2)}, \ldots, a^{(n)} \in \mathbb{R}^d$. We would like to optimally determine the allocation of these points to k sets to minimize the proximity of points to each other within the same cluster.

In the basic clustering problems, the points $a^{(1)}, a^{(2)}, \ldots, a^{(n)}$ usually represent quantitative counterparts of features of each item with respect to certain attributes. The goal is to separate the points into k mutually disjoint groups with respect to their similarities in these quantitative scores. The points assigned to the same cluster should correspond to the items that have the same characteristics and the points assigned to different clusters should have some significant dissimilarities in some attributes.

One way to attack this clustering problem is to assign a *center* $z^{(i)}$ for each cluster i, design a suitable measure of distance $\|a^{(j)} - z^{(i)}\|$ and minimize a total cost function based on these distances.

In the basic k-facility location problem, the objective is very similar to the above situation. We want to decide on the location of factories (*centers*) $z^{(i)}$, $i \in \{1, 2, \ldots, k\}$ under a suitable measure of distance $\|a^{(j)} - z^{(i)}\|$ (note that the distance function corresponding to the pair i, j may involve quantities to be shipped in addition to the cost of shipping) and minimize a total cost function based on these distance functions.

For the rest of the section, we will use the data mining terminology. For connections to facility location problems and some interesting computational experiments, see for instance, Zvereva [77]. Similar problems also arise in the exciting application area of image segmentation.

Suppose we chose some functions $f_{j\ell}$ to measure the closeness of pairs of points. E.g., a popular choice is to have

$$f(a^{(j)}, a^{(\ell)}) := \exp\left(-\frac{\|a^{(j)} - a^{(\ell)}\|_2}{\sigma}\right),$$

for some positive σ. Note that in this particular example, the function $f_{j\ell}$ is the same function for all pairs. Then one defines the *affinity matrix* $W \in \mathbb{S}^n$ as

$$W_{j\ell} := f(a^{(j)}, a^{(\ell)}), \forall j, \ell.$$

With these notations, Peng and Wei [52] work with the following SDP relaxation for the underlying clustering problem:

$$\begin{array}{rrcl} \inf & \langle W, I - X \rangle & & \\ & X\bar{e} & = & \bar{e}, \\ & \langle I, X \rangle & = & k, \\ & X & \geq & 0, \\ & X & \succeq & 0. \end{array}$$

Note that in the above SDP problem every entry of the variable matrix X is required to be nonnegative.

This is a relaxation, because X does not have to have every one of its entries 0,1 (this could be enforced by a constraint $X^2 = X$; but then we no longer have an SDP). At the price of increasing the complexity of the relaxation, we can tighten it further. One idea may be to add the constraint

$$\begin{bmatrix} I & X \\ X & X \end{bmatrix} \succeq 0.$$

The latter is equivalent to $X^2 \preceq X$ (Schur complement of I in the above block matrix; we used Lemma 1.1). Now, notice that $X \preceq I$ is already implied by the SDP relaxation. Suppose $X \succ 0$. Apply the automorphism $X^{1/2} \cdot X^{1/2}$ to both sides of $X \preceq I$. Then we deduce

$$X^2 \preceq X \iff X \preceq I.$$

Note that the constraints

$$X\bar{e} = \bar{e}, \quad X \geq 0, \quad X \succeq 0$$

already imply $X \preceq I$. Therefore, adding the constraint $\begin{bmatrix} I & X \\ X & X \end{bmatrix} \succeq 0$ does not help tighten the relaxation.

Another approach to data mining tries to classify the given points $a^{(1)}, a^{(2)}, \ldots, a^{(n)}$ by utilizing minimum volume ellipsoids, see Shioda and Tunçel [64]. In such an approach, we try to classify points into k clusters such that a global measure of the volumes of the k minimum volume ellipsoids is minimized. An interesting objective function yielding such a global measure is

$$-\sum_{i=1}^{k} \ln(\det(X^{(i)})),$$

where $X^{(i)}$ is the positive definite matrix determining the size and shape of the minimum volume ellipsoid containing all points assigned to cluster i. Or, one might prefer an objective function of the form

$$-\sum_{i=1}^{k} w_i \ln(\det(X^{(i)})),$$

for given positive weights w_i. A much needed subroutine in attacking these problems is an algorithm which takes a set of points as input and returns the minimum volume ellipsoid containing all these given points. Unlike the main clustering problem, for the latter problem we have many efficient algorithms; moreover, the problem can be formulated as:

$$\begin{array}{lrcl} \min & -\ln(\det(X)) & & \\ & \left(a^{(j)}\right)^T X a^{(j)} - 2\left(A^{(j)}\right)^T u + z & \leq & 1, \quad \forall j \in \{1,2,\ldots,n\}; \\ & \begin{bmatrix} z & u^T \\ u & X \end{bmatrix} & \succ & 0. \end{array}$$

