1 Solving the Subset-Sum Problem with Grover's Algorithm

Grover's algorithm is a quantum search algorithm that provides a quadratic speedup over classical brute-force search methods for unstructured problems. The Subset-Sum problem is an NP-complete problem that can be framed as an unstructured search, making it a suitable candidate for Grover's algorithm. This report details the implementation of a quantum solution to the Subset-Sum problem for a small instance, as demonstrated in the provided code.

1.1 Problem Statement

The Subset-Sum problem is defined as follows: given a set of integers (weights) $W = \{w_1, w_2, \ldots, w_n\}$ and a target integer T, determine if there exists a subset of W whose elements sum exactly to T.

For this implementation, the specific problem instance is:

- **Weights**: $W = [1, 2, 4, 5]$ $(n = 4)$
- **Target Sum**: $T = 7$

Each possible subset can be represented by a binary string $x = x_1 x_2 \ldots x_n$, where $x_i = 1$ if w_i is included in the subset and $x_i = 0$ otherwise. The goal is to find a string x such that:

$$\sum_{i=1}^{n} x_i w_i = T$$

1.2 Algorithm Implementation

The quantum algorithm uses Grover's search to find the binary string(s) x that satisfy the sum condition. The main components are the state preparation, the oracle that marks the solution states, and the diffuser that amplifies their probability.

1.2.1 Classical Solution for Verification

Before running the quantum algorithm, a classical brute-force search is performed to identify the solutions. This serves two purposes:

1. **Verification**: It provides the expected outcome to verify the quantum results.
2. **Iteration Count**: It determines the number of solutions, M, needed to calculate the optimal number of Grover iterations, $k \approx \frac{\pi}{4}\sqrt{\frac{N}{M}}$, where $N = 2^n$.

For $W = [1, 2, 4, 5]$ and $T = 7$, the classical search finds two solutions:

- $2 + 5 = 7$, corresponding to the bitstring '0101'.
- $1 + 2 + 4 = 7$, corresponding to the bitstring '1110'.

Note: Qiskit's bitstring ordering is right-to-left. The code correctly reverses the strings to match this convention, resulting in '1010' and '0111' as the target solutions. With $N = 16$ and $M = 2$, the optimal number of iterations is calculated as $k = \lfloor \frac{\pi}{4}\sqrt{\frac{16}{2}} \rfloor = 2$.

1.2.2 Quantum State Preparation

The algorithm begins by preparing a uniform superposition of all possible subset combinations. This is achieved by applying a Hadamard (H) gate to each of the n variable qubits, which represent the binary string x:

$$|\psi\rangle = H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

An additional flag qubit is initialized to the $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$ state, which is necessary for the phase kickback mechanism used by the oracle.

1.2.3 Oracle U_ω

The oracle, U_ω, is the core of the algorithm. It identifies and "marks" the states corresponding to correct subsets by inverting their phase. This is achieved in several steps:

1. **Quantum Adder**: A quantum adder circuit is constructed to compute the sum $S = \sum x_i w_i$. The implementation uses a Quantum Fourier Transform (QFT) based adder. This adder works by first applying a QFT to an auxiliary 'sum' register. Then, for each variable qubit $|x_i\rangle$ that is in the $|1\rangle$ state, a series of controlled phase rotations are applied to the 'sum' register. The angle of each rotation depends on the corresponding weight w_i. Finally, an inverse QFT transforms the 'sum' register back, so it holds the binary representation of the sum S.

2. **Comparison**: The computed sum S in the 'sum' register is compared to the target T. This is done by applying X gates to the sum qubits where the corresponding bit in T's binary representation is '0'. After this, a multi-controlled X (MCX) gate flips the flag qubit if and only if all sum qubits are now in the $|1\rangle$ state, which only happens if $S = T$.

3. **Uncomputation**: To preserve the integrity of Grover's algorithm, all auxiliary operations must be reversed. The comparison and sum calculations are uncomputed by applying the inverse of their respective circuits. This restores the 'sum' register to the $|0\rangle$ state, leaving only the phase flip on the solution states.

