Ciência de Dados Quântica 2022/23

QUBO 4 QAOA:

Quantum **A**pproximate **O**ptimization **A**lgorithm

LUÍS PAULO SANTOS

Material de Consulta

- Fred Glover, Gary Kochenberger, Yu Du;
  Quantum Bridge Analytics I: A Tutorial on Formulating and Using QUBO Models
  https://arxiv.org/pdf/1811.11538.pdf

- Qiskit Summer School 2021: 5.2 – Introduction to QAOA and Applications

- Qiskit Summer School 2021: Lab2 – Introduction to Variational Algorithms

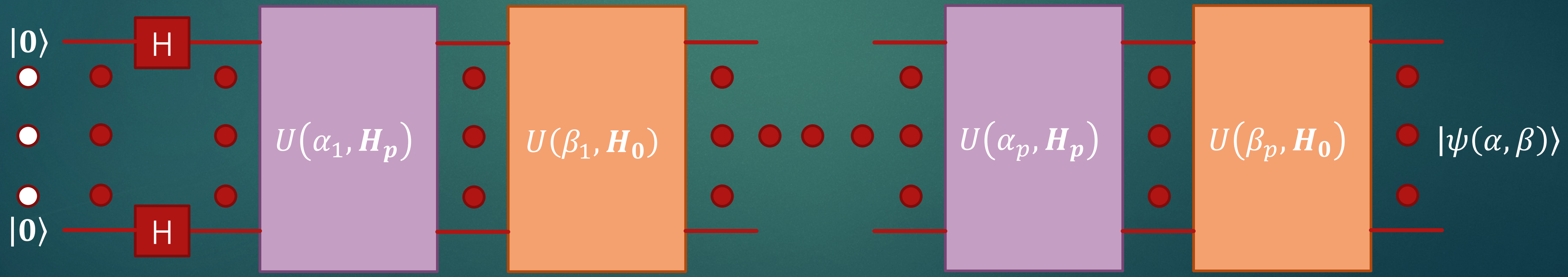
Combinatorial Optimization

- Combinatorial optimization problems:
  finding an optimal object out of a finite set of objects.
- Our formulation:
  finding optimal bit strings, $z = \{0,1\}^{\otimes n}$, out of a set of finite bitstrings
- These problems are NP-hard

QAOA operator

- Let $\boldsymbol{H_0}$ be the mixer Hamiltonian and $\boldsymbol{H_p}$ be the problem Hamiltonian
- Then $U(\beta, \boldsymbol{H_0}) = \boldsymbol{e^{-i\beta H_0}}$ and $U(\alpha, \boldsymbol{H_p}) = \boldsymbol{e^{-i\propto H_p}}$ are the unitaries, with α and β capturing both time evolution and the $\frac{1}{p}$ term associated with trotterization
- Finally we get:

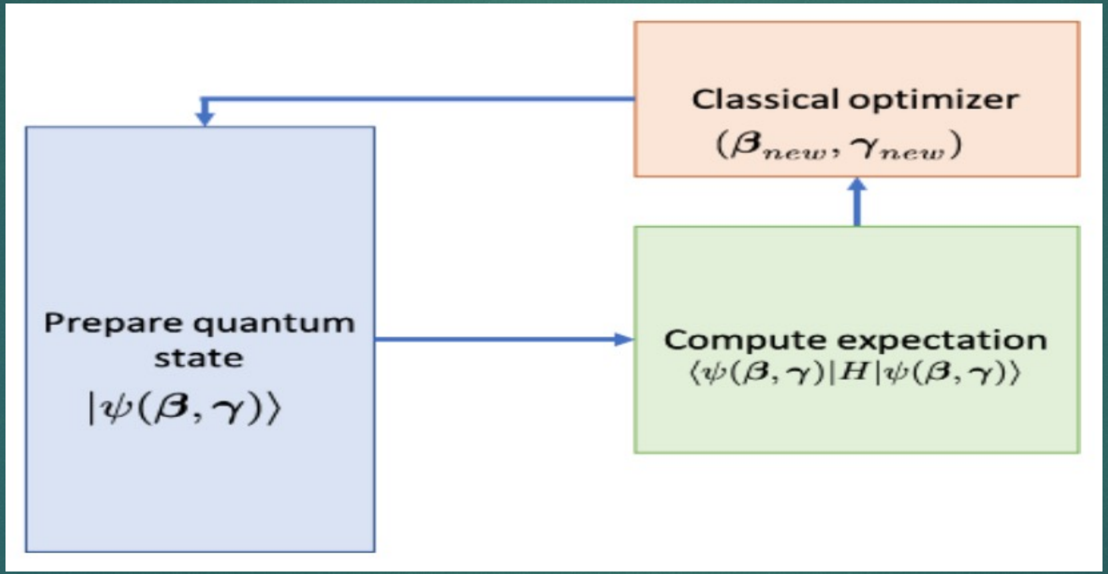
$$|\psi(\alpha,\beta)\rangle = U(\beta_p, \boldsymbol{H_0})U(\alpha_p, \boldsymbol{H_p}) \cdots U(\beta_1, \boldsymbol{H_0})U(\alpha_1, \boldsymbol{H_p})H|0\rangle$$



QAOA: overall

- find (α^*, β^*) such that the expectation of H_p is minimized:

$$|\psi(\alpha^*, \beta^*)\rangle = \underset{\alpha,\beta}{\operatorname{argmin}} \langle|\psi(\alpha, \beta)| H_p\, |\psi(\alpha, \beta)\rangle$$



- sample basis states $|z\rangle$ from $|\psi(\alpha^*, \beta^*)\rangle$ to find a solution

Problem Statement

- We want to use near term quantum hardware to solve classical combinatorial optimization problems
- Adiabatic Computing (in general) and QAOA (in particular) seem like good candidates to investigate whether a quantum advantage can be unleashed
- But how do we transform a classical combinatorial problem into the specification of an Hamiltonian suitable for the QAOA framework?

