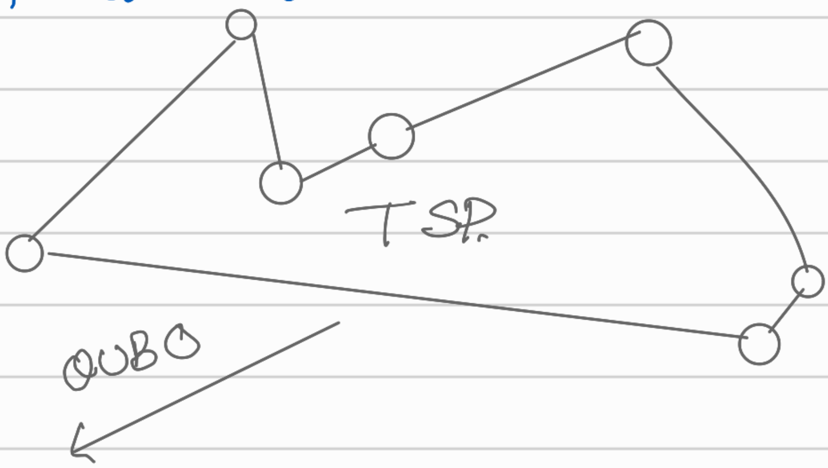
Improving Quantum and Classical Decomposition Methods for VRP.

Workflow

Starting from TSP instance of 7 nodes, and seeing that problem to a QUBO form, with dim = 36