Now suppose we are interested in finding a good and simple inner-approximation to the feasible region of an LP problem. Suppose that the feasible region of the LP is bounded and is given as

$$P := \{x \in \mathbb{R}^d : Ax \leq b\},$$

where $a^{(1)}, a^{(2)}, \ldots, a^{(m)}$ denote the columns of A. Then the problem of finding the maximum volume ellipsoid contained in P can be solved by solving the following determinant maximization problem:

$$\begin{array}{lrcl} \min & -\ln(\det(X)) & & \\ & \|Xa^{(i)}\|_2 + \left(a^{(i)}\right)^T u & \leq & b_i, \quad \text{for all } i \in \{1,2,\ldots,m\} \\ & X \succeq 0, & & u \in \mathbb{R}^d. \end{array}$$

Another typical example is from Optimal Experiment Design:

$$\begin{array}{lrcl} \min & -\ln(\det(X)) & & \\ & X - \sum_{i=1}^{n} u_i h^{(i)} \left(h^{(i)}\right)^T & = & 0, \\ & \bar{e}^T u & = & 1, \\ & u \in \mathbb{R}^n_+, & & X \succeq 0, \end{array}$$

where $h^{(i)} \in \mathbb{R}^d$ are test vectors ($n > d$ and $\left\{h^{(i)} : i \in \{1, 2, \ldots, n\}\right\}$ contains a basis) and the objective function tries to minimize the determinant of the error covariance matrix of the experiment.

Instead, if we prefer to minimize the norm of the error covariance matrix, then the objective function would be:

$$\inf \ \lambda_1(X^{-1}).$$

Adding a new variable $z \in \mathbb{R}$, this latter problem becomes an SDP:

$$\begin{array}{lrcl} \sup & & z & \\ & zI - \sum_{i=1}^{n} u_i h^{(i)} \left(h^{(i)}\right)^T & \preceq & 0, \\ & \bar{e}^T u & = & 1, \\ & u \in \mathbb{R}^n_+. & & \end{array}$$

For further details and many other similar applications, see Vandenberghe, Boyd and Wu [73] and the references therein.

The above models for optimal experiment design are called *single-response models*. In practice, there are many applications requiring multiple (and possibly correlated) measurements of responses. Such models are called *multi-response models*. For an extension of the above SDP approach to multi-response models, see Atashgah and Seifi [4].

7 Max Cut and VLSI Design

Given a simple undirected graph G with node set V and edge set E, any subset U of the node set define a *cut* in the graph. The *shores* of the cut defined by U are: U and $V \setminus U$. The set of edges that go between the two shores of a cut are the edges that are cut (we denote this set of edges by $\delta(U)$ which is equal to $\delta(V \setminus U)$ since the graph is undirected).

We may be interested in the maximum cardinality cut in a graph; i.e., maximizing $|\delta(U)|$ where $U \subseteq V$. Or, more generally, given a weight function (on the edges of G) $w \in \mathbb{R}^E$, we define the weight of a cut U by

$$\sum_{e \in \delta(U)} w_e$$

and ask for a cut in G of maximum weight. We just defined the *Max Cut problem*.

This problem has been very extensively studied in Combinatorial Optimization. A very comprehensive reference is the book by Deza and Laurent [18].

Classical applications of Max Cut problem to VLSI circuit design go back at least to Chen et al. [16] and Pinter [54]. For a good exposure, see Barahona [6].

Max Cut problem played a quite special role in SDP. The techniques used on it (including those based on SDP) become well-known tools in Combinatorial Optimization, Theoretical Computer Science and many other areas outside Continuous Optimization. The breakthrough result of Goemans and Williamson [23] was the key. Let $W \in \mathbb{S}^V$ represent the weights w:

$$W_{ij} := \begin{cases} w_{ij}, & \text{if } \{i,j\} \in E; \\ 0, & \text{otherwise.} \end{cases}$$

Then the following is an SDP relaxation of Max Cut:

$$\begin{array}{rrcl} \max & -\frac{1}{4}\langle W, X\rangle & & \left(+\frac{1}{4}\langle W, \bar{e}\bar{e}^T\rangle\right) \\ & \operatorname{diag}(X) & = & \bar{e}, \\ & X & \succeq & 0, \end{array}$$

where we wrote the constant term in the objective function in parentheses and $\operatorname{diag}(X)$ denotes the vector formed by the diagonal entries of X: $X_{11}, X_{22}, \ldots$. The dual of this SDP was studied before

$$\begin{array}{rrcl} \min & \bar{e}^T y & & \left(+\frac{1}{4}\langle W, \bar{e}\bar{e}^T\rangle\right) \\ & \operatorname{Diag}(y) - S & = & -\frac{1}{4}W, \\ & y \in \mathbb{R}^V,\ S & \succeq & 0. \end{array}$$

(In the above, $\operatorname{Diag}(y)$ denotes the diagonal matrix in $\mathbb{S}^V$ with diagonal entries: $y_1, y_2, \ldots, y_{|V|}$.) However, until Goemans-Williamson result utilizing the above SDP pair, the best approximation result for Max Cut was essentially a $\frac{1}{2}$-approximation algorithm. Goemans-Williamson proved that using an approximately optimal solution $\bar{X}$ of the above primal SDP, and a *random hyperplane technique*, their algorithm *rounds* positive semidefinite matrix $\bar{X}$ to a cut U in G with the property that the expected weight of the generated cut U is at least 0.87856 times the weight of the Max Cut in G.