Problem Statement

- It has been found that the framework of **Quadratic Unconstrained Binary Optimization (QUBO)** can embrace a wide set of important combinatorial optimization problems

- Once formulated as a QUBO these problems can be efficiently solved using QUBO solvers, including QAOA within the quantum context

- QUBO is a special case of Quadratic Programming

Quadratic Programming

- $\min_{x}(x^T Qx + c^T x), \quad x \in R^N, \quad c \in R^N, \quad Q \in R^{N*N}$
- subject to
  - $Ax \leq b$ linear constraints
  - $x^T Qx + c^T x \leq r_i$ quadratic constraints
  - $l_j \leq x_j \leq u_j$ range constraints

QUBO – Quadratic Unconstrained Binary Optimization

- $\min_z (z^T Q z + c^T z), \quad z \in \{0,1\}^n, \quad c \in R^N, \quad Q \in R^{N*N}$
- subject to
  - $Ax \leq b$ linear constraints
  - $x^T Q x + c^T x \leq r_i$ quadratic constraints
  - $l_j \leq x_j \leq u_j$ range constraints
- Binary variables
- no variable constraints
  - although linear constraints $Ax = b$ can be supported

QUBO – Basic definitions

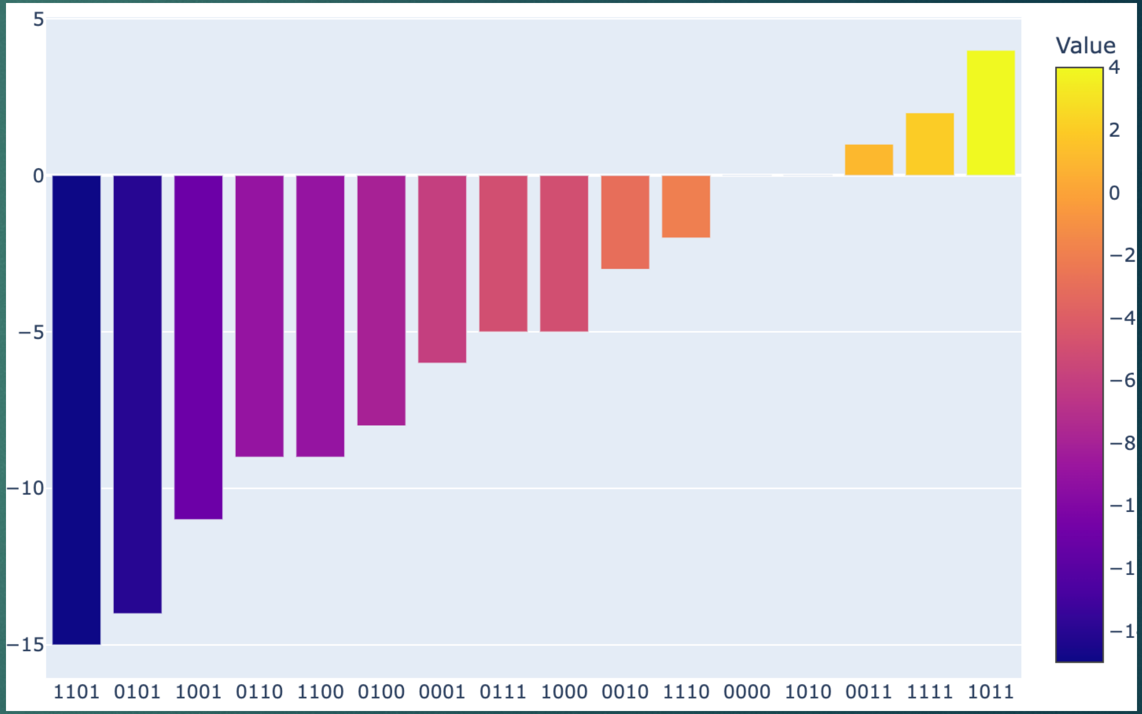
- $\min_z (z^T Q z + c^T z), \quad z \in \{0,1\}^n, \quad c \in R^N, \quad Q \in R^{N*N}$
- $c \in R^N$, is a vector containing the coefficients of the linear terms of the objective function, i.e. c_i is the coefficient of the term z_i
- $Q \in R^{N*N}$, is a square symmetric matrix containing the coefficients of the quadratic terms of the objective function, i.e. q_{ij} is the coefficient of the term $z_i z_j$
- Q is symmetric since $z_i z_j = z_j z_i$, for binary variables z_t
  - If Q is not given on a symmetric form it can always be made symmetric by redefining it:

$$q_{ij} = (q_{ij} + q_{ji})/2, \qquad \forall i, j; j \neq i$$

QUBO – Basic example I

$$\min_z(\underbrace{-5z_1 - 8z_2 - 3z_3 - 6z_4}_{\text{linear}} + \underbrace{4z_1z_2 + 8z_1z_3 + 2z_2z_3 + 10z_3z_4}_{\text{quadratic}})$$

We can evaluate $f(z)$ for $\forall z$, obtaining



$$\min_z\left(\{z_1 \quad z_2 \quad z_3 \quad z_4\}\begin{bmatrix}0 & 2 & 4 & 0\\ 2 & 0 & 1 & 0\\ 4 & 1 & 0 & 5\\ 0 & 0 & 5 & 0\end{bmatrix}\begin{Bmatrix}z_1\\ z_2\\ z_3\\ z_4\end{Bmatrix} + \{-5 \quad -8 \quad -3 \quad -6\}\begin{Bmatrix}z_1\\ z_2\\ z_3\\ z_4\end{Bmatrix}\right)$$

