Quantum Computers and NP-hard problems

Imagine that your mom's birthday is approaching and you want to buy her an expensive gift, to show how much you love her. But because you are a student, you are a little low on cash. So you decide to get a part time job as a delivery person. Going from house to house delivering parasols is a bit tiring, so you sit down one day and try to find the most efficient route through each delivery location in the city, but wait!! There seem to be almost infinite possible roots you could take and calculating the distance travelled in each of them will take you weeks. This is hard !! As a matter of fact this is NP-Hard !!
The problem described above is the Travelling Salesman problem which is also an example of an NP-complete problem. These problems currently do not have any time efficient algorithms, and are a subset of NP-Hard problems . Solving this would save millions of gallons of fuel and would help save the environment from excessive pollution due to vehicles.

These problems have teased mathematicians for the past two centuries. U'd imagine what sorcery would make it possible now. Well we do not have Sorcery or Magic. But what we do have is *Quantum Computers.* Quantum Computers use the peculiar properties of subatomic particles in order to provide an exponential speed up over the existing computers.

One peculiar thing about NP-complete problems is that their solutions might be hard to find but they are easy to verify. In layman terms, rather than solving the given problem if we are aware of the solutions that exist to the problem, they can be easily verified.
So maybe we can search all the possible options and then just check if they are correct or not. This is not at all doable for a normal computer but quantum computers are special and they can use the famous Grover Search Algorithm for testing all the possible cases in parallel using Quantum Superposition.

Q. But what is the Grover Search algorithm ?

Grover Search Algorithm was invented by Lov Grover in 1996 and is one of the Crown Jewels in the Quantum Computing field. The main reason for its fame is the exponential speedup that it provides. Classical Computation algorithms for finding a particular object in an unstructured database have a computational complexity of O(N) whereas the Grover Search Algo only requires O(sqrt(N)).

The Grover Search algorithm can be used to find a specific item in an unstructured database. The initial step is to encode all of the database components into quantum states, and then we generate a superposition of all of these quantum states. Now comes the important part: an oracle (a black box machine) is utilised to find the desired state among all possible states. It does this by establishing a phase difference on the quantum state we are looking for, which marks our target state; the amplitude of this target state is then increased, increasing the probability of us measuring this state. This process is repeated until the probability of finding the target state is high enough.

Let's try to understand it in a simpler way - Let's say you have a 1000 rubber balls and 1 metal ball. All the balls look identical(your unstructured database) and you want to find the metal ball without having to go through all the balls one by one.
An efficient way to solve this problem is to drop all the balls in a fluid. All the rubber balls with lower density will drown slower as compared to the heavy metal ball. The more depth of the fluid(repetition of the oracle) will lead to more difference in the height of the balls. Making it more clear which ball is the metal one.

The above example is analogous to the Grover Search Algorithm where your oracle is the fluid which highlights the state (in this case, the ball) that you are looking for. Furthermore repetition increases the chances of finding the right state.

This all seems okay but has anyone ever solved one of these problems using Quantum Computers?

Qiskit implementation on 3SAT Problem
Although solving the traveling salesman problem seems a bit hard, let's try to solve a little simpler NP-Complete problem.The 3 satisfiability is a mathematical problem which can be easily solved using IBM's Qiskit.

Problem
The problem can be stated as

$$f(a_1, a_2, a_3) = (\neg a_1 \vee a_2 \vee \neg a_3) \wedge (a_1 \vee a_2 \vee a_3) \wedge (a_1 \vee \neg a_2 \vee a_3)$$

Each "a" can be either 0 or 1. All the symbols have the following meanings

$\neg : NOT$ $\qquad$ $\vee : OR$ $\qquad$ $\wedge : AND$

The terms within the Parenthesis "()" are said to be clauses. So the above equation has 3 clauses and contains 3 literals inside it. As we have 3 literals the problem is called the 3-SAT problem.
The values of "a" for which make the right hand side equal to one are the solutions for the satisfiability problem.

Solution
The first step is to arrange the above problem into DIMACS format.

```
[13] dimacs = '''
     c example DIMACS CNF 3-SAT
     p cnf 3 5
     -1 2 -3 0
     1 2 3 0
     1 -2 3 0
     '''
```

Once we are done with that we can use QISKIT Aqua for calling `Grover` Search Algorithm along with the `LogicalExpressionOracle` . The Logical Expression Oracle uses the DIMACS format file to get the desired oracle for our particular case. Finally the algorithm is executed with the run command.

```
[15] result = grover.run(quantum_instance)
     print(result.assignment)

     [1, -2, 3]
```

As you can see the solution is [1,-2,3] in the DIMACS format.As a result, the solution in standard Boolean algebra can be expressed as [0,1,0]. Which is one of the solutions to our satisfiability problem.

Some of you might wonder why I did not use the plot_histogram command to plot all the solutions, but the output is not accurate, so I would recommend you not to use it.

The SAT problem looks purely abstract math but it in reality has a lot of industrial applications, from checking equivalence of Electrical Logic circuits to Artificial Intelligence and Machine Learning. For a more in- depth look into the applications look here.

……

While solving big NP-complete problems using Quantum Computers at the moment seems a bit far-fetched, the above example proves that we are on the right path and soon we will be able to optimize transport routes, better our industrial practices and much more.

References

1. **https://en.wikipedia.org/wiki/NP-completeness**
2. **https://physicsworld.com/a/quantum-algorithm-provides-new-approach-to-np-hard-problem/**
3. **https://medium.com/coinmonks/quantum-np-complete-no-problem-9fd5d5f8b2e0**
4. **https://qiskit.org/textbook/ch-applications/satisfiability-grover.html**
5. **https://qiskit.org/documentation/stable/0.24/tutorials/algorithms/08_grover_examples.html**
6.