Of course, we can perform an obvious linear program relaxation replacing the binary constraint by

$$\begin{array}{ll} 1 - x_i \geq 0 & i = 1, 2, \ldots, n \\ x_i \geq 0 & i = 1, 2, \ldots, n. \end{array}$$

The Reformulation Linearization Technique (RLT) proposed by Sherali and Adams (around 1990's) is based on the following fact. Construct redundant quadractic constraints and perform a linearization. In the above case, we can construct three types of quadratic constraints:

$$\begin{array}{ll} x_i x_j \geq 0 & i, j = 1, 2, \ldots, n \\ (1 - x_i)(1 - x_j) \geq 0 & i, j = 1, 2, \ldots, n \\ (1 - x_i) x_j \geq 0 & i, j = 1, 2, \ldots, n \end{array}$$

Replacing all $x_i x_j$ by w_{ij}, we will have the following linear program relaxation.

$$\left\{ \begin{array}{lll} \text{minimize} & \displaystyle\sum_{i,j=1}^{n} Q_{ij} w_{ij} + \sum_{i=1}^{n} q_i x_i & \\ \text{subject to} & w_{ij} \geq 0 & 1 \leq i, j \leq n \\ & 1 - x_i - x_j + w_{ij} \geq 0 & 1 \leq i, j \leq n \\ & x_j - w_{ij} \geq 0 & 1 \leq i, j \leq n. \end{array} \right. \tag{3}$$

Of course the optimal value of (3) is smaller than or equal to the one of (2).

We can apply the RLT for a more general quadratic constrained quadratic program.

$$\left\{ \begin{array}{lll} \text{minimize} & \boldsymbol{x}^T \boldsymbol{Q}_0 \boldsymbol{x} + \boldsymbol{q}_0^T \boldsymbol{x} & \\ \text{subject to} & \boldsymbol{x}^T \boldsymbol{Q}_i \boldsymbol{x} + \boldsymbol{q}_i^T \boldsymbol{x} + \gamma_i \leq 0 & 1 \leq i \leq r \\ & \boldsymbol{a}_j^T \boldsymbol{x} + b_j \leq 0 & 1 \leq j \leq s. \end{array} \right. \tag{4}$$

Then we can take the products of the linear constraints and perform a similar linearization.

$$\begin{array}{rl} & 1 \leq \forall j \leq \forall k \leq s \\ & -(\boldsymbol{a}_j^T \boldsymbol{x} + b_j)(\boldsymbol{a}_k^T \boldsymbol{x} + b_k) \\ = & -\boldsymbol{x}^T \boldsymbol{a}_j \boldsymbol{a}_k^T \boldsymbol{x} - b_k \boldsymbol{a}_j^T \boldsymbol{x} - b_j \boldsymbol{a}_k^T \boldsymbol{x} - b_j b_k \leq 0. \end{array}$$

At the end, we can obtain the following relaxation for (4):

$$\left\{ \begin{array}{lll} \text{minimize} & \displaystyle\sum_{p,\ell=1}^{n} [\boldsymbol{Q}_0]_{p\ell} w_{p\ell} + \boldsymbol{q}_0^T \boldsymbol{x} & \\ \text{subject to} & \displaystyle\sum_{p,\ell=1} [\boldsymbol{Q}_i]_{p\ell} w_{p\ell} + \boldsymbol{q}_i^T \boldsymbol{x} + \gamma_i \leq 0 & 1 \leq i \leq r \\ & \displaystyle -\sum_{p,\ell=1}^{n} [\boldsymbol{a}_j \boldsymbol{a}_k^T]_{p\ell} w_{p\ell} - b_k \boldsymbol{a}_j^T \boldsymbol{x} - b_j \boldsymbol{a}_k^T \boldsymbol{x} - b_j b_k \leq 0 & 1 \leq j, k \leq s \\ & \boldsymbol{a}_j^T \boldsymbol{x} + b_j \leq 0 & 1 \leq j \leq s. \end{array} \right. \tag{5}$$

In some cases, (5) can be unbounded even when (4) has a bounded optimal value. In this case, it is necessary to add some redundant constraints for (4) in (5).