From QUBO to QAOA

$$C(z) = \sum_{i=1, j=1, i \neq j}^{n} z_i \, q_{ij} z_j + \sum_{i=1}^{n} c_i z_i$$

$$H_p = \sum_{i=1, j=1, i \neq j}^{n} \frac{1}{4} q_{ij} \, Z_i \, Z_j - \sum_{i=1}^{n} \frac{1}{2} \left(c_i + \sum_{j=1}^{n} q_{ij} \right) Z_i + \underbrace{\left(\sum_{i=1, j=1, i \neq j}^{n} \frac{q_{ij}}{4} + \sum_{i=1}^{n} \frac{c_i}{2} \right)}_{\text{Drop}}$$

The Hamiltonian is given by a sum of Pauli Z's, as expected

QUBO – Basic example I

$$\min_{z}\left(\{z_1 \quad z_2 \quad z_3 \quad z_4\}\begin{bmatrix}0 & 2 & 4 & 0\\ 2 & 0 & 1 & 0\\ 4 & 1 & 0 & 5\\ 0 & 0 & 5 & 0\end{bmatrix}\begin{Bmatrix}z_1\\ z_2\\ z_3\\ z_4\end{Bmatrix}+\{-5 \quad -8 \quad -3 \quad -6\}\begin{Bmatrix}z_1\\ z_2\\ z_3\\ z_4\end{Bmatrix}\right)$$

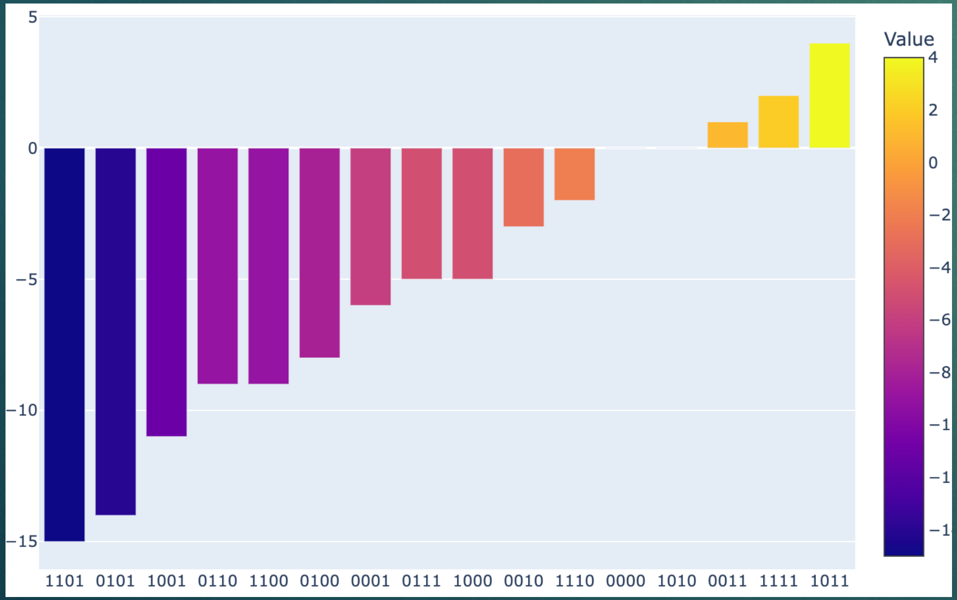
$$H_p = \sum_{i=1,j=1,i\neq j}^{n} \frac{1}{4} q_{ij}\, Z_i\, Z_j - \sum_{i=1}^{n} \frac{1}{2}\left(c_i + \sum_{j=1}^{n} q_{ij}\right) Z_i$$

