A Quantum Algorithm for MAXSAT

Aman Chawla
Indian Institute of Technology Delhi

September 1, 2024

Abstract

In this report, we use a Large Language Model (LLM) called ChatGPT to create a new quantum algorithm for the satisfiability problem MAXSAT[1]. We then have the LLM generate a script that implements the algorithm and presents the results. Next we perform a quantum analysis and compute the measurement probability of the various outcomes of the algorithm.

Part I
Basic Formulation

To design a unitary operation V using Kempe's quantum random walk approach, with the goal of finding the target state that represents the maximal value or a value closest to the maximal value of a weighted sum of m known clauses, we can leverage the structure of quantum walks on a Hilbert space associated with the n-qubit system.

Problem Setup:

The key components of the problem setup are as follows.

Quantum State Space: The system has n qubits, leading to a computational basis of dimension 2^n. Each basis state $|x\rangle$ corresponds to a specific assignment of 0 or 1 to the n variables.

Clauses: There are m clauses, each contributing to a weighted sum based on the specific assignment of variables in $|x\rangle$. The goal is to find the state $|x*\rangle$ that either maximizes or is closest to maximizing this weighted sum.

[1]LLM and ML/AI use is becoming more prevalent in fields related to quantum physics [1].

Quantum Walk: We will design V based on a quantum walk that efficiently explores the computational basis, guiding the evolution towards the desired target state $\mid x*\rangle$. Next, we discuss our approach in detail.

1 Weighted Sum Objective Function

Define an objective function $f(x)$ that represents the weighted sum of the clauses for a given assignment x. The goal is to find the state $\mid x*\rangle$ where $f(x*)$ is maximal.

2 Hilbert Space and Oracle Function

Hilbert Space: The state space is C^{2^n} where each basis state $\mid x\rangle$ corresponds to a potential solution.

Oracle Function O_f: Define an oracle unitary O_f that marks the basis states based on the value of $f(x)$ [2]. Specifically, if the value of $f(x)$ is larger than the value of $f(x)$ in all previous steps, the state x is marked as red and previous marked red states are flipped to blue. For another example: $O_f \mid x\rangle = -1^{f(x)} \mid x\rangle$. This oracle flips the phase of the state if $f(x)$ is above a certain threshold, thereby marking promising states.

3 Quantum Random Walk on the Hypercube:

Consider a quantum walk on the n-dimensional hypercube, where each vertex corresponds to a state $|x\rangle$, and edges represent transitions between states that differ by one bit. Define a step operator S that applies a quantum walk on this hypercube, transitioning between adjacent vertices (states) [3].

The step operator could be designed as: $S = \sum_x \sum_{i=1}^{n} \mid x \bigoplus e_i\rangle \langle x \mid \otimes \mid i\rangle \langle i \mid$ where e_i is the basis vector corresponding to flipping the i-th bit.

4 Combining Walk with Oracle O_f :

The key idea is to combine the quantum walk operator S with the oracle O_f. The evolution of the quantum state is then driven by the repeated application of these operations: $W = O_f S$. This unitary W effectively "guides" the quantum walk by marking promising states (based on $f(x)$) and then spreading amplitude towards those states.

5 Reflection and Iterative Amplification:

Reflection about the Average: Introduce a reflection operation R that reflects the current state about the average state [4]. This is akin to Grover's diffusion operator: $R = 2|\varphi_0\rangle\langle\varphi_0| - 1$ where $|\varphi_0\rangle$ is the uniform superposition state [5].

Iterative Application: The final procedure involves iteratively applying the composite operator $V = RW$ several times. This amplifies the amplitude of the states that are marked by the oracle (i.e., states close to the maximum value of $f(x)$.

6 Measurement:

After sufficient iterations (the optimal number can be determined based on the problem size n), the final state is measured. The probability of obtaining the state $|x*\rangle$ that maximizes or is closest to maximizing $f(x)$ should be significantly amplified.