Find $B \in \mathbb{R}^{|V| \times d}$ (here, $d \leq |V|$) such that $BB^T = \bar{X}$. Then focus on the columns of B^T:

$$B^T =: \left[b^{(1)}, b^{(2)}, \ldots, b^{(|V|)}\right].$$

Now, here is the randomization (Random Hyperplane Technique). We pick randomly, using the uniform distribution on the unit hypersphere in $\mathbb{R}^d$ ($\{h \in \mathbb{R}^d : \|h\|_2 = 1\}$) a vector u. Then, we define

$$U := \{i \in V : u^T b^{(i)} \geq 0\}.$$

Goemans and Williamson [23] proved that the expected total weight of this cut U is at least

0.87856.(weight of the max. cut).

After a de-randomization of this randomized algorithm, their results led to the following theorem.

Theorem 7.1. *Let $G = (V, E)$ with $w \in \mathbb{Q}_+^E$ (rational, nonnegative weights on the edges) be given. Then a cut in G of weight at least*

0.87856.*(weight of the max. cut)*

can be computed in polynomial-time.

8 Information and Communication Theory, Geometric Representations of Graphs, Lovász' ϑ-function

Suppose we have two groups communicating via a noisy channel using an alphabet. We further assume that noise can only cause confusion among certain pairs of letters and these are known by us. (E.g., letter a may be only confused with letters g, o and u; letter b may only be confused with letters d, k, l and o.) Notice that we can easily (and naturally) represent these confusion causing pairs by a simple undirected graph G with node set V and edge set E as follows: For every letter in the alphabet, we create a node in the graph G, and we connect nodes u and v by an edge if the letters corresponding to the nodes $u, v \in V$ can be confused with each other.

Now, let us consider the problem of finding the maximum number of letters that can be transmitted without confusion. This problem is equivalent to the problem of finding a maximum cardinality subset S of V in the graph G such that for every $u, v \in S$, $\{u, v\} \notin E$. The latter problem is the *maximum cardinality stable set problem* and has been studied extensively in combinatorial optimization and graph theory literature. While this is good news, the bad news is that it is $\mathcal{NP}$-hard. We denote by $\alpha(G)$ the cardinality of the largest stable set (also called the *stability number* of G) in G.

In the above setting of the communication problem, a very interesting parameter of the communication channel is its *Shannon Capacity*. This quantity can be defined using the graph G, a fundamental graph product operation and the concept of the stability number of a graph.

Given two graphs $G = (V, E)$ and $H = (W, F)$ we define their *strong product* as follows:

- create $|V| \times |W|$ nodes in $|V|$ rows and $|W|$ columns;
- for each column, create a copy of G;
- for each row, create a copy of H;
- focus on nodes (i, u) and (j, v) with $i \neq j$ and $u \neq v$; connect these two nodes if $\{i, j\} \in E$ and $\{u, v\} \in F$.

Now, we denote the strong product of G with itself by G^2. It should be clear from the definition of G^2 that $\alpha(G^2)$ is the maximum number of words of length two so that for every pair of words there exists at least one i (among these two positions) such that the ith letters are different and they cannot be confused with each other. Continuing this way, we define G^k and we deduce that $\alpha(G^k)$ determines the maximum number of k-letter words that can be used in this communication model without confusion. Then the *Shannon Capacity* of G is

$$\Theta(G) := \lim_{k \to \infty} \left[\alpha\left(G^k\right)\right]^{1/k}.$$

It is easy to observe from the above definitions that

$$\alpha\left(G^k\right) \geq [\alpha(G)]^k, \quad \forall k \in \mathbb{Z}_{++}.$$

This communication problem and the underlying measure, Shannon Capacity are very interesting; however, we expressed it as a function of the stability number of G^k. So, we still need to find ways of computing or at least approximation the stability number of graphs.

The most elementary LP relaxation is given by

$$\begin{array}{rrcl} \max & \bar{e}^T x & & \\ & x_i + x_j & \leq & 1, \quad \{i,j\} \in E, \\ & x & \in & [0,1]^V. \end{array}$$

The feasible region of the above LP relaxation is called the *Fractional Stable Set Polytope* which we denote by FRAC(G). We denote by STAB(G) the convex hull of incidence vectors of stable sets in G. So,

$$\text{STAB}(G) = \text{conv}\left[\text{FRAC}(G) \cap \{0,1\}^V\right].$$

A much tighter relaxation of STAB(G) is given by so-called *theta body* of G (see [24]). We denote it by TH(G). TH(G) is a convex set which can be expressed as the feasible region of an SDP problem. Namely, it is the set of $x \in \mathbb{R}^V$ satisfying for some $Y \in \mathbb{S}_+^{\{0\}\cup V}$ the following constraints:

- $Ye_0 = \text{Diag}(Y) = \begin{pmatrix} 1 \\ x \end{pmatrix}$,
- $Y_{ij} = 0, \quad \forall \{i,j\} \in E.$

In the above formulation, Y is $(|V|+1)$-by-$(|V|+1)$ and we indexed the columns and the rows of Y by $0, 1, 2, \ldots, |V|$. (The first index is special, the remaining indices correspond to the nodes in G.)