$$H_p = Z_1\, Z_2 + 2\, Z_1\, Z_3 + 0.5\, Z_2\, Z_3 + 2.5\, Z_3\, Z_4 + 0.5\, Z_1 - 2.5\, Z_2 + 3.5\, Z_3 - 0.5\, Z_4$$

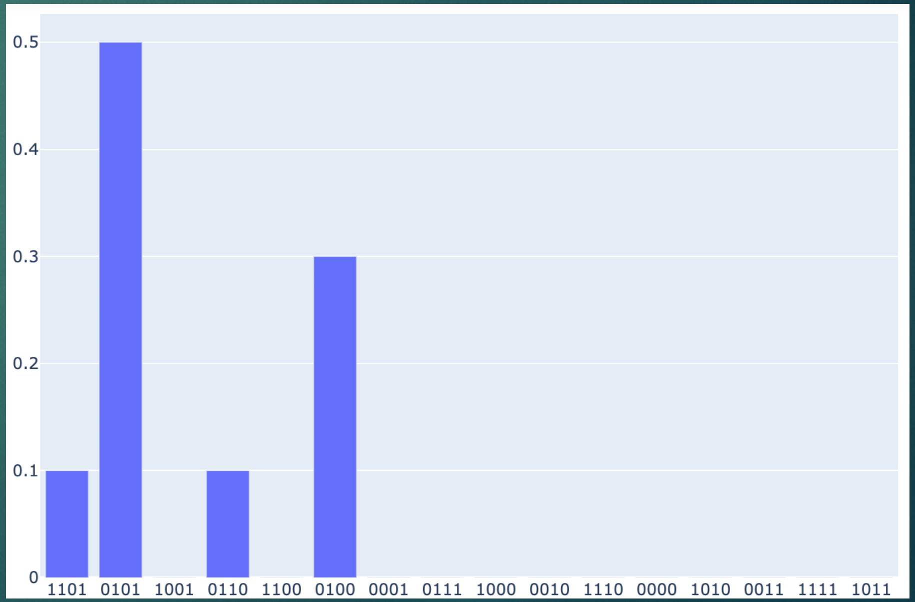
QUBO – Basic example I

$$H_p = Z_1\, Z_2 + 2\, Z_1\, Z_3 + 0.5\, Z_2\, Z_3 + 2.5\, Z_3\, Z_4 + 0.5\, Z_1 - 2.5\, Z_2 + 3.5\, Z_3 - 0.5\, Z_4$$

$$U(\propto_t, H_p) = e^{-i \propto_t H_p} = R_{Z_1\, Z_2}(\propto_t) * R_{Z_1\, Z_3}(2 \propto_t) * R_{Z_2\, Z_3}(0.5 \propto_t) * R_{Z_3\, Z_4}(2.5 \propto_t) *$$
$$* R_{Z_1}\,(0.5 \propto_t) * R_{Z_2}\,(-2.5 \propto_t) * R_{Z_3}\,(3.5 \propto_t) * R_{Z_4}\,(-0.5 \propto_t)$$



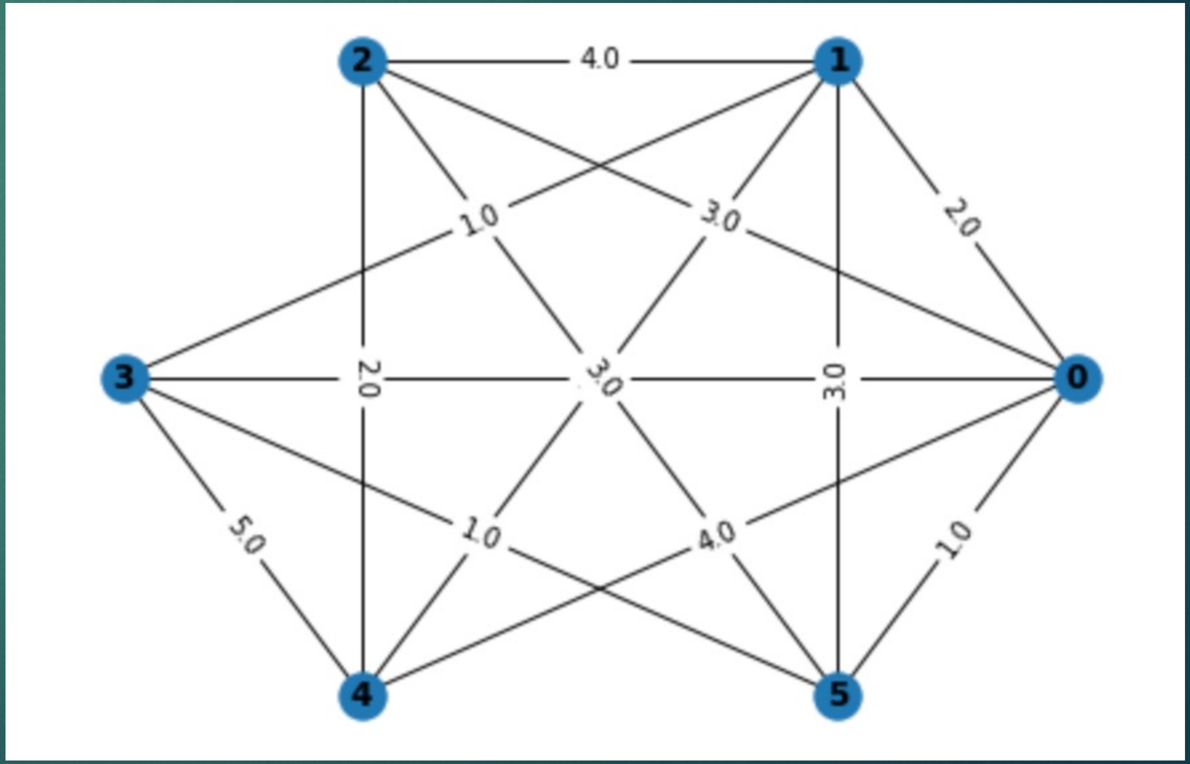
p=2



QUBO – MaxCut

- Given the weighted adjacency matrix of a graph cut it onto two subgraphs, such that the cut is maximal

```
w =np.array([[0., 2., 3., 2., 4., 1.],
             [2., 0., 4., 1., 1., 3.],
             [3., 4., 0., 0., 2., 3.],
             [2., 1., 0., 0., 5., 1.],
             [4., 1., 2., 5., 0., 0.],
             [1., 3., 3., 1., 0., 0.]])
```



QUBO – MaxCut

- The cost function is given by :

$$C(z) = \sum_{i,j,i\neq j}^{n} w_{ij} * z_i(1 - z_j) \qquad \Leftrightarrow \qquad C(z) = \sum_{i,j,i\neq j}^{n} w_{ij} * z_i - \sum_{i,j,i\neq j}^{n} w_{ij} * z_i z_j$$