Summary:

1. Initialize: Start with the uniform superposition of all basis states. $|\varphi_0\rangle = \frac{1}{2^n}\sum_0^{2^{n-1}} |x\rangle$.

2. Quantum Walk + Oracle: Apply the composite operator $W = O_f\ S$ to guide the quantum walk towards states with higher objective values $f(x)$.

3. Amplitude Amplification: Use a reflection operator R to amplify the probability amplitude of promising states. The overall unitary $V = RW$ is applied iteratively.

4. Measurement: Measure the final state, which will most likely collapse to the state $|x*\rangle$ that maximizes $f(x)$ or is close to the maximum. This approach leverages Kempe's quantum random walk technique combined with amplitude amplification to efficiently find the target state $|x*\rangle$ in a high-dimensional search space.

Results

The clauses' weights were set randomly. The algorithm executed and provided the following output, shown in Figure 1. The counts obtained for each clause are clearly indicated. Based on the graph, the clauses "001" and "110" are highly probable to be the solution.

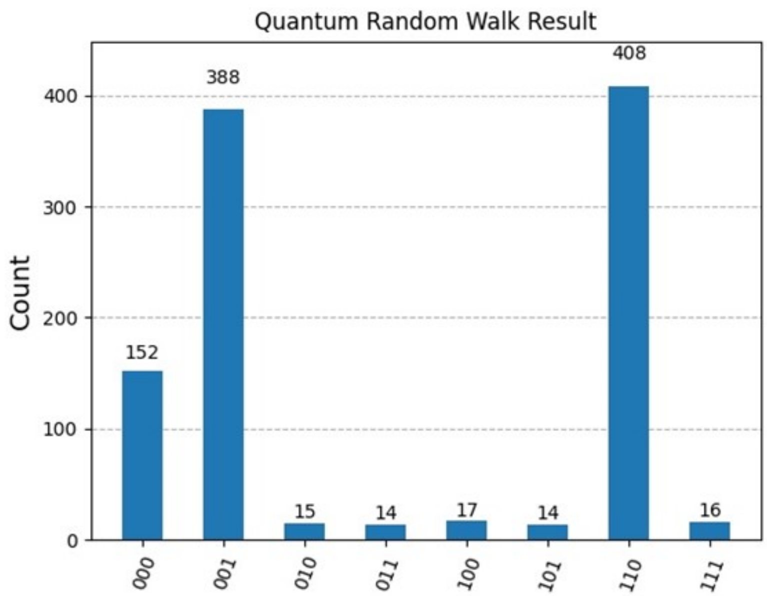

Figure 1: First run of the algorithm.

Next Steps

Theory: Our next step is to understand the step and reflection operators more deeply. Then we must perform an analysis of the probability of finding the maximization solution.

Algorithm: Our next step is to compare the performance of this approach to the standard QAOA algorithm, understanding clearly where the difference lies and how it impacts performance. We would also like to set the weights manually and not at random, in a systematic way.

Part II

Extended Oracle

To implement the described coloring mechanism in a quantum context, you can design a quantum circuit that uses an oracle O_f to compare the values of a function $f(x)$ at different inputs and mark the states accordingly. Here's how you can conceptualize and implement this:

Quantum Oracle O_f

Definition The oracle O_f is defined to compare the value of $f(x)$ with the maximum value of $f(x)$ seen so far. If $f(x)$ is larger, the oracle marks the corresponding quantum state $|x\rangle$ as "red" by flipping a specific ancilla qubit, and if it is smaller, the state is not marked. Additionally, if a state is marked as red, previous red-marked states are flipped to blue.

