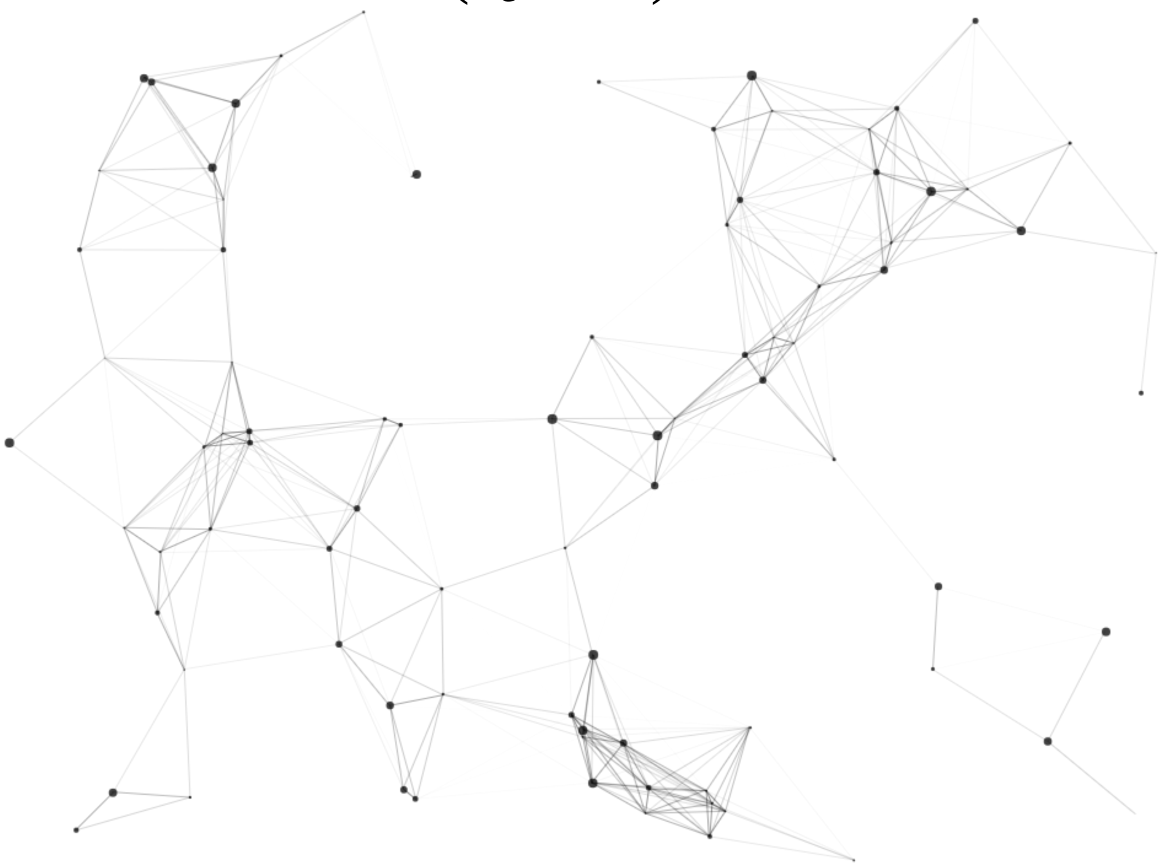
A PROJECT REPORT ON

QUADRATIC UNCONSTRAINT BINARY OPTIMIZATION

(QUBO)



QRD LAB

By

Sourav Ghosh

BASIC STUDIES

That I have gone through for doing the project

- Insight of a quantum computer
- What is qubit?
- Basic Quantum Mechanics
- Quantum circuit
- Quantum Gates
    - Single qubit gats
    - Control gates
- Deustch-Jozsa Algorithm
- Bernstein –Vizirani Algorithm
- What are QUBOs?
- What are the real life implication of QUBOs?
- How can we solve QUBOs easily with a quantum computer?

STEP 1: SELECTED A QUBO

With five binary variables , n=5

$$\max_{x\in\{0,1\}^n} \sum_{(i,j)\in E} w_{ij}(x_i + x_j - 2x_i x_j)$$

Then added a constraint to it because real life problems has constraints.

$$\max_{x\in\{0,1\}^n} \sum_{(i,j)\in E} w_{ij}(x_i + x_j - 2x_i x_j)$$

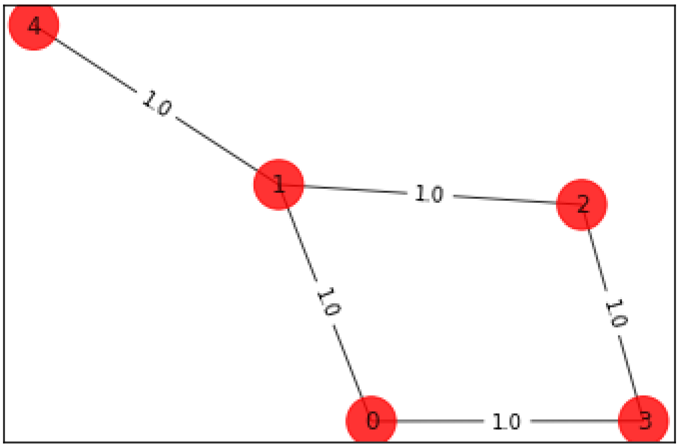
$$\text{subject to: } \sum_{i=0}^{n-1} x_i = b$$