3.3 Exercises

1. Show that the constraints $0 \leq x_i \leq 1 \quad (i = 1, 2, \ldots, n)$ are redundant for (3) and therefore, unnecessary.
2. Give an example of a polyhedron defined by $\{\boldsymbol{y} \in \mathbb{R}^m \mid \boldsymbol{A}^T \boldsymbol{y} \leq \boldsymbol{c}\}$ where $\boldsymbol{A}$ and $\boldsymbol{c}$ have elements equal to $0, \pm 1$, but its vertices do not have integer values (coordinates).

4 Semidefinite Program Relaxation

4.1 Semidefinite Program (SDP)

If we take $\mathcal{K} = \mathbb{S}^n_+$ in (CLP), where $\mathbb{S}^n_+$ stands for the closed convex cone of real $n \times n$ symmetric positive semidefinite matrices, we have a Semidefinite Program (SDP).

$$\text{(SDP)} \begin{cases} \text{minimize} & \langle \boldsymbol{C}, \boldsymbol{X} \rangle \\ \text{subject to} & \langle \boldsymbol{A}_i, \boldsymbol{X} \rangle = \boldsymbol{b}_i \quad (i = 1, 2, \dots, m) \\ & \boldsymbol{X} \in \mathbb{S}^n_+, \end{cases}$$

and its associated dual problem:

$$\text{(DCLP)} \begin{cases} \text{maximize} & \langle \boldsymbol{b}, \boldsymbol{y} \rangle \\ \text{subject to} & \displaystyle\sum_{i=1}^{m} \boldsymbol{A}_i y_i + \boldsymbol{S} = \boldsymbol{C}, \\ & \boldsymbol{S} \in \mathbb{S}^n_+. \end{cases}$$

4.2 Maximum Cut Problem

Given a graph $G = (V, E)$ where $V := \{1, 2, \dots, n\}$ is the set of vertices and $E \subseteq V \times V$ is the set of edges, let us define *positive* weights for each existing edge $(u, v) \in E$ as $w : E \to \mathbb{R}_{++}$.

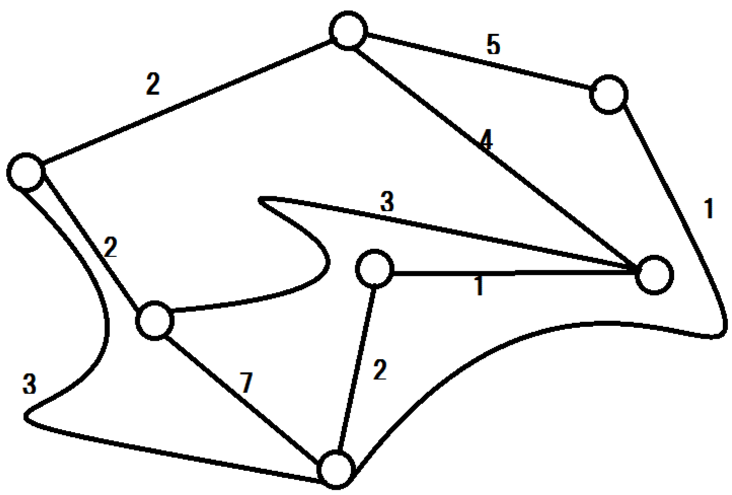

Figure 1: A undirected graph with positive weights.

Given a subset $S \subseteq V$, the subset of edges $\delta(S) := \{(u, v) \in E \mid u \in S,\ v \in V \backslash S\}$ is called a *cut* of the graph $G = (V, E)$. We can associate a *capacity* for the cut by summing all the edges' weights of the cut, $\delta_w(S) := \sum_{u \in S, v \in V \backslash S} w(u, v)$.

The problem to find a *minimum cut* (capacity) among all possible ones of a undirected graph with positive weights is an "easy" problem. This can be understood in the sense that we can use for instance the famous Ford and Fulkerson method to find the maximum flow of the graph which also gives the minimum cut of the graph from a specific source vertex to a sink vertex. The application of this method for all pair of vertices for instance will give the desired result. Of course, there are other complex algorithms which guarantee polynomial-time number of steps to determine the minimum cut of a undirected graph.