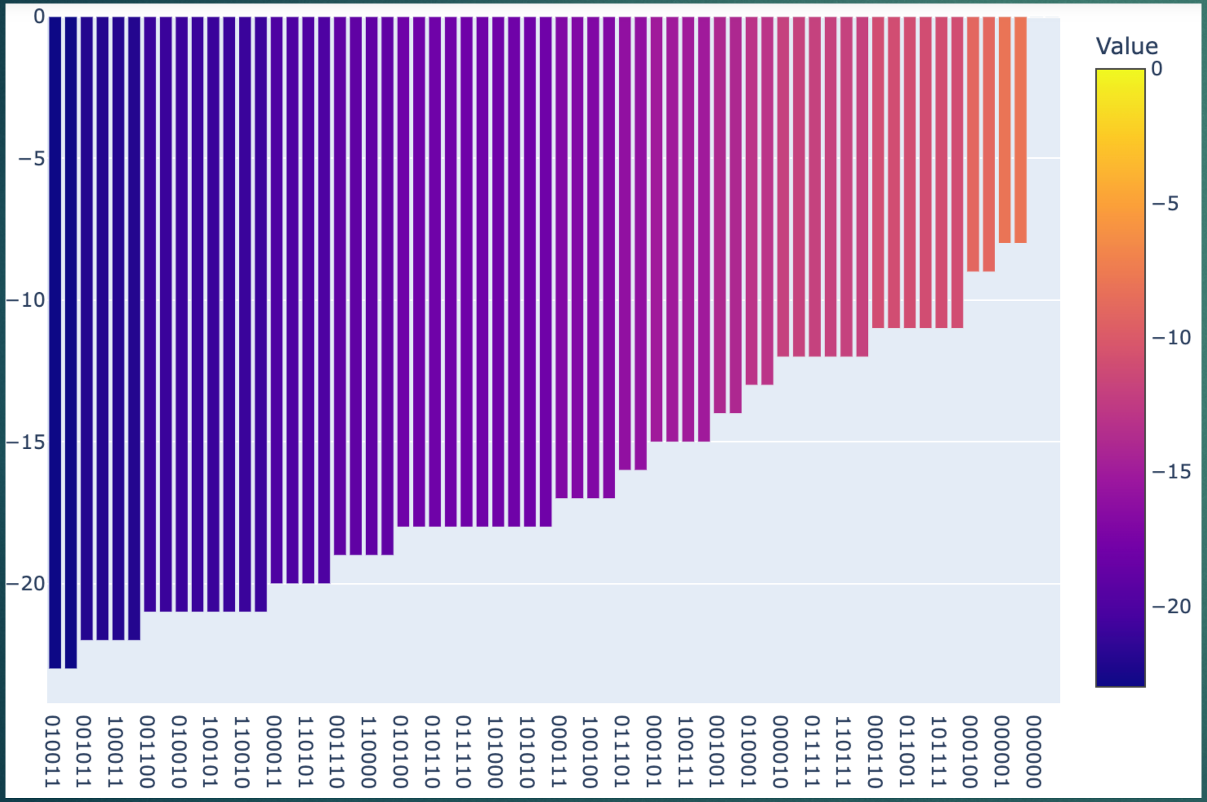
- spinning the sign to convert to a minimization problem

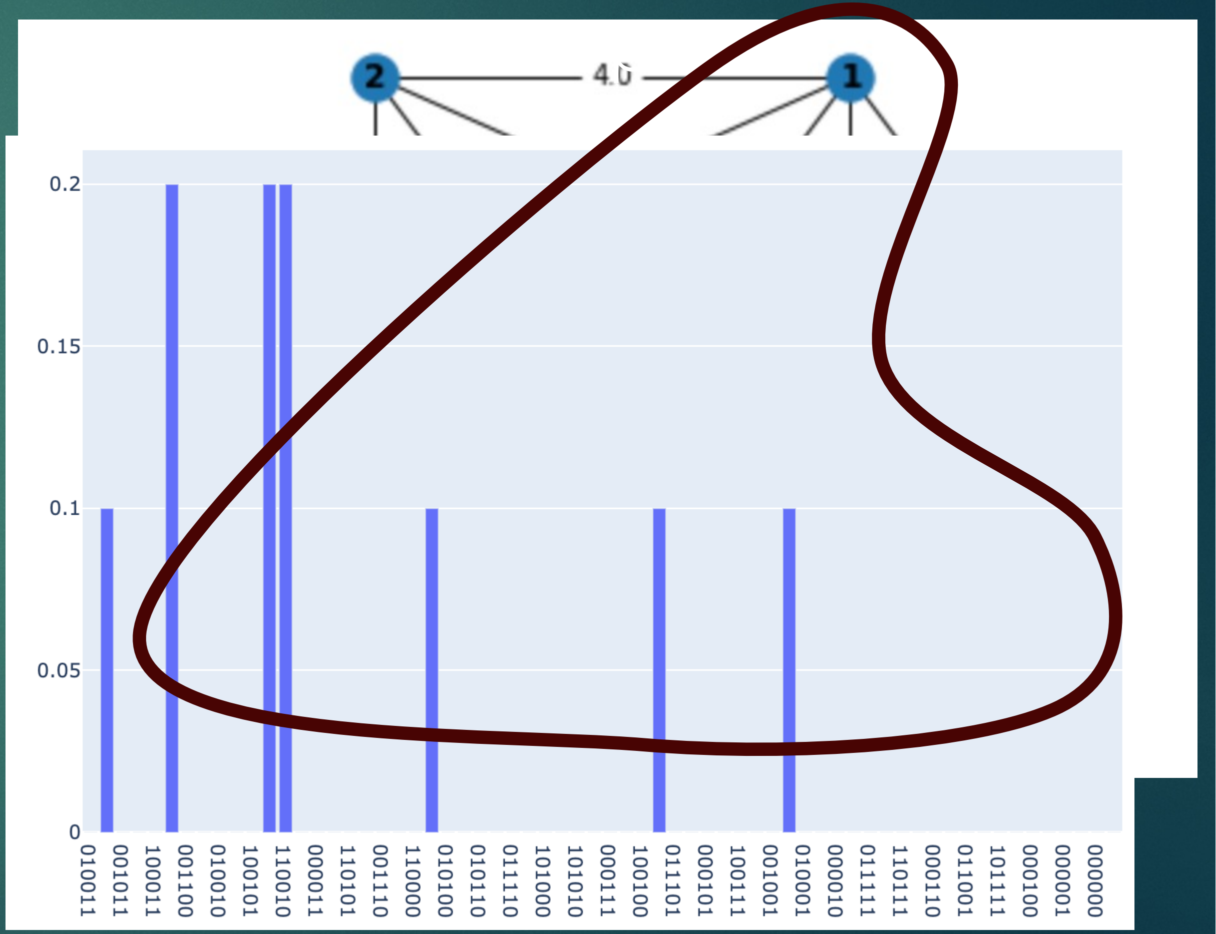
$$C(z) = \sum_{i,j,i\neq j}^{n} -w_{ij} * z_i + \sum_{i,j,i\neq j}^{n} w_{ij} * z_i z$$