An impressive result about this SDP relaxation is that it is equivalent to the convex relaxation of STAB(G) defined by *orthonormal representation constraints* (see for instance [24]). Let us see what these constraints are: $\{u^{(i)} : i \in V\} \subset \mathbb{R}^V$ is called an *orthonormal representation of* G if

- $\|u^{(i)}\|_2 = 1, \quad \forall i \in V$ and,
- $\langle u^{(i)}, u^{(j)} \rangle = 0, \quad \forall \{i,j\} \in E.$

So, we represent the nodes of the graph by unit vectors so that *unrelated nodes with respect to* E are *unrelated geometrically* (i.e., orthogonal to each other). Hence, we converted the algebraic representation (by nodes and edges) of the graphs to a geometric one. Note however that in general, it is not possible to construct G from any orthonormal representation. (Consider the set $\{e_1, e_2, \ldots, e_n\}$ which gives an orthonormal representation of every graph G with n nodes.)

Now, for every $c \in \mathbb{R}^V$ with $\|c\|_2 = 1$, the linear constraint

$$\sum_{j \in V} \left(c^T u^{(j)}\right)^2 x_j \leq 1$$

is called an *orthonormal representation constraint for G* and it is a valid inequality for STAB(G). In particular, for every clique $\mathcal{C}$ in G, the *clique inequality*

$$\sum_{j \in \mathcal{C}} x_j \leq 1$$

is an orthonormal representation constraint. Based on these clique inequalities, we can define the *clique polytope of G*:

$$\mathrm{CLQ}(G) := \left\{ x \in [0,1]^V : \sum_{j \in \mathcal{C}} x_j \leq 1, \quad \forall \text{ cliques } \mathcal{C} \text{ in } G \right\}.$$

We have

$$\mathrm{STAB}(G) \subseteq \mathrm{TH}(G) \subseteq \mathrm{CLQ}(G) \subseteq \mathrm{FRAC}(G).$$

Lovász' θ-function is

$$\theta(G) := \max \left\{ \bar{e}^T x : x \in \mathrm{TH}(G) \right\}.$$

We can compute $\theta(G)$ by solving an SDP! Moreover, it is not difficult to prove

$$\theta\left(G^k\right) = \left[\theta(G)\right]^k, \quad \forall k \in \mathbb{Z}_{++}.$$

Therefore, we can *sandwich* the Shannon Capacity of G between the Lovász theta number and the stability number of G:

$$\alpha(G) \leq \Theta(G) \leq \theta(G), \quad \forall \text{ graphs } G.$$

For a more detailed exposition to this and related results, see Knuth [27] and Lovász [36, 37].

9 Network Design, Branch-and-Bound, Sparsest Cut

Many network design problems and multi-commodity flow problems have connections to the *sparsest cut problem* which is to find a subset $U \subset V$ attaining the following minimum:

$$\min_{U \subset V : |U| \leq \frac{|V|}{2}} \left\{ \frac{|\delta(U)|}{|U|} \right\}.$$

This minimum can be approximated by the optimum objective value of the following SDP problem (up to a factor of $|V|$):

$$\begin{array}{ll} \min & \sum_{\{i,j\} \in E} (X_{ii} + X_{jj} - 2X_{ij}) \\ & \begin{array}{rcl} X_{ij} + X_{jk} - X_{ik} - X_{jj} & \leq & 0, \quad \forall i,j,k; \\ \sum_{i<j} (X_{ii} + X_{jj} - 2X_{ij}) & = & 1, \\ X & \in & \mathbb{S}^V_+. \end{array} \end{array}$$

See, Arora, Rao and Vazirani [3].

A very closely related problem is: given a constant $c \in (0,1)$, find a cut $U \subset V$ which is an optimal solution of

$$\min_{U \subset V : |U| \geq c|V| \text{ and } |V \setminus U| \geq c|V|} \left\{ \frac{|\delta(U)|}{|U|} \right\}.$$

This problem is called the *c-balanced graph separation problem* which also admits a useful SDP relaxation similar to the one we mentioned above for the sparsest cut problem:

$$\begin{array}{lrcl} \min & \frac{1}{2}\sum_{\{i,j\}\in E}(1 - X_{ij}) & & \\ & \operatorname{diag}(X) & - & \bar{e}, \\ & X_{ij} + x_{jk} - X_{ik} & \leq & 1, \quad \forall i,j,k; \\ & \sum_{i<j} X_{ij} & = & \frac{|V|(|V|-1)}{2} - 2c(1-c)|V|^2, \\ & X & \in & \mathbb{S}_+^V. \end{array}$$

The c-balanced graph separator problem tries to separate the given graph into two pieces such that each piece has a significant portion (that is, at least $c|V|$) of the nodes and the interaction between the pieces (measured by the ratio $\frac{|\delta(U)|}{|U|}$) is minimized. Such separations are very useful in branch-and-bound like schemes in solving various network design, graph optimization problems.

