Question about SDP relaxation of QKP

Qiyao Wei

March 2021

Suppose we have n variables. Denote our quadratic optimization term $Q \in R^{nxn}$, linear optimization term $q \in R^n$, linear capacity constraint $c \in R^n$, and the capacity term $C \in R$. Let us phrase the Quadratic Knapsack Problem (QKP) as follows

$$\begin{array}{l} \text{maximize } \left\{ \sum_{i=1}^{n} q_i x_i + \sum_{i=1}^{n} \sum_{j=1, i \neq j}^{n} Q_{ij} x_i x_j :, x \text{ binary} \right\} \\ \text{subject to } x \in \{0,1\}^n : \sum_{i=1}^{n} c_i x_i \leq C; x_i \in \{0,1\} \text{ for } i = 1, \ldots, n \end{array}$$

We can see that this is a standard quadratic programming question, with a quadratic optimization term and linear optimization term, as well as a linear constraint term. We know that for general quadratic programming questions, its SDP relaxation is as follows

Quadratically Constrained Quadratic Programming:
z$^* :=$ Maximize $\mathbf{x}^T Q \mathbf{x} + 2\mathbf{q}^T \mathbf{x}$
s.t. x$^T A_i \mathbf{x} + 2\mathbf{a}_i^T \mathbf{x} (\leq, =, \geq) b_i, \forall i = 1, \ldots, m$

that can be homogenized by adding an auxiliary variable:

$z^* :=$ Maximize $\mathbf{x}^T Q \mathbf{x} + 2x_{n+1} \mathbf{q}^T \mathbf{x}$

s.t. x$^T A_i \mathbf{x} + 2x_{n+1} \mathbf{a}_i^T \mathbf{x} (\leq, =, \geq) b_i, \forall i = 1, \ldots, m,$
x$_{n+1}^2 = 1$

$$\begin{array}{ll} \text{Maximize} & \begin{pmatrix} Q & \mathbf{q} \\ \mathbf{q}^T & 0 \end{pmatrix} \bullet X \\ s.t. & \begin{pmatrix} A_i & \mathbf{a}_i \\ \mathbf{a}_i^T & 0 \end{pmatrix} \cdot X = b_i, \forall i = 1, \ldots, m, \\ \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} \cdot X = 1 & \\ & X \succeq 0, \end{array}$$

where X $\in S^{n+1}$
Therefore, we do a couple of things to achieve this SDP formulation

a. We slightly abuse notation here by using the same Q and q for the QKP and SDP notations. Note that the Q matrices are the same, whereas the q in the SDP notation is half of that in the QKP notation, due to our multiplication factor by 2 in our general QCQP definition.

b. We create $n+1$ constraints, one for each n variables being binary (either 0 or 1), and the last one for the capacity constraint. The capacity constraint is simple, as we would simply have $A = 0$ and $a = c/2$, in our SDP formulation. Recall that c is the linear constraints defined in the QKP problem. For the binary variable constraint, we constrain the i_{th} variable to satisfy $x_i^2 - x_i = 0$. In order to do that, the i_{th} element on the diagonal of A must be 1, i.e. $A[i][i] = 1$, and the i_{th} element of a would be -0.5, i.e. $a[i] = \frac{-1}{2} = -0.5$. This is the exact same expression as the one in our general QCQP formulation.

c. All other constraints should be easy to understand. Aside from the previous two, we simply enforce that X is PSD, and the auxiliary variable squares to 1, i.e. it is either -1 or 1

Here is my question: If I relaxed the problem this way, why didn't my MOSEK solver give integer matrix solutions? Rather, it still solved for a continuous solution. In other words, all elements of X (other than the auxiliary variable) were floating point numbers. Why is that?