§ 3 SDP relaxations of quadratic programs

<u>Def 1</u> A quadratic program (QP) is an optimization problem of the following form

$$qp = \sup \; x^T Q_0 x + b_0^T x + \alpha_0$$
$$x \in \mathbb{R}^n$$
$$x^T Q_j x + b_j^T x + \alpha_j = 0 \qquad j \in [m],$$

where $Q_0, Q_1, \ldots, Q_m \in S^n$, $b_0, \ldots, b_m \in \mathbb{R}^n$, $\alpha_0, \ldots, \alpha_m \in \mathbb{R}$ are given

One can model many optimization problems as quadratic programs, for example problems with $0/1$-variables

$$x_j^2 - x_j = 0$$

or with $-1/+1$-variables

$$x_j^2 - 1 = 0.$$

QPs are generally not convex programs and it can be NP-hard to find an optimal solution ($\rightarrow$ Chapter V).

<u>Goal</u>: Compute upper bounds for qp efficiently.

Theorem 2

$$qp \leq \inf y_0$$
$$y_0, \dots, y_m \in \mathbb{R}$$
$$y_0 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \sum_{j=1}^{m} y_j \begin{bmatrix} \alpha_j & \frac{1}{2} b_j^T \\ \frac{1}{2} b_j & Q_j \end{bmatrix} - \begin{bmatrix} \alpha_0 & \frac{1}{2} b_0^T \\ \frac{1}{2} b_0^T & Q_0 \end{bmatrix} \in S^{n+1}_{\geq 0}.$$

<u>Proof</u> We have

$$x^T Q_j x + b_j^T x + \alpha_j = \left\langle \begin{bmatrix} \alpha_j & \frac{1}{2} b_j^T \\ \frac{1}{2} b_j & Q_j \end{bmatrix}, \begin{bmatrix} 1 & x^T \\ x & xx^T \end{bmatrix} \right\rangle,$$

where $\begin{bmatrix} 1 & x^T \\ x & xx^T \end{bmatrix} = \begin{bmatrix} 1 \\ x \end{bmatrix} \begin{bmatrix} 1 & x^T \end{bmatrix} \in S^{m+1}_{\geq 0}$.

So,

$$qp \leq \sup \left\langle \begin{bmatrix} \alpha_0 & \frac{1}{2} b_0^T \\ \frac{1}{2} b_0 & Q_0 \end{bmatrix}, X \right\rangle$$

$$X \in S^{m+1}_{\geq 0}, \quad X_{11} = 1$$

$$\left\langle \begin{bmatrix} \alpha_j & \frac{1}{2} b_j^T \\ \frac{1}{2} b_j & Q_j \end{bmatrix}, X \right\rangle = 0, \quad j \in [m].$$

Dualizing this semidefinite program and applying weak duality yields the statement of the theorem. ⊠

<u>Remarks</u>:
- If we impose "rank $X = 1$", then $qp = \sup$.
  One <u>relaxes</u> the rank condition in Theorem 2.
- Advantages of SDP:
  - efficiently computable (ellipsoid method, IPM).
  - feasible solutions provide an easy way to upper bound qp.

Chapter V Approximation algorithms

§1 MAXCUT

Def. 1 Let $G = (V, E)$ be an undirected graph with weight function $w = (w_{ij}) \in \mathbb{R}^E_{\geq 0}$.

A subset $S \subseteq V$ defines a cut $\delta(S) \subseteq E$ by

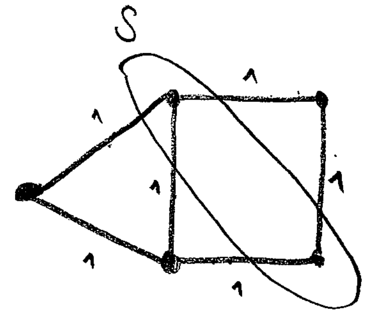
$\delta(S) = \{ \{i,j\} \in E : |\{i,j\} \cap S| = 1 \}$.

The weight of cut $\delta(S)$ is

$$w(\delta(S)) = \sum_{\{i,j\} \in \delta(S)} w_{ij}.$$

The MAXCUT problem asks for a cut of maximal weight:

$$mc(G, w) = \max_{S \subseteq V} w(\delta(S))$$



$mc(G, w) = 5$.

Computational complexity of MAXCUT

Def. 2 (MAXCUT; decision version)

input: $G = (V, E)$ graph, $w \in \mathbb{Q}^E_{\geq 0}$ weight, $k \in \mathbb{N}$

output: YES if $\exists\, S \subseteq V : \delta(S) \geq k$; NO otherwise.

Def. 3 (PARTITION)

input: $a_1, \dots, a_n \in \mathbb{N}$

output: YES if $\exists S \subseteq [n] : \sum_{i \in S} a_i = \sum_{i \notin S} a_i$;

NO otherwise.

It is well-known that PARTITION is an NP-complete problem (→ see classical book "Computers and Intractability" by Garey and Johnson, 1979)

Theorem 4 MAX CUT is NP-complete.

Proof: 1.) MAXCUT $\in$ NP

Clear: One can verify in polynomial time, given $S \subseteq [n]$, that $\delta(S) \geq k$.

2.) PARTITION $\leq_p$ MAXCUT. (i.e. if MAXCUT $\in$ P, then PARTITION $\in$ P).

Suppose $a_1, \dots, a_n$ are given. Define an input for MAXCUT as follows:

$G = K_n$ (complete graph on n vertices), $w_{ij} = a_i a_j$, $k = \frac{\sigma^2}{4}$,

with $\sigma = \sum_{i=1}^{n} a_i$.

For $S \subseteq [n]$ define $a(S) = \sum_{i \in S} a_i$.

We have

$$w(\delta(S)) = \sum_{\substack{i\in S\\ j\notin S}} w_{ij} = \sum_{\substack{i\in S\\ j\notin S}} a_i a_j = \left(\sum_{i\in S} a_i\right)\left(\sum_{j\notin S} a_j\right)$$

$$= a(S)\,(\sigma - a(S)) \leq \sigma^2/4 = k$$

and equality holds iff $a(S) = \frac{\sigma}{2}$, i.e. $a(S) = a([n]\setminus S)$.
So there exists a cut with weight $\geq k$ iff one can partition $a_1, \dots, a_n$. $\boxtimes$

MAX CUT is APX-hard: If $P \neq NP$, then one cannot approximate MAXCUT with a factor better than $\frac{16}{17} \approx 0{,}941\ldots$
(Håstad, 2001)

Formulation of MAXCUT as a QP

$$mc(G,w) = \max \ \frac{1}{2} \sum_{\{i,j\}\in E} w_{ij}\,(1 - x_i x_j)$$
$$x \in \mathbb{R}^V$$
$$x_i^2 - 1 = 0 \quad \text{for } i \in V.$$

Now: consider SDP relaxation and analyze its quality.