There are also many applications of SDP in the related area of graph realization which in turn has applications in biology (e.g., molecular confirmation), structural engineering design and wireless sensor network localization. See, for instance So and Ye [67, 68], Al-Homidan and Wolkowicz [1].

10 Portfolio Optimization, Financial Mathematics

At the time of this writing, financial mathematics in general and portfolio optimization in particular make up some of the most popular application areas of optimization. Moreover, SOCP and SDP play very important roles.

The basic objective functions optimize a suitable linear combination of expected return and estimated risk of the portfolio. Under suitable assumptions, we can express upper bound constraints on variance as well as short risk constraints as second order cone constraints. Also, using SDP, we can work with constraints on the higher moments (not just the expectation and the variance) but this usually requires larger size SDPs.

For many applications and various different approaches, see Bertsimas and Sethuraman [8], Cornuéjols and Tütüncü [17], Lobo [34], Lobo, Fazel and Boyd [35], Li and Tunçel [33] and the references therein.

11 Robust Optimization

In most applications of optimization, there are significant parts of the data that are subject to uncertainty. Moreover, in many applications, the optimization model itself is an approximation to a real phonemenon. Even in the simplest applications, operations researchers rely on forecasting techniques and/or cost analysis methods to estimate unit costs, unit profits, demands, etc.

Since the birth of OR and LP theory, various approaches have been developed. Most notable ones are:

- sensitivity analysis,
- chance-constrained optimization,
- stochastic optimization,
- and more recently *Robust Optimization.*

Robust optimization requires an uncertainty set $\mathcal{U}$ for the data space and strives for the best solution with respect to the worst scenerio in the uncertainty set. For, example, let us consider a given LP problem:

$$\begin{array}{rrcl} \min & c^T x & & \\ & Ax & \geq & b. \end{array}$$

We can use a new variable x_{n+1} and rewrite the LP as

$$\begin{array}{rrcl} \min & x_{n+1} & & \\ & -c^T x + x_{n+1} & \geq & 0, \\ & Ax & \geq & b. \end{array}$$

So, without loss of generality, we may assume that the objective function of the given LP is certain and known exactly. Thus, our uncertainty set may be assumed to be a subset of the space where the pairs (A, b) live.

A *robust solution* in this context means $\tilde{x}$ such that

$$A\tilde{x} \geq b, \quad \forall (A, b) \in \mathcal{U}.$$

I.e., $\tilde{x}$ is feasible for every possible realization of data in the uncertainty set $\mathcal{U}$. Among all of these robust solutions, we want the one that minimizes our objective function.

There are many reasonable choices for $\mathcal{U}$ which allow the resulting robust optimization problem to be formulated as an SOCP or SDP problem (although many other choices for $\mathcal{U}$ lead to $\mathcal{NP}$-hard problems). See for instance: Ben-Tal and Nemirovskii [7], El Ghaoui, Oustry and Lebret [19], Bertsimas and Sim [10, 11, 9], Mulvey, Vanderbei and Zenios [41], Hanafizadeh and Seifi [25], Moazeni [39], Moazeni and Tunçel [40] and the references therein.

There are many successful applications of Robust Optimization, a classical one being in the area of Truss Topology Design (see [7]).

12 Universal Approaches with Lift-and-Project Methods

We will start with 0,1 mixed integer programming. Suppose we are given the data for the following form of 0,1 mixed integer programming:

$$\begin{array}{rrcl} \min & c^T x + d^T z & & \\ & Ax + Bz & \leq & b, \\ & x & \in & \mathbb{R}^n_+, \\ & z & \in & \{0,1\}^d. \end{array}$$

Note that

$$z \in \{0,1\}^d \iff z_j^2 - z_j = 0, \ \forall j \in \{1, 2, \ldots, d\}.$$

Further notice that the constraints

$$z_j^2 - z_j \leq 0, \ \forall j \in \{1, 2, \ldots, d\}$$

define a convex set. So, in this quadratic formulation of the original problem, the *difficult* constraints are

$$z_j^2 - z_j \geq 0, \ \forall j \in \{1, 2, \ldots, d\}.$$

Now, consider the $(d+1)$-by-$(d+1)$ symmetric matrix

$$Y := \begin{bmatrix} 1 & z^T \\ z & zz^T \end{bmatrix}.$$

Let us refer to the first row and first column of Y by the index 0. Then the equations

$$\mathrm{diag}(Y) = Ye_0$$

precisely state that $z_j^2 = z_j, \ \forall j \in \{1, 2, \ldots, d\}$. Moreover, these equations are *linear* in the matrix variable Y. Such Y (as defined above) is always positive semidefinite; however, for Y to have the above structure, we also need it to be of rank at most one. The latter rank constraint is indeed *hard* (for instance it makes the feasible region potentially *nonconvex* an even *disconnected*). So, we will relax this rank constraint. Let us consider a positive semidefinite $Y \in \mathbb{S}^{d+1}$ such that $Y_{00} = 1$. Then Y looks like

$$Y := \begin{bmatrix} 1 & z^T \\ z & Z \end{bmatrix}.$$

By Schur Complement Lemma,

$$Y \succeq 0 \iff Z \succeq zz^T.$$

We ideally would like to have $Z = zz^T$ which can be equivalently written as

$$Z \succeq zz^T \text{ and } Z \preceq zz^T.$$

Our SDP relaxation throws away the nonconvex constraint $Z \preceq zz^T$. So, we have the SDP relaxation

$$\begin{array}{rrcl} \min & c^T x + d^T z & & \\ & Ax + Bz & \leq & b, \\ & \operatorname{diag}(Z) & = & z, \\ & \begin{bmatrix} 1 & z^T \\ z & Z \end{bmatrix} & \succeq & 0, \\ & x & \in & \mathbb{R}^n_+. \end{array}$$