- write in the QUBO representation

$$c_i = \textstyle\sum_{j=1}^{n} -w_{ij} \qquad\qquad q_{ij} = w_{ij},\ i \neq j$$

QUBO – MaxCut





QUBO – Number Partitioning

- Given a set $\mathcal{S}$ of numbers partition to into two disjoint subsets, $\mathcal{S}_0$ and $\mathcal{S}_1$, such that the sum of their elements is minimal:

$$\min \left| \sum_{s \in \mathcal{S}_0} s - \sum_{s \in \mathcal{S}_1} s \right|$$

- Let $\mathcal{S} = \{s_1, \dots, s_n\}$ and $z_i \in \{0,1\}, i = 1 \dots n$ indicate to which subset s_i belong to
- $sum\ \mathcal{S}_1 = \sum_{i=1}^{n} s_i * z_i$; $sum\ \mathcal{S}_0 = \sum_{i=1}^{n} s_i - \sum_{i=1}^{n} s_i * z_i$
- Let S $= \sum_{i=1}^{n} s_i$ then $sum\ \mathcal{S}_0 - sum\ \mathcal{S}_1 = S - \sum_{i=1}^{n} s_i * z_i$

- Let our objective function be

$$\mathrm{C(z)} = \left(S - \sum_{i=1}^{n} s_i * z_i \right)^2$$

QUBO – Number Partitioning

$$\mathrm{C(z)} = \left(S - \sum_{i=1}^{n} s_i * z_i\right)^2$$

$$C(z) = S^2 - 2S\sum_{i=1}^{n} s_i * z_i + \left(\sum_{i=1}^{n} s_i * z_i\right)^2$$

$$\left(\sum_{i=1}^{n} s_i * z_i\right)^2 = \sum_{i=1}^{n} {s_i}^2 * z_i + \sum_{i,j=1,i\neq j}^{n} 2s_i s_j * z_i z_j \qquad {z_i}^2 = z_i, z_i \in \{0,1\}$$

$$C(z) = S^2 - 2S\sum_{i=1}^{n} s_i * z_i + \sum_{i=1}^{n} {s_i}^2 * z_i + \sum_{i,j=1,i\neq j}^{n} 2s_i s_j * z_i z_j$$

$$C(z) = S^2 + 4\sum_{i=1}^{n} s_i(s_i - S) * z_i + 8\sum_{i,j=1,i\neq j}^{n} s_i s_j * z_i z_j$$

QUBO – Number Partitioning

$$\min_{z} C(z) = \min_{z}\left(\sum_{i=1}^{n} s_i(s_i - S) * z_i + 2\sum_{i,j=1,i\neq j}^{n} s_i s_j * z_i z_j\right)$$

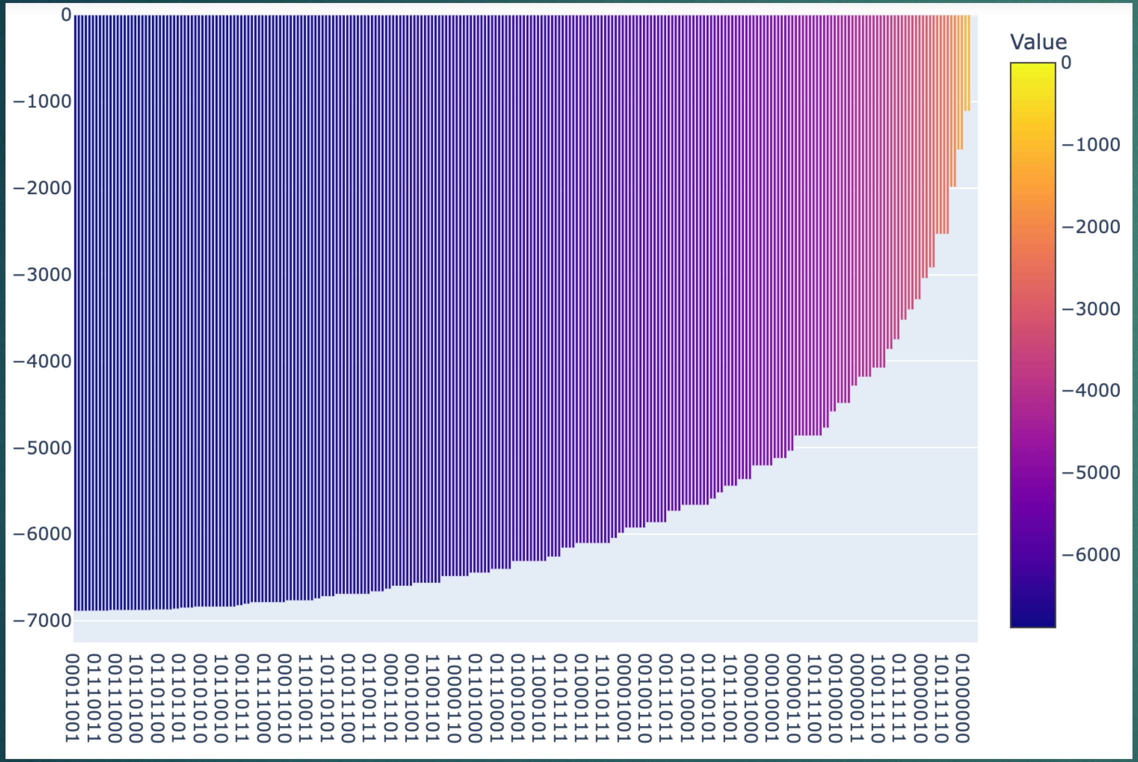
$c_i = s_i(s_i - S)$ $\qquad$ $q_{ij} = s_i s_j$

