Ch8: The Quantum Approximate Optimization Algorithm

Problem definition

A combinatorial optimization problem consists in the search for a solution in a discrete set (the feasible set) of possible solutions such that a (possibly multidimensional) function is optimized.

Given binary constraints:

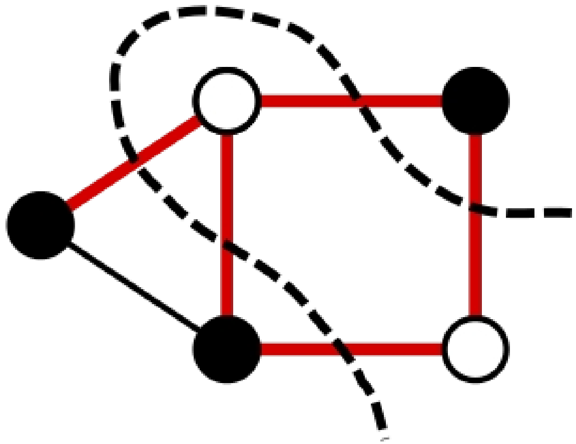
$$x \in \{0,1\}^n \qquad C_a(x) = \begin{cases} 1 & \text{if } x \text{ satisfies the constraint } a \\ 0 & \text{if } x \text{ doesn't} \end{cases}$$

maximize: $$C(x) = \sum_a C_a(x)$$

Some examples of combinatorial optimization problems include travelling saleman, vehicle routing, and graph coloring problems.

The MaxCut problem

To provide a concrete example of, let us consider the MaxCut problem. As input, the MaxCut problem takes a graph $G=(V,E)$, which is characterized by a set of nods V and a set of undirected edges E. The task is to partition the nodes into two sets, such that the number of edges crossing these sets is maximized



The above figure provides an illustrative example, in which a graph with five nodes and six edges is partitioned into two sets that result in a cut of size five.

The MaxCut problem

In general, the MaxCut problem is characterized by the following unconstrained discrete maximization problem:

$$\text{maximize:} \sum_{(i,j)\in E} x_i(1 - x_j) \qquad\qquad x_i \in \{0,1\} \quad \forall i \in N$$

In this formulation, there is one binary decision variable for each node in the graph, indicating which set it belongs to. The objective function consists of one term for each edge in the graph.

This term is 0 if the the nodes of that edge take the same value and 1 otherwise. Consequently, the optimal solution of this problem will be a maximal cut of the graph G.

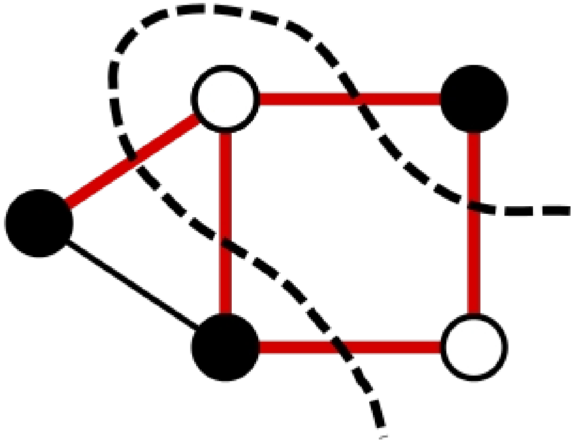
The MaxCut problem

Coming back to previous example:

