ECE 590/CSC 591:Quantum Computing Project Proposal Combinatorial Optimization: Subset Sum Problem

Kushal Batra
Department of CSC
NC State University
Raleigh, USA
kbatra2@ncsu.edu

Atharva Gupta
Department of ECE
NC State University
Raleigh, USA
agupta37@ncsu.edu

A. ***PROJECT WEBPAGE LINK***

B. *What Is Combinatorial Optimization?*

Combinatorial optimization is the process of searching for maxima (or minima) of an objective function F whose domain is a discrete but large configuration space (as opposed to an N-dimensional continuous space).

Some simple examples of typical combinatorial optimization problems are:

1) Traveling Salesman Problem
2) Bin-Packing
3) Subset Sum Problem
4) Knapsack Problem
5) Hamiltonian path problem

In this project we will be working on the Subset Sum Problem.

C. *What is Subset Sum Problem?*

The **Subset Sum Problem** belongs to category of decision problems. The problem states that given a set (or multi-set) of integers, is it possible to construct a non-empty subset whose sum is equal to zero? For example, given the set { -40, -2, 1, 2, 3, 4 ,5 ,6 } , the answer is "yes" because the subset { -2 ,2 } sums to zero.

The Subset Sum Problem belongs to complexity class of NP-complete, meaning it is easier to evaluate whether the final result obtained is correct or not. Also, this checking can be done in polynomial time. The problem itself takes non-deterministic polynomial time to find the solution. The subset sum problem is used in the field of complexity theory and cryptography.

In computational complexity theory, a problem is NP-complete when it can be solved by a restricted class of brute force search algorithms and it can be used to simulate any other problem with a similar algorithm. More precisely, each input to the problem should be associated with a set of solutions of polynomial length, whose validity can be tested quickly (in polynomial time), such that the output for any input is "yes" if the solution set is non-empty and "no" if it is empty [1]

The **Subset Sum Problem** can be thought of as a modification to **knapsack problem** where the limit of the bag is infinite and the assumption that the weight of the elements can be negative.

D. *Variations of Subset Sum Problem*

1) Subset Sum Problem where the sum is equal to some constant. This variation is extension of the question such that the sum instead of 0 can be any other integer k Example - Suppose the parent set is {-2 ,0 ,1 ,3}. and we are find subset such that sum equals 2. The answer is "yes" the subset {-2 ,1 ,3} gives the sum=2.
2) Subset Sum Problem where it is possible to subdivide this set into two strict subsets such that the sum of elements of one set is equal to another set. Example - Suppose the parent set is {-2 ,0 ,1 ,3}. The answer is "yes" the subsets are {-2 ,3} and {0 ,1 } where the sum of elements is 1

For simplicity we will just be working with the simple variation where the subset sum has to be equal to 0.

E. *Approach*

For our project we will try to implement the Subset Sum Problem problem using Quantum Computers. We will try to run it in Qiskit . So suppose we have 'x' elements in our set and we are supposed to find the subset with 'm' elements such that m¡=x and the sum of 'm' elements =0. So to work with Quantum Circuits we will have 'x' Qubits as our input to the circuit and then the output will be in terms of qubit value {0,1}. If the i^{th} line's output is '0' it signifies that the i^{th} element is not included in our answer. Similarly, If the i^{th} line's output is '1' it signifies that the i^{th} element is included in our answer.

In Classical Computing world, the problem of Subset Sum Problem has complexity of $O(2^n)$. The objective of our approach is to achieve Quantum Supremacy. Even if we are not able to show that the algorithm produces Quantum Supremacy it should at least be Quantum advantageous.In other words it is

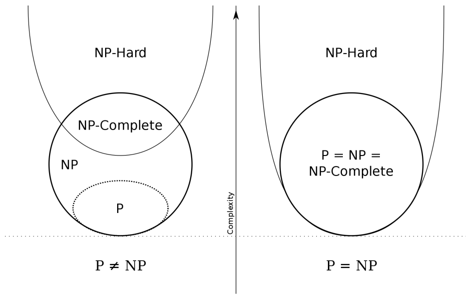

Fig. 1. Euler Diagram for P, NP, NP-Complete

sufficient enough for us to show that the complexity achieved from the Quantum Computing should be less than O(2^n).

F. Challenges

1) Figure out a way to construct circuit and determine what gates to use.
2) The Qiskit right now supports till 30 Qubits. So the maximum element number of elements in our array for our input should be way less than 30.
3) One of the biggest problem would be to optimize the use of ancilla bits so that we can use more and more qubits for the input.
4) Determine whether the problem falls under Quantum supremacy class or quantum advantage class.

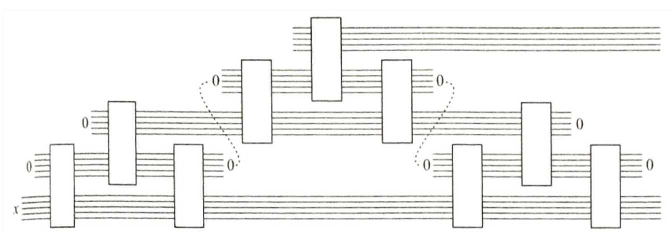

Fig. 2. Problem with Un-entanglement of Qubits

G. Timeline

TABLE I
TIMELINE

DATE	TASK
WEEK-1	Create a Proposal Report and a Web Page for our project.
WEEK 2	Identify the technical aspect of our project
WEEK-3	Start Implementing
WEEK-4	Take Feedback and Debug
WEEK-5	Complete implementation Update the Webpage
WEEK-6	Final Presentation

REFERENCES

[1] https://en.wikipedia.org/wiki/NP-completeness
[2] Bani-Ahmad, Sulieman. (2015). The Subset-Sum Problem: Revisited with an Improved Approximated Solution. International Journal of Computer Applications. 114. 10.5120/20043-7214.
[3] Bernstein, Daniel Jeffery, Stacey Lange, Tanja Meurer, Alexander. (2013). Quantum Algorithms for the Subset-Sum Problem. 10.1007/978-3-642-38616-9-2.
[4] Daskin, Ammar. (2017). A Quantum Approach to Subset-Sum and Similar Problems.
[5] Flaxman, Abraham & Przydatek, Bartosz. (2005). Solving Medium-Density Subset Sum Problems in Expected Polynomial Time. Lecture Notes in Computer Science. 3404. 305-314. 10.1007/978-3-540-31856-9_25.
[6] Zahedinejad, Ehsan & Zaribafiyan, Arman. (2017). Combinatorial Optimization on Gate Model Quantum Computers: A Survey.
[7] Lobe, Elisabeth. (2017). Solving Combinatorial Optimization Problems via D-Wave's Quantum Annealer.