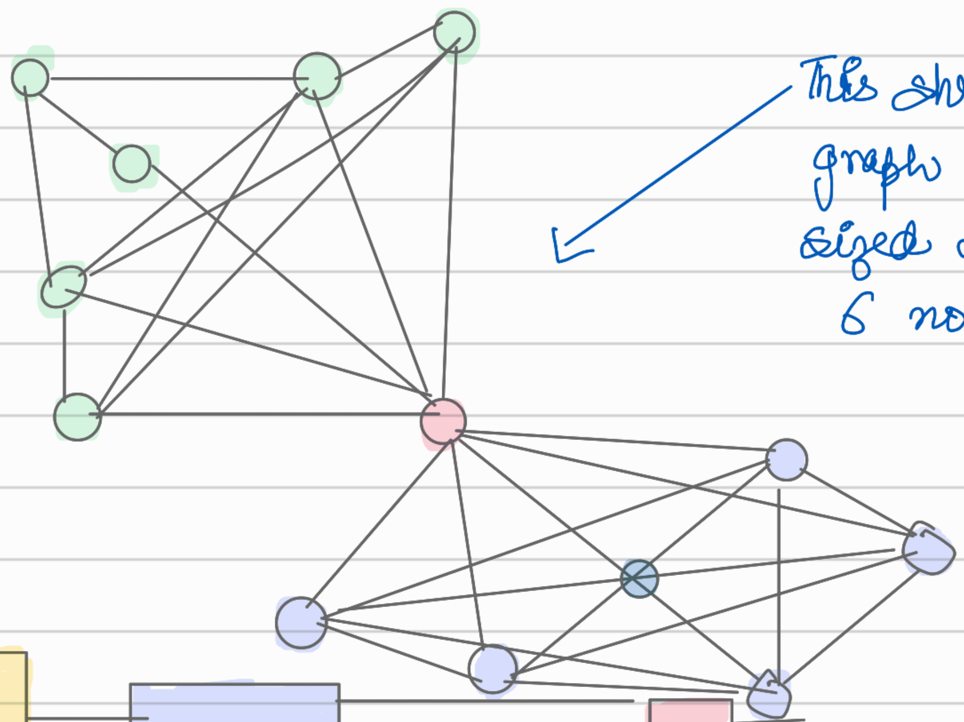


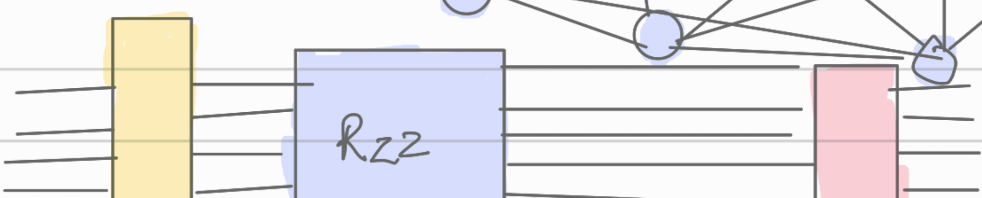
$$\min x^t Q x$$

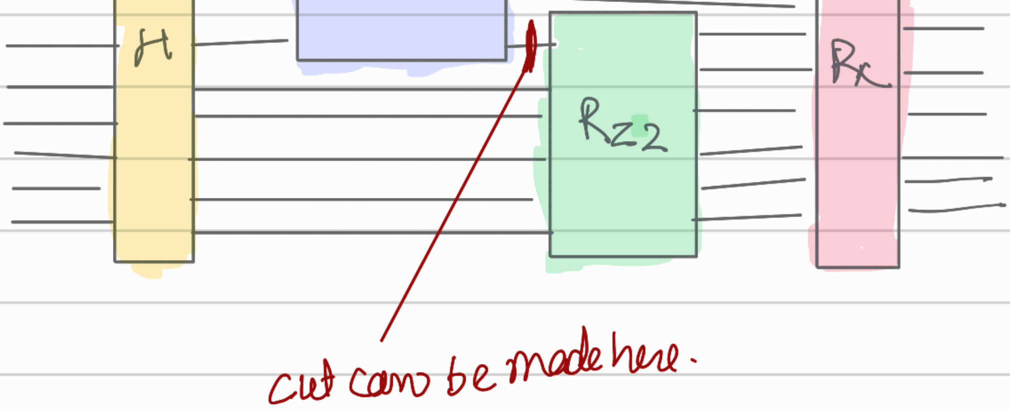
Any QUBO problem can be reformulated as a weighted Max cut instance by introducing one extra node.

Then heuristic graph partitioning algorithm is applied to search for a balanced vertex separator, which separates the graph into green and blue node subset which are connected only via the vertex separator (in red)

This shrinking reduce the graph to two equally sized subgraphs of 6 nodes each







QUBO and Max Cut

$$\min_{x \in \{0,1\}^n} \sum_{i,j} q_{ij} x_i x_j \qquad q_{ij} \in \mathbb{R}$$

For a given TSP instance on N vertices, the model requires $n = (N-1)^2$ variables

Any QUBO problem can be translated into an equivalent max cut problem with $(n+1)$ vertices.

Proof:

A Pseudo-Boolean quadratic function is

$$f(x) = x^T Q x + c^T x \qquad x \in \{0,1\}^n, \; Q \in \mathbb{R}^{n}, \; c \in \mathbb{R}^n$$

Since $x_i^2 = x_i$, we can assume that the diagonal of Q is zero.

Finding the minimum of this 'f' is an NP-Hard problem. it is polynomially solvable if all the elements of Q are non-positive.

The function 'f' can be written in the form

$$f(x) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} q_{ij} x_i x_j + \sum_{i=1}^{n} c_i x_i$$

Set $s_i = 2x_i - 1$

$$f(x(s)) = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{1}{4} q_{ij} s_i s_j + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{1}{4} q_{ij} s_i + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{1}{4} q_{ij} s_j + \sum_{i=1}^{n} \frac{1}{2} c_i s_i + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{1}{4} q_{ij} + \sum_{i=1}^{n} \frac{1}{2} c_i$$

$$\Rightarrow \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{1}{4} q_{ij} s_i s_j + \sum_{i=1}^{n} \left[\left(\frac{1}{4}\left(\sum_{j=1}^{i-1} q_{ji} + \sum_{j=i+1}^{n} q_{ij} \right) + \frac{1}{2} c_i \right] s_i + C_1$$

where $s_i \in \{-1,+1\}$ and $C_1 = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{1}{4} q_{ij} + \sum_{i=1}^{n} \frac{1}{2} c_i$, with an additional variable s_0, and

$$w_{0j} = \frac{1}{4}\left(\sum_{i=1}^{i-1} q_{ji} + \sum_{j=i+1}^{n} q_{ij} \right) + \frac{1}{2} c_i \qquad , 1 \le j \le n$$

$$w_{ij} = \frac{1}{4} q_{ij} \qquad 1 \le i < j \le n$$

we get $g(s) = \sum_{i=0}^{n-1}\sum_{j=i+1}^{n} w_{ij} s_i s_j + C_1$ where $s \in \{-1,+1\}^{n+1}$

Since $g(s) = g(-s)$ we assume that $s_0 = 1$, our problem now is to minimize g.