Steps to Implement the Coloring

1. Initialize the Quantum State: Start with a quantum register $|x\rangle$ representing the inputs, an ancillary register $|c\rangle$ to store color states (where $|0\rangle$ is uncolored, $|1\rangle$ is red, and $|2\rangle$ is blue), and a register $|m\rangle$ to track the maximum $f(x)$ seen so far.

$$|\psi\rangle = |x\rangle \otimes |c\rangle \otimes |m\rangle \tag{1}$$

2. Compute $f(x)$: Use a quantum circuit to compute $f(x)$ and store it in a temporary register $|f\rangle$.

$$|\psi\rangle \to |x\rangle \otimes |c\rangle \otimes |m\rangle \otimes |f(x)\rangle \tag{2}$$

3. Compare $f(x)$ with the Maximum Value: Apply a comparator quantum circuit to compare $f(x)$ with the value stored in the $|m\rangle$ register. If $f(x)$ is larger, set $|m\rangle$ to $f(x)$, and flip the color qubit in $|c\rangle$ to mark the state as red (value 1).

$$|\psi\rangle \to |x\rangle \otimes |1\rangle \otimes |f(x)\rangle \quad \text{if } f(x) > m \quad \text{(Mark as red)} \tag{3}$$

Otherwise, the state remains unmarked or as blue.

4. Unmark Previous Red States: To flip previously marked red states to blue (value 2), you need to iterate over the states. Implement a controlled operation that checks if a state was previously marked as red and flips it to blue if the current state is marked red.

$$|\psi\rangle \to |x\rangle \otimes |2\rangle \otimes |f(x)\rangle \quad \text{(Flip previous red to blue)} \tag{4}$$

5. Reset Ancilla Registers: After marking the states, reset the ancilla registers $|f\rangle$ and $|m\rangle$ to their initial states, if necessary, to ensure they don't interfere with subsequent operations.

Quantum Circuit Implementation

The circuit implementing this coloring procedure consists of:

Function Evaluation:

A subcircuit that evaluates $f(x)$.

Comparator:

A comparator subcircuit that compares $f(x)$ with the stored maximum value.

Coloring:

Controlled operations to mark states as red or flip them to blue.

Reset Operations:

If necessary, reset operations for ancilla registers.

Conclusion

This process leverages quantum parallelism to efficiently mark and unmark quantum states based on the comparison of function values. The key operations are the quantum oracle O_f, the comparator, and the controlled operations for flipping colors. This approach ensures that at each step, the state corresponding to the largest $f(x)$ so far is marked as red, and all previous red-marked states are flipped to blue.

Part III

Alternate Approach to Oracle Design Using ChatGPT

To implement the described oracle unitary O_f that colors the basis states based on the function $f(x)$, we can define a quantum operation that marks states as red or blue according to the specified criteria. Here's a step-by-step method to achieve this:

Step 1: State Representation

Assume we have a quantum state represented as a superposition of basis states:

$$|\psi\rangle = \sum_x \alpha_x |x\rangle \tag{5}$$

where $|x\rangle$ are the basis states and α_x are the corresponding amplitudes.

Step 2: Auxiliary Register

Introduce an auxiliary qubit $|c\rangle$ to represent the color of the state. This qubit will be initialized to $|0\rangle$ (representing "no color" or "blue").

Step 3: Define the Oracle Unitary O_f

The oracle unitary O_f needs to:

1. Compare the current $f(x)$ with the maximum value encountered in previous iterations.
2. Mark the current state $|x\rangle$ as red if $f(x)$ is larger than any previously marked state, and flip previous marked states from red to blue.

Step 4: Quantum Operation for Coloring

1. Initialization: Set $|c\rangle = |0\rangle$.

2. Oracle Operation: The operation O_f will be defined as follows:

- For each basis state $|x\rangle$, if $f(x)$ is greater than the maximum value seen so far, perform the following:
- Flip the auxiliary qubit $|c\rangle$ from $|0\rangle$ (blue) to $|1\rangle$ (red).
- Store the maximum value $f(x)$ for comparison in subsequent calls.

3. Implementation: The operation can be summarized as:

$$O_f|x\rangle|c\rangle = \begin{cases} |x\rangle|1\rangle & \text{if } f(x) > \max_f \\ |x\rangle|0\rangle & \text{otherwise} \end{cases} \tag{6}$$

Here, we would also need a mechanism to update $\max_f$ (the maximum value seen so far), which can be done using a classical computation or an additional quantum register to keep track of the maximum.