Note that the condition $Z \in \mathbb{S}^d_+$ is implied by the existing constraints. Moreover, $Z \in \mathbb{S}^d_+$ implies $\operatorname{diag}(Z) \geq 0$; hence, $z \in \mathbb{R}^d_+$. Finally, Schur complement lemma implies $z_i^2 \leq z_i$ for every i, yielding (together with $z \geq 0$) $z \in [0,1]^d$. Using the work of Balas, Ceria and Cornuéjols [5], Lovász and Schrijver [38], and Sherali and Adams [60], we can keep strengthening the above formulation (at a cost of large increases in the number of variables and constraints).

Let

$$\mathcal{F}_0 := \left\{ \begin{bmatrix} x \\ z \end{bmatrix} \in \mathbb{R}^{n+d} : \begin{bmatrix} x \\ z \\ Z \end{bmatrix} \text{ is feasible in the SDP relaxation for some } Z \right\}.$$

Consider for $k \geq 1$,

$$\mathcal{F}_k := \left\{ \begin{bmatrix} x \\ z \end{bmatrix} \in \mathbb{R}^{n+d} : \begin{array}{l} Y := \begin{bmatrix} 1 & z^T \\ z & Z \end{bmatrix} \succeq 0, \operatorname{diag}(Z) = z, \\ \frac{1}{z_i} Y e_i, \frac{1}{1-z_i} Y(e_0 - e_i) \text{ are feasible in } \mathcal{F}_{k-1} \text{ for some choice of } x \in \mathbb{R}^n \end{array} \right\}.$$

In the above definition of $\mathcal{F}_k$, if z_i is 0 or 1 then one of the conditions have a division by zero and we simply ignore that condition. This is justified in our context.

Using the theory of lift-and-project methods, it is not difficult to prove that if $\mathcal{F}_0$ is bounded then we need to generate at most d of these sets $\mathcal{F}_k$ to reach a convex formulation of the convex hull of the original 0,1 mixed programming problem.

Lift-and-project methods have been generalized to handle problems beyond 0,1 mixed integer programming problems. See, Sherali and Adams [59], Sherali and Alameddine [61], Sherali and Tuncbilek [63], Kojima and Tunçel [29, 30, 28], Lasserre [31], Parrilo [50] for methods applicable to constraints given as inequalities on polynomial functions of x and z as well as the more general setting of infinitely many quadratic inequalities.

References

[1] Al-Homidan, S. and Wolkowicz, H. (2005), Approximate and exact completion problems for Euclidean distance matrices using semidefinite programming. *Linear Algebra Appl.*; 406, 109–141.

[2] Alizadeh, F. (1995), Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*; 5, 13–51.

[3] Arora, S., Rao, S. and Vazirani, U. (2004), Expander flows, geometric embeddings and graph partitioning. *Proc. 36th Annual ACM Symposium on Theory of Computing*; 222–231.

[4] Atashgah, A. B. and Seifi, A. (2007). Application of semi-definite programming to design of multi-response experiments. *IIE Trans.*; 39, 763–769.

[5] Balas, E., Ceria, S. and Cornuéjols, G. (1993), A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Math. Program.*; 58, 295–323.

[6] Barahona, F. (1990), On some applications of the Chinese Postman Problem. In: Korte, B., Lovász, L., Schrijver, A., and Prömel, H. J. (Eds), *Paths, Flows and VLSI Layout*, Springer-Verlag; 1–16.

[7] Ben-Tal, A. and Nemirovski, A. (2001), *Lectures on Modern Convex Optimization.* MPS-SIAM Series on Optimization: SIAM, Philadelphia.

[8] Bertsimas D. and Sethuraman, J. (2000), Moment problems and semidefinite optimization. In: Wolkowicz, H., Saigal, R. and Vandenberghe, L. (Eds), *Handbook of Semidefinite Programming Theory, Algorithms, and Applications*, Kluwer Academic Publishers: MA, USA; 469–509.

[9] Bertsimas, D. and Sim, M. (2006), Tractable approximations to robust conic optimization problems. *Math. Program.*; 107, 5–36.

[10] Bertsimas, D. and Sim, M. (2004), The price Of robustness. *Operations Research*; 52, 35–53.

[11] Bertsimas, D. and Sim, M. (2003), Robust discrete optimization and network flows. *Math. Program.*; 98, 49–71.