Define the graph $G = (V,E)$ with node set $V = \{0,1,2,\ldots,n\}$ and edge set $E = \{ij \mid 1 \le i < j \le n\}$ and take w_{ij} as the edge weight of $ij \in E$. Each assignment of values $+1$ and -1 to the variables s_i corresponds to a partition of vertex $V^+ = \{i \in V \mid s_i = +1\}$ and $V^- = \{i \in V \mid s_i = -1\}$. So ignoring C_1, we can write

$$\min \sum_{ij \in E(V^+)} w_{ij} + \sum_{ij \in E(V^-)} w_{ij} - \sum_{ij \in \delta(V^+)} w_{ij}$$

Equivalent Max cut problem.

or

$$\min - 2 \cdot \sum_{ij \in \delta(V^+)} w_{ij} + C_2.$$

Transforming QUBO into a Max Cut problem allows us to reduce the problem size by graph shrinking

Graph Shrinking

A hybrid heuristic for the weighted max cut, also well-suited for integration into exact branch-and-cut algorithms.

For large instances we reduce the problem size according to a linear relaxation such that the reduced problem can be handled by quantum machines of limited size.

Enhancing Quantum Algorithms for Quadratic Unconstrained Binary Optimization

A hybrid heuristic for weighted max-cut problem, that employs a linear programming relaxation, rendering it well-suited for integration into exact branch-and-cut algorithms. For large problem instances, we reduce the problem size according to a linear relaxation such that the reduced problem can be handled by quantum machines of limited size.

The algorithm builds on a well-known linear relaxation of Max Cut. Solving linear relaxations is the basis of branch-and-cut algorithms for IP, since they provide upper bounds on an optimum solution value.

classical-size reduction scheme

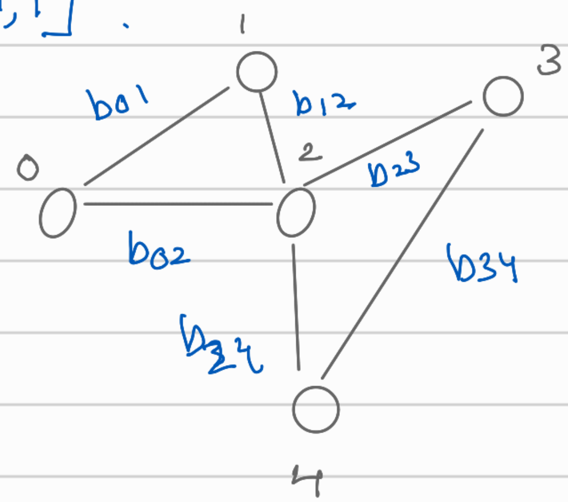
THE ALGORITHM

i) Correlation $\longrightarrow$ ii) Reduced problem size $\rightarrow$ iii) QAOA $\rightarrow$ iv) Reconstruct.

A) Computing Correlations To reduce the problem size of an instance too large to be solved by current quantum hardware, the algorithm relies on correlations.

A correlation between a vertex pair quantifies the tendency of the vertices pair being in equal or opposite partitions in an optimum cut.

For a subset $S \subseteq V \times V$ of vertex pairs, correlations are a set $\{b_{uv} \mid uv \in S\}$ where $b_{uv} \in [-1, 1]$.



$S = \{01, 12, 23, 02, 34, 24\}$
set of vertex pair

b_{uv} for each $uv \in S$

$b_{01}, b_{12}, b_{23}, b_{02}, b_{34}, b_{24} \in [-1, 1]$

Correlations are called optimal, if there is an optimum cut $\delta(W)$ such that $b_{uv} \geq 0$ ($b_{uv} < 0$) if u and v lie in equal (opposite) partitions in $\delta(W)$

$$b_{uv} := 1 - 2x^*_{uv} \in [-1, 1] \qquad x^* \text{-solution}$$

if $b_{uv} \geq 0$ uv lies in the same partition
$b_{uv} < 0$ uv lies in opposite partition

x^* is the result, that means $x^* = 1$ if they are in the different

x is the result, then means $x_{uv} = 1$ if they are in the output and $=0$ otherwise.

B) Shrinking We reduce the problem size by identifying vertex pairs which have a large absolute correlation.

$$\sigma := \begin{cases} \operatorname{sign}(b_{uv}) & b_{uv} \neq 0 \\ 1 & b_{uv} = 0 \end{cases}$$

If $\sigma = 1$ ($\sigma = -1$), we enforce u and v to lie in equal(opposite) partition.