Step 5: Post-Oracle State

After applying the oracle, the state of the system would change, and the auxiliary qubit will reflect the current state's color.

Step 6: Measuring the Auxiliary Qubit

To observe the result of the coloring operation, measure the auxiliary qubit. The result will indicate whether the last state marked was red or blue:

- If measured $|1\rangle$: the state $|x\rangle$ is marked red.
- If measured $|0\rangle$: the state $|x\rangle$ is marked blue.

Summary

The oracle O_f can be effectively designed to use a quantum superposition to compare values of $f(x)$, updating an auxiliary state based on the criteria provided. This coloring process can be integrated into a larger quantum algorithm that leverages the oracle for optimization or search problems, such as in the context of Grover's algorithm or other quantum search techniques.

Note

The above construction assumes access to the function $f(x)$ in a way compatible with quantum operations (e.g., a quantum circuit or a classical query). The auxiliary qubit helps track the color state and can be manipulated based on the function evaluations during the oracle's execution.

Part IV
Quantum Analysis

In this part, we will undertake a full quantum mechanical analysis of the algorithm presented in Part I. We begin our analysis with a starting state which is a "cat" state. The starting state is specified as

$$|\phi_0\rangle = \frac{1}{2^n} \sum_{x=0}^{2^{n-1}} |x\rangle. \tag{7}$$

Reconsider Schrodinger's famous cat thought experiment in which the cat is equally probable to be dead or alive since one doesn't know the state of the executioner's weapon.

7 Step Operator

Next, we consider the random walk step operator from Kempe's work (Equation (12) in [6]). The operator may be written as

$$S = \sum_x \sum_{i=1}^{n} |x \oplus e_i\rangle\langle x| \otimes |i\rangle\langle i| \tag{8}$$

where e_i is a quantity that is linked with the ket $|i\rangle$. For example if it is a spin-half qubit space so $n = 2$, then $|i\rangle$ may stand for spin up when $i = 1$ and spin down when $i = 2$. And, relatedly, e_1 could be a positive 1 and e_2 a negative 1. The first step is to figure out the application of the step operator on the $|\phi_0\rangle$ state. We note that

$$S|\phi_0\rangle = \sum_x \sum_{i=1}^{n} |x \oplus e_i\rangle\langle x| \otimes |i\rangle\langle i| \left(\frac{1}{2^n} \sum_{x=0}^{2^{n-1}} |x\rangle \right) \tag{9}$$

by simply plugging in. We change the second label x to a y in order to avoid confusing the two enumerations. We have,

$$S|\phi_0\rangle = \sum_x \sum_{i=1}^{n} |x \oplus e_i\rangle\langle x| \otimes |i\rangle\langle i| \left(\frac{1}{2^n} \sum_{y=0}^{2^{n-1}} |y\rangle \right). \tag{10}$$

As the next step we explicitly write out (expand out) the basis kets $|i\rangle$. We have,

$$S|\phi_0\rangle = \left[\sum_x |x \oplus e_1\rangle\langle x| \otimes |1\rangle\langle 1| + \sum_x |x \oplus e_2\rangle\langle x| \otimes |2\rangle\langle 2| + \ldots + \sum_x |x \oplus e_n\rangle\langle x| \otimes |n\rangle\langle n| \right] \left(\frac{1}{2^n} \sum_{y=0}^{2^{n-1}} |y\rangle \right). \tag{11}$$

Here the x values index the positions on the line if it is a one dimensional random walk. The initial state is a quantum superposition of all the line indices. Thus x can follow the same pattern as y. The direct sum terms e_i represent actions that transition the ket $|y\rangle$ to the ket $|y \oplus e_i\rangle$. This relies on the orthonormality of the $|y\rangle$ kets. It is therefore more illustrative if we rewrite the expansion (Equation (11)) as follows, by switching the order of the tensor-ed operators. We have,

$$S|\phi_0\rangle = \left[|1\rangle\langle 1| \otimes \sum_x |x \oplus e_1\rangle\langle x| + |2\rangