2
Unconstrained Binary Quadratic Programming

Definition 1 (Unconstrained Binary Quadratic Programming) *The Unconstrained Binary Quadratic Programming (UBQP) problem may be written as:*

$$
\begin{aligned}
z = \min \quad & \frac{1}{2}x^T Q x - b^T x \\
s.t.: \quad & x \in \{0,1\}^n
\end{aligned}
$$

Where Q is an $n \times n$ matrix and b is an vector with n elements.

We can assume without loss of generality that Q is symmetric, due to the following theorem:

Theorem 2 *For every square matrix Q, there exist a square symmetric matrix Q' such that $x^T Q x = x^T Q' x$ for every $x \in \mathbb{R}^n$.*

Proof. First observe that $x^T Q x = x^T Q^T x$, since:

$$
\begin{aligned}
x^T Q^T x &= < x^T Q^T, x > \\
&= < x^T, Qx > \\
&= x^T Q x
\end{aligned}
$$

If we let $Q' = \frac{Q+Q^T}{2}$ then:

$$
\begin{aligned}
x^T Q' x &= \frac{x^T Q x + x^T Q^T x}{2} \\
&= x^T Q x
\end{aligned}
$$

∎

2.1 Modeling power of UBQP

To show its modeling power, we will show how one can easily model two classic $\mathcal{NP}$-Hard problems.

2.1.1 Max Cut

The UBQP can model, for instance, the Max Cut problem, which is defined as follows:

Let $G = (V, E)$ be a graph with edge costs c_{ij}. The Max Cut problem consists of finding $V' \subseteq V$ such that $\sum_{(i,j)\in E | i \in V', j \notin V'} c_{ij}$ is maximum.

Theorem 3 (UBQP can model Max Cut) *It is possible to model the Max Cut problem using UBQP.*

Proof. Let $G = (V, E)$ be a graph with edge costs c_{ij}. Then build the following UBQP:

$$\begin{aligned} \min \quad & f(x) = -\sum_{(i,j)\in E} f_{ij}(x) \\ \text{s.t.:} \quad & x \in \{0,1\}^n \end{aligned}$$

Where:

- $n = |V|$, so that there is one variable x_i for each vertex i.
  A vector $x \in \{0,1\}^n$ represents the set $V = \{i | x_i = 0\}$.
- $f_{ij}(x) = c_{ij}(x_i + x_j - 2x_i x_j)$.
  It can be easily seen that $f_{ij}(x) = 0$ if $x_i = x_j$ and $f_{ij}(x) = c_{ij}$ if $x_i \neq x_j$.
  So $f_{ij}(x)$ represents the contribution of the edge (i, j) to the cut value.

From the observations above, it is easy to see that $f(x)$ is equal to minus the value of the cut represented by x, so minimizing $f(x)$ is indeed equivalent to maximizing the cut.

∎

2.1.2 Max Clique

The Max Clique problem can be defined as follows:

Let $G = (V, E)$ be a graph. A clique in this graph is a subset V' of V such that $i, j \in V' \Rightarrow (i, j) \in E$. The Max Clique problem consists of finding the clique V^* with the maximum number of vertices.

Theorem 4 (UBQP can model Max Clique) *It is possible to model the Max Clique problem using UBQP.*

Proof. Let $G = (V, E)$ be a graph. Then build the following UBQP:

$$\begin{aligned} \min \quad & f(x) = \sum_{(i,j)\notin E} 2x_i x_j - \sum_{i\in V} x_i \\ \text{s.t.:} \quad & x \in \{0,1\}^n \end{aligned}$$

Where $n = |V|$, so that there is one variable x_i for each vertex i. So a vector $x \in \{0,1\}^n$ represents the subset $\{i|x_i = 1\}$.

If x represents a clique it is easy to see that $f(x)$ will be minus the cardinality of that clique (there will be no $(i, j) \notin E$ such that $x_i x_j = 1$).

It rest showing that the optimum will always represent a clique. Well, suppose it doesn't. So there are $x_i = x_j = 1$ such that $(i, j) \notin E$. In this case, if we change the value of x_i to 0, the objective function will be decreased in at least one, so x is not optimal. ∎