Let $\mathcal{S} = \{25, 7, 13, 31, 42, 17, 21, 10\}$

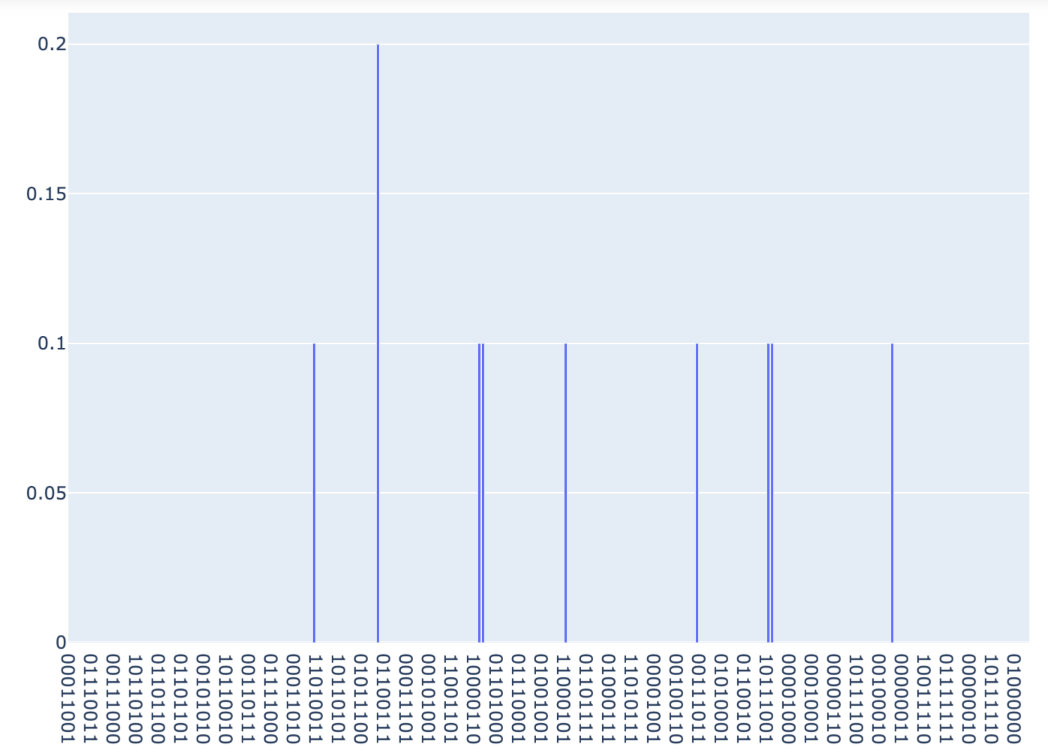
One optimal solution:

- $z = \{0,\ 0,\ 0,\ 1,\ 1,\ 0,\ 0,\ 1\}$
- $\mathcal{S}_0 = \{25, 7, 13, 17, 21\}; sum = 80$
- $\mathcal{S}_1 = \{31, 41, 10\}; sum = 81$

QUBO – Number Partitioning



p=2



QUBO with linear equality constraints

- Consider a QUBO $z^T Q' z + c'^T z$
- subject to M constraints of the form $A_m z = b_m$

- Each constraint m can be converted into a quadratic constraint of the form:

$$P\left(\sum_{i=1}^{n}(a_{mi}z_i) - b_m\right)^2 = P\left[\sum_{i=1}^{n}\left(a_{mi}^2 - 2a_{mi}b_m\right)z_i + \sum_{i,j=1,j\neq i}^{n}\left(2a_{mi}a_{mj}\right)z_i z_j + b_m^2\right]$$

- The quadratic constraints are added to the original QUBO to build a “constrained” QUBO

Fred Glover, Gary Kochenberger, Yu Du;
Quantum Bridge Analytics I: A Tutorial on Formulating and Using QUBO Models
https://arxiv.org/pdf/1811.11538.pdf

QUBO with linear equality constraints

$$\min_z (20z_1 + 5z_2 + 7z_3 + 10z_4)$$

s.t.

$$10z_1 + 5z_2 + 5z_3 + 5z_4 = 10$$

- P=1
- $\min_z \begin{pmatrix} 20z_1 + 5z_2 + 7z_3 + 10z_4 - 100z_1 - 75z_2 - 75z_3 - 75z_4 + \\ 100z_1z_2 + 100z_1z_3 + 100z_1z_4 + \\ 50z_2z_3 + 50z_2z_4 + 50z_3z_4 \end{pmatrix}$