STEP 2: Removing the constraints with added penalty term. With DOcplex

```
\ This file has been generated by DOcplex
\ ENCODING=ISO-8859-1
\Problem name: MaxCut

Maximize
 obj: 6 x0 + 7 x1 + 6 x2 + 6 x3 + 5 x4 + [ - 2 x0^2 - 8 x0*x1 - 4 x0*x2
      - 8 x0*x3 - 4 x0*x4 - 2 x1^2 - 8 x1*x2 - 4 x1*x3 - 8 x1*x4 - 2 x2^2
      - 8 x2*x3 - 4 x2*x4 - 2 x3^2 - 4 x3*x4 - 2 x4^2 ]/2 -4
Subject To

Bounds
 0 <= x0 <= 1
 0 <= x1 <= 1
 0 <= x2 <= 1
 0 <= x3 <= 1
 0 <= x4 <= 1

Binaries
 x0 x1 x2 x3 x4
End
```

The penalty added equation looks like above.

Now to visualize the problem we have embedded this problem in a five variable maxcut problem.



For demonstrating we have taken the weights as 1 and we have chosen the edges as.

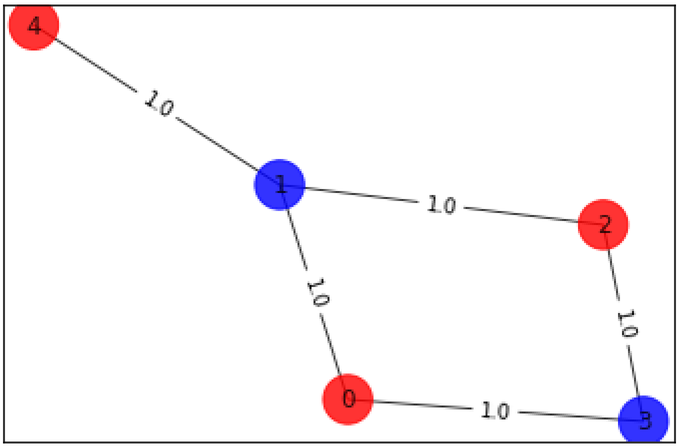
```
edges= [(0,1,1.0),(0,3,1.0),(1,2,1.0),(2,3,1.0),(1,4,1.0)]
```

We can change the weights and edges by our need.

▪ Solved the QUBO with help of DOcplex

```
from qiskit.optimization import QuadraticProgram
from qiskit.optimization.algorithms import CplexOptimizer
#created a quadratic program
qp=QuadraticProgram()
#we have created the quadratic model via Docplex, now we have imported the model into the quadritic program
qp.from_docplex(mdl)
#now using CPLEX we are solving the problem
cplex = CplexOptimizer()
result=cplex.solve(qp)
#printing the result and plotting it
print(result)
plot_result(G, result.x)
```

The result came as we have expected



The 1 and 3 node is in same group and the 0,2,4 is in different group . that's why the total number of edges between one to another group is larger that is 5.

The outcome of docplex is.

```
optimal function value: 5.0
optimal value: [ 0.  1. -0.  1. -0.]
status: SUCCESS
```

Optimizaation via Ising Hamiltonian.

Mapping of QUBO to Ising Hamiltonian

Suppose a QUBO

$$\min_{x \in \{0,1\}^n} \sum_{i,j=0}^{n-1} A_{ij} x_i x_j + c$$

1. Substitute
   $x_i = (1 - z_i)/2$,
   where $z_i \in \{-1, +1\}$.

2. Replace
   $z_i z_j = \sigma_Z^i \otimes \sigma_Z^j$, and
   $z_i = \sigma_Z^i$,
   where σ_Z^i denotes the Pauli Z-matrix $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ on the i-th qubit.

From QUBO to Finding Hamiltonian Groundstates

Suppose a QUBO

$$\min_{x\in\{0,1\}^n} \sum_{i,j=0}^{n-1} A_{ij}x_i x_j + c$$

which has been mapped to an Ising Hamiltonian:

$$H = \sum_{i,j=0}^{n-1} \tilde{A}_{ij}\sigma_Z^i \otimes \sigma_Z^j + \sum_{i=0}^{n-1} \tilde{b}_i \sigma_Z^i + \tilde{c}$$

Finding the groundstate of H is equivalent to solving the QUBO:

$$\min_{|\psi\rangle} \langle\psi|H|\psi\rangle$$

Remark: H is diagonal, and thus, all computational basis states are eigenstates.