Score $= 0$

Score $= 0$

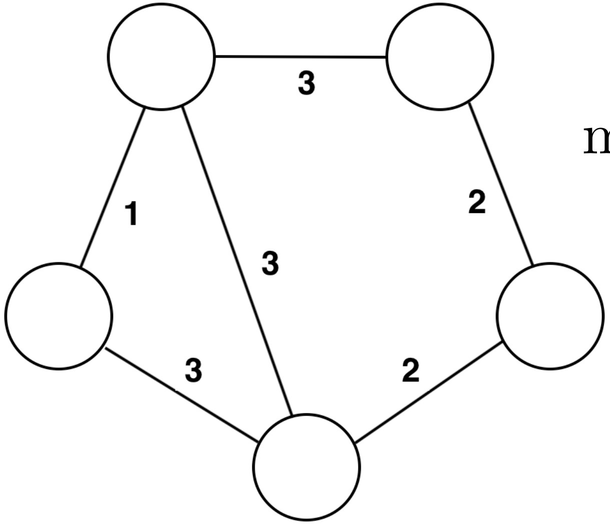
Score $= +1$



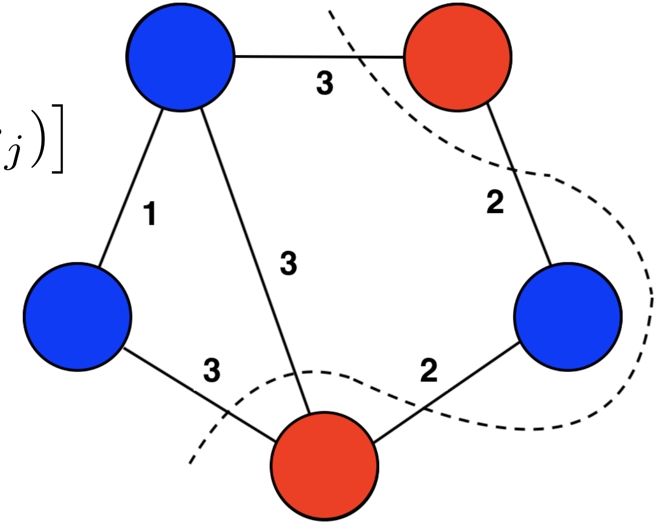
Score $= +5$ (max)

The weighted MaxCut problem

In this version each edge has a real number, its weight, and the objective is to maximize not the number of edges but the total weight of the edges between a first set of nodes and its complement:



$$\text{maximize:} \sum_{(i,j)\in E} w_{ij}\left[x_i(1-x_j)\right]$$



Cost Hamiltonian

The Quantum Approximate Optimization Algorithm (QAOA) leverages gate-based quantum computing for finding high-quality solutions to combinatorial optimization problems.

To apply this algorithm the user first translates the cost function into an equivalent quantum cost Hamiltonian. For max-cut:

$$x_i \longrightarrow \frac{1 - Z_i}{2} \qquad\qquad C(x) \longrightarrow H_C = \sum_{(i,j)\in E} \frac{1}{2}(1 - Z_i Z_j)$$

thus this Hamiltonian is diagonal in the computational basis.

8.2 The quantum approximate optimization algorithm

Adiabatic evolution

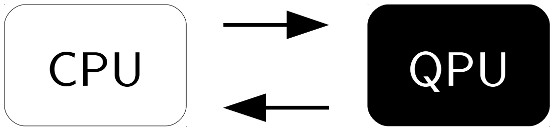
A way to implement cost Hamiltonians on quantum circuit is by adiabatic evolution:

$$H(t) = (1-t)\overset{\substack{\text{Driver Hamiltonian}\\ \downarrow}}{H_D} + t\underset{\substack{\uparrow\\ \text{Cost Hamiltonian}}}{H_C}$$

$$\begin{cases} H_D = \sum_i X_i \\ H_C = \sum_{(i,j)\in E} \frac{1}{2}(1 - Z_i Z_j) \end{cases}$$

In practice we exponentiate and parametrize in p steps by p betas and p gammas:

$$U_{ansatz} = e^{-i\beta_p H_D} e^{-i\gamma_p H_C} \ldots e^{-i\beta_0 H_D} e^{-i\gamma_0 H_C}$$



where betas and gammas are to optimize with a classical optimizer (see notebook).

Algorithm steps

Algorithm 12 Quantum approximate optimization algorithm

Input:

- Number of rounds of optimization r
- Two size r array of angles, γ and β.
- Hamiltonians C_α corresponding to the clauses of the optimization problem.

Output:

- An approximation to the solution of problem in Eq. (66).

Procedure:

Step 1. Construct the n-qubit uniform superposition state by applying $H^{\otimes n}$ to $|0\ldots 0\rangle$

for $1 \le k \le r$ **do**

Step 2a. Apply $\prod_{\alpha=1}^{m} e^{-i\gamma[k]C_\alpha}$

Step 2b. Apply $\prod_{j=1}^{n} e^{-i\beta[k]X_j}$

end for

Step 3. We will call the state so constructed $|\beta,\gamma\rangle$. The expectation value, $\sum_{\alpha=1}^{m} \langle\beta,\gamma|C_\alpha|\beta,\gamma\rangle$, gives an approximate solution to the problem.