[12] Borwein, J. M. and Wolkowicz, H. (1981), Regularizing the abstract convex program. *J. Math. Anal. Appl.*; 83, 495–530.

[13] Boyd, S., Ghaoui, L. E., Feron, E. and Balakrishnan, V. (1994), *Linear Matrix Inequalities in System and Control Theory.* SIAM, Philadelphia.

[14] Boyd, S. and Vandenberghe, L. (2004), *Convex Optimization.* Cambridge University Press: Cambridge, UK.

[15] Bruno, J. L., Coffman Jr., E. G. and Sethi, R. (1974), Scheduling independent tasks to reduce mean finishing time. *Comm. for the ACM*; 17, 382–387.

[16] Chen, R. W., Kajitani, Y. and Chan, S. P. (1983), A graph theoretic via minimization algorithm for two-layer printed circuit boards. *IEEE Trans. Circuits Syst.*; 30, 284–299.

[17] Cornuéjols, G. and Tütüncü, R. (2007), *Optimization Methods in Finance*. Cambridge University Press, Cambridge.

[18] Deza, M. and Laurent, M. (1997), *Geometry of Cuts and Metrics*, Springer-Verlag.

[19] El Ghaoui, L., Oustry, F. and Lebret, H. (1998), Robust solutions to uncertain semidefinite programs. *SIAM J. Optim.*; 9, 33–52.

[20] Faraut, J. and Korányi, A. (1994), *Analysis on Symmetric Cones*. Oxford University Press: NY, USA.

[21] Goemans, M. X. (1998), Semidefinite programming in combinatorial optimization. *Proceedings of the International Congress of Mathematicians*. Vol. III. Doc. Math.; Extra Vol. III, 657–666.

[22] Goemans, M. X. (1997), Semidefinite programming in combinatorial optimization. *Math. Program.*; 79, 143–161.

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

[24] Grötschel, M., Lovász, L. and Schrijver, A. (1988) *Geometric Algorithms and Combinatorial Optimization*. Springer, New York.

[25] Hanafizadeh, P. and Seifi, A. (2004), A unified model for robust optimization of linear programs with uncertain parameters. *Transactions on Operational Research*; 16, 25–45.

[26] Kawaguchi, T. and Kyan, S. (1986), Worst case bound of an LRF schedule for the mean weighted flow-time problem. *SIAM J. Comput.*; 15, 1119–1129.

[27] Knuth, D. E. (1994), The sandwich theorem, *Electron. J. Combin.*; 1, approx. 48 pp. (electronic).

[28] Kojima, M. and Tunçel, L. (2002), Some fundamental properties of successive convex relaxation methods on LCP and related problems. *J. Global Optim.*; 24, 333–348.

[29] Kojima, M. and Tunçel, L. (2000), Cones of matrices and successive convex relaxations of nonconvex sets. *SIAM J. Optim.*; 10, 750–778.

[30] Kojima, M. and Tunçel, L. (2000), Discretization and localization in successive convex relaxation methods for nonconvex quadratic optimization problems. *Math. Program.*; 89, 79–111.

[31] Lasserre, J. B. (2002), Polynomials nonnegative on a grid and discrete optimization. *Trans. Amer. Math. Soc.*; 354, 631–649.

[32] Lenstra, J. K., Rinooy Kan, A. H. G. and Brucker, P. (1977), Complexity of machine scheduling problems. *Annals of Discrete Mathematics*; 1, 343–362.

[33] Li, A. and Tunçel, L. (2006), Some applications of symmetric cone programming in financial mathematics. *Transactions on Operational Research*; 17, 1–19.

[34] Lobo, M. S. (2000), *Convex and Robust Optimization with Applications in Finance.* Ph.D. thesis: Stanford University, USA.

[35] Lobo, M. S., Fazel, M. and Boyd, S. (2007), Portfolio optimization with linear and fixed transaction costs. *Ann. Oper. Res.*; 157, 341–365.

[36] Lovász, L. (2003), Semidefinite programs and combinatorial optimization. In: Recent advances in algorithms and combinatorics. CMS Books Math./Ouvrages Math. SMC, 11: Springer, New York.

[37] Lovász, L. (1979), On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*; 25, 1–7.

[38] Lovász, L. and Schrijver, A. (1991), Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*; 1, 166–190.

[39] Moazeni, S. (2006), *Flexible Robustness in Linear Optimization.* M.Math. Thesis: Dept. of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Canada.

[40] Moazeni, S. and Tunçel, L. (2006), A quick-and-dirty approach to flexible robustness in linear and semidefinite optimization. *in preparation.*

[41] Mulvey, J. M., Vanderbei, R. J. and Zenios, S. A. (1995), Robust optimization of large-scale systems. *Operations Research*; 43, 264–281.

[42] Nemirovski, A. (2004), Proximal method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Optim.*; 15, 229–251.

[43] Nemirovski, A. and Yudin, D. B. (1983), *Problem Complexity and Method Efficiency in Optimization.* Wiley-Interscience Series in Discrete Mathematics, John Wiley & Sons, Inc.: New York.