Converted QUBO to ising Hamiltonian

])

The Hamiltonian looked like

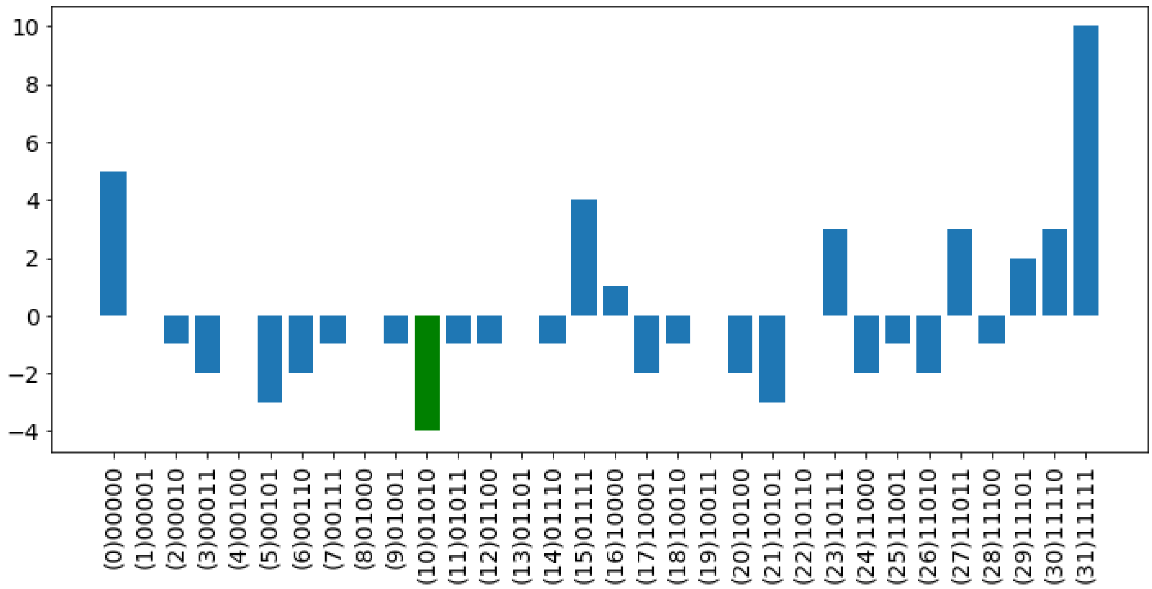
```
H= SummedOp([
  -0.5 * IIIIZ,
  -0.5 * IIIZI,
  -0.5 * IZIII,
  -0.5 * IIZII,
  -0.5 * ZIIII,
  IIIZZ,
  0.5 * IIZIZ,
  IIZZI,
  IZIIZ,
  0.5 * IZIZI,
  IZZII,
  0.5 * ZIIIZ,
  ZIIZI,
  0.5 * ZIZII,
  0.5 * ZZIII
])
```

** I = identity matrix, Z= Pauli Z matrix

H is a unitary matrix so the plotting the diagonal elements will give me idea about the eigenvalues. The eigenstate related to the minimum eigenvalues will be the solution.

Plotting of diagonal of the matrix.

```
H_matrix = np.real(H.to_matrix())  #converts the hamiltonian to matrix form
print('dim(H):', H_matrix.shape)  #printing the dimension of the matrix
print(H_matrix)                   # printing the matrix
print(H_matrix.diagonal())        # prining the diagonal of the matrix
print(min(H_matrix.diagonal()))   # printing the minimum element of the diagonal
# finding the indices or indexes of the diagonal of the matrix where the minimum element lies
dia_min_indices = list(np.where(H_matrix.diagonal()==min(H_matrix.diagonal())))[0]
print(dia_min_indices)
#plotting the diagonal elements by a bar graph
plt.figure(figsize=(12, 5))
plt.bar(range(2**n),H_matrix.diagonal())
plt.bar(dia_min_indices,H_matrix.diagonal()[dia_min_indices],color='g')#plotting minimum indices in green
plt.xticks(range(2**n),['('+str(i)+'){0:05b}'.format(i) for i in range(2**n)],rotation=90,fontsize=14)
plt.yticks(fontsize=14)
plt.show()
```