1.2.4 Diffuser and Iteration

After the oracle marks the solution states, the diffuser (or amplitude amplification) operator is applied. The diffuser inverts the amplitudes of all states about their mean. This has the effect of increasing the amplitudes of the marked (negative phase) states and decreasing the amplitudes of the others. The oracle and diffuser are applied iteratively $k = 2$ times to maximize the probability of measuring a solution state.

1.3 Simulation and Results

The complete Grover circuit was simulated with 1024 shots. The simulation results show the probability distribution over all possible subsets.
The probabilities of measuring the two correct solutions are significantly amplified:

- `'1010'`: Probability ≈ 0.488

- `’0111’`: Probability ≈ 0.454

All other non-solution states have a combined probability of less than 6%. The bar chart in Figure 1 visually confirms this outcome, with two prominent peaks corresponding to the correct bitstrings.

1.4 Analysis and Conclusion

The results from the quantum simulation align perfectly with the classically computed solutions. The Grover’s algorithm implementation successfully identified the two subsets of $W = [1, 2, 4, 5]$ that sum to $T = 7$. The high probabilities associated with the bitstrings ‘’1010’‘ (for subset $\{2, 5\}$) and ‘’0111’‘ (for subset $\{1, 2, 4\}$) demonstrate the power of amplitude amplification in solving search problems.

This implementation successfully showcases how Grover’s algorithm can be applied to the Subset-Sum problem by designing a suitable quantum oracle that checks the sum condition. For this small-scale problem ($n = 4$), the algorithm correctly and efficiently found the solutions.

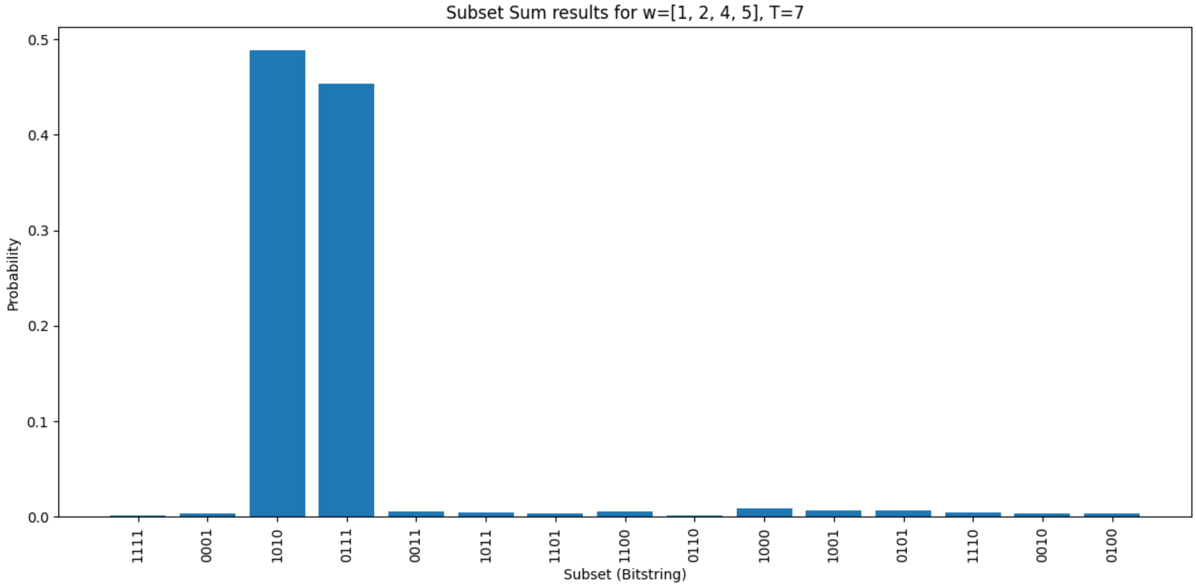

Figure 1: Probability distribution of measured outcomes from the simulation. The peaks at ‘’1010’‘ and ‘’0111’‘ correspond to the correct solutions.