[44] Nesterov, Yu. (to appear), Dual extrapolation and its applications for solving variational inequalities and related problems. *Math. Prog.*

[45] Nesterov, Yu (2004), Rounding of convex sets and efficient gradient methods for linear programming problems. *Core Discussion Paper* 2004/4: Louvain-la-Neuve, Belgium.

[46] Nesterov, Yu. (2003), Unconstrained convex minimization in relative scale. *Core Discussion Paper* 2003/96: Louvain-la-Neuve, Belgium.

[47] Nesterov, Yu. (2005), Smooth minimization of nonsmooth functions. *Math. Prog.*; 103, 127–152.

[48] Nesterov, Yu. E. and Nemirovskii, A. S. (1994), *Interior-Point Polynomial Algorithms in Convex Programming.* SIAM: Philadelphia, PA, USA.

[49] Nesterov, Yu. E. and Todd, M. J. (1997), Self-scaled barriers and interior-point methods for convex programming. *Math. Oper. Res.*; 22, 1–46.

[50] Parrilo, P. A. (2003), Semidefinite programming relaxations for semialgebraic problems. *Math. Program.*; 96, 293–320.

[51] Pataki, G. (2007), On the closedness of the linear image of a closed convex cone. *Math. Oper. Res.*; 32, 395–412.

[52] Peng, J. and Wei, Y. (2007), Approximating k-means-type clustering via semidefinite programming. *SIAM J. Optim.*; 18, 186–205.

[53] Pinedo, M. (2002), *Scheduling Theory, Algorithms, and Systems.* Second Edition, Prentice-Hall.

[54] Pinter, R. Y. (1984), Optimal layer assignment for interconnect. *J. VLSI Comput. Syst.*; 1, 123–137.

[55] Polik, I. and Terlaky, T. (2007), Exact duality for optimization over symmetric cones. *Tech. Report 2007/10*: McMaster University, Hamilton, Ontario, Canada.

[56] Ramana, M. V. (1997), An exact duality theory for semidefinite programming and its complexity implications. *Math. Program.*; 77, 129–162.

[57] Ramana, M. V., Tunçel, L. and Wolkowicz, H. (1997), Strong duality for semidefinite programming. *SIAM J. Optim.*; 7, 641–662.

[58] Rockafellar, R. T. (1967), *Convex Analysis.* Princeton University Press: NJ, USA.

[59] Sherali, H. D. and Adams, W. P. (1999), *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems.* Nonconvex Optimization and its Applications, 31. Kluwer Academic Publishers: Dordrecht.

[60] Sherali, H. D. and Adams, W. P. (1990), A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Disc. Math.*; 3, 411–430.

[61] Sherali, H. D. and Alameddine, A. (1992), A new reformulation-linearization technique for bilinear programming problems. *J. Global Optim.*; 2, 379–410.

[62] Sherali, H. D. and Lee, Y. (1996), Tighter representations for set partitioning problems. *Discrete Appl. Math.*; 68, 153–167.

[63] Sherali, H. D. and Tuncbilek, C. H. (1995), A reformulation-convexification approach for solving nonconvex quadratic programming problems. *J. Global Optim.*; 7, 1–31.

[64] Shioda R. and Tunçel, L. (2007), Clustering via minimum volume ellipsoids. *Comput. Optim. Appl.*; 37, 247–295.

[65] Skutella, M. (1998), Semidefinite relaxations for parallel machine scheduling. *Proc. FOCS.*

[66] Smith, W. E. (1956), Various optimizers for single-stage production. *Naval Research and Logistics Quarterly*; 3, 59–66.

[67] So, A. M.-C. and Ye, Y. (2007), Theory of semidefinite programming for sensor network localization. *Math. Program.*; 109, 367–384.

[68] So, A. M.-C. and Ye, Y. (2005), A semidefinite programming approach to tensegrity theory and realizability of graphs. *manuscript*.

[69] Todd, M. J. (2001), Semidefinite optimization. *Acta Numer.*; 10, 515–560.

[70] Tunçel, L. (to appear), *Polyhedral and Semidefinite Programming Methods in Combinatorial Optimization*.

[71] Tunçel, L. (2001), On the Slater condition for the SDP relaxations of nonconvex sets. *Oper. Res. Lett.*; 29, 181–186.

[72] Tunçel, L. and Wolkowicz, H. (2008), Strong duality and minimal representations for cone optimization. *Research Report*; Dept. of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada.

[73] Vandenberghe, L, Boyd, S. and Wu, S.-P. (1998), Determinant maximization with linear matrix inequality constraints. *SIAM J. Matrix Anal. Appl.*; 19, 499–533.

[74] Wolkowicz, H. (1981), Some applications of optimization in matrix theory. *Linear Algebra Appl.*; 40, 101–118.

[75] Wolkowicz, H., Saigal, R. and Vandenberghe, L. Eds. (2000), *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*. Kluwer Academic Publishers: Boston, MA.

[76] Ye, Y. (1997), *Interior-Point Algorithms: Theory and Analysis*. John Wiley and Sons: NY.

[77] Zvereva, M. (2007), *Mathematical Programming Formulations of the Planar Facility Location Problem*. M.Math. Thesis: Dept. of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo. Canada.