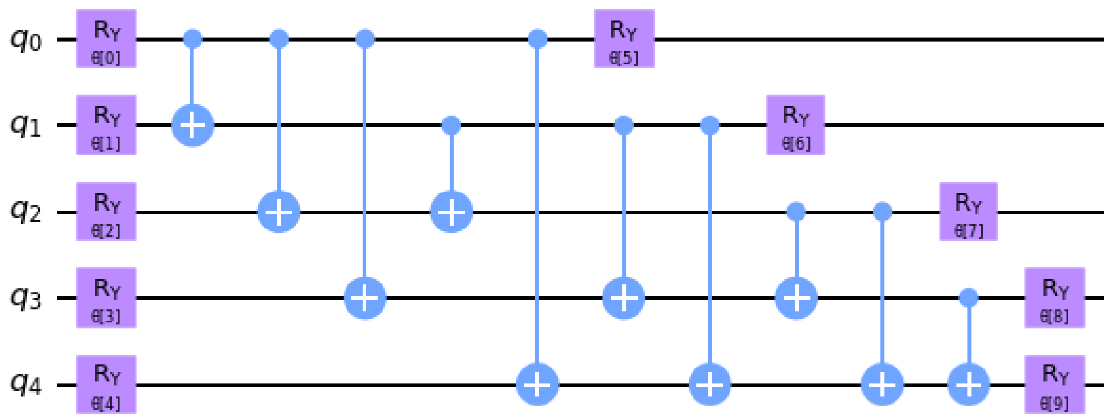
Now we have applied the Variational Quantum Eigensolver (VQE)

As we have taken the real valued part of the Hamiltonian so we have prepared a wave function that has only real amplitudes with the help of **RealAmplitudes**

Replace minimization over all $|\psi\rangle$ by minimization over a parametrized subset $|\psi(\theta)\rangle$:

$$\min_{\theta} \langle \psi(\theta) | H | \psi(\theta) \rangle$$

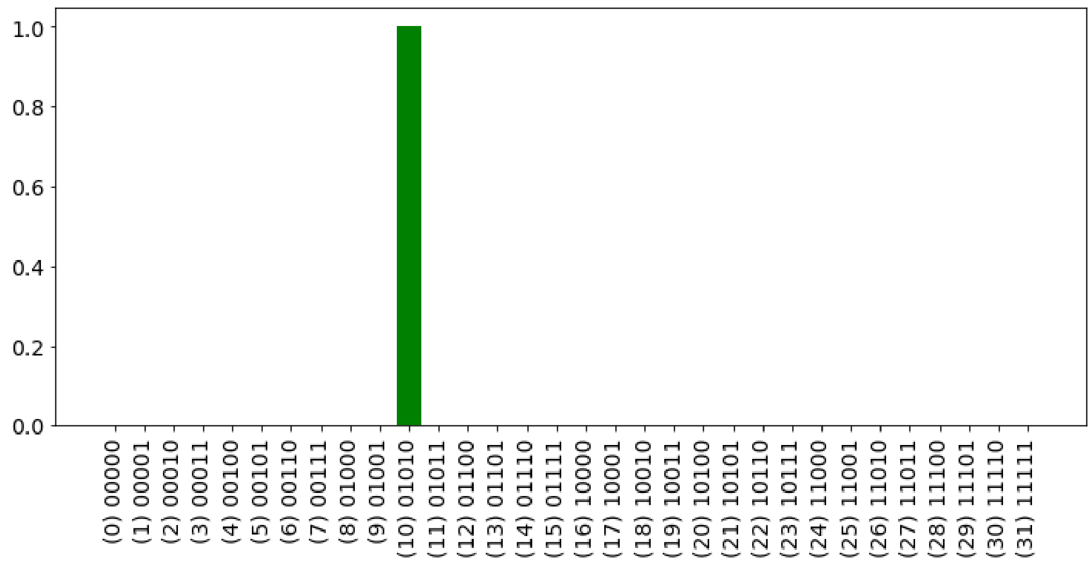
The circuit looks like



Then we have put the quantum state and the Hamiltonian in the VQE it and it gives us the minimum eigenstate and eigenvector.

```
from qiskit.aqua.algorithms import VQE
vqe = VQE(H,qc, quantum_instance= Aer.get_backend('statevector_simulator'))
#applied the hamiltonian to the prepared quantum state or ansatz
result=vqe.run()
print('optimal value:', np.round(result.eigenvalue, decimals=4))
probabilities = np.abs(result.eigenstate)**2
plt.figure(figsize=(12, 5))
plt.bar(range(2**n), probabilities)
plt.bar(dia_min_indices, probabilities[dia_min_indices], color='g')
plt.xticks(range(2**n), ['('+str(i)+') {0:05b}'.format(i) for i in range(2**n)], rotation=90, fontsize=14)
plt.yticks(fontsize=14)
plt.show()
```

Now we have again plotted the result in same format



The minimum eigenvalue is `optimal value: (-4+0j)`

We can see from the graph that the state connected to the minimum eigenvalue is 01010 that is the same result we have got from Docplex

01010 means 0 th,3 rd ,5 th qubit are in a group ans 2 nd , 4 th qubit are in a different group.