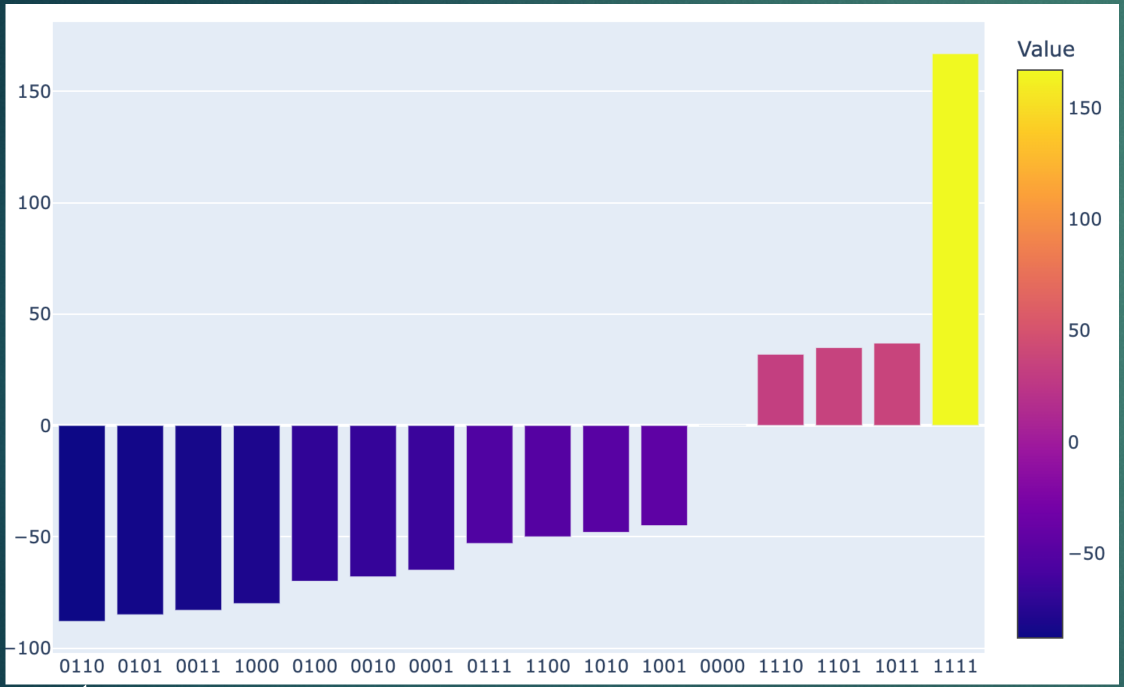
$$\min_z \begin{pmatrix} -90z_1 - 70z_2 - 68z_3 - 65z_4 + \\ 100z_1z_2 + 100z_1z_3 + 100z_1z_4 + \\ 50z_2z_3 + 50z_2z_4 + 50z_3z_4 \end{pmatrix}$$

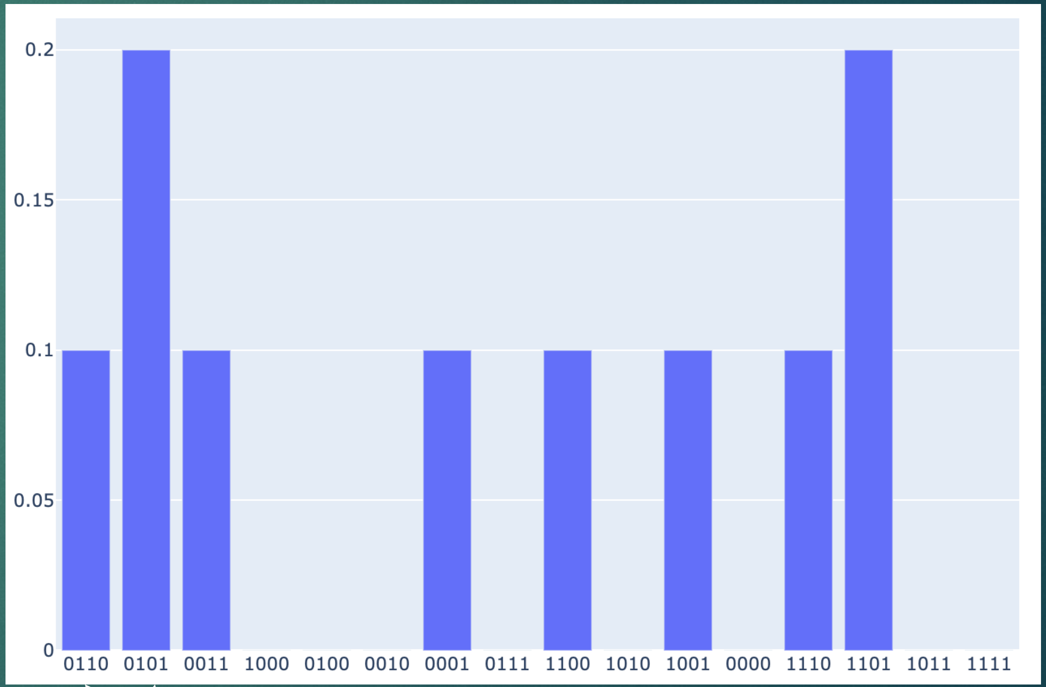
QUBO with linear equality constraints

$$\min_z (20z_1 + 5z_2 + 7z_3 + 10z_4)$$

s.t. $10z_1 + 5z_2 + 5z_3 + 5z_4 = 10$

p=2 ; P = 1





0110: min = 12 ; constraint 10=10

0101: min = 15 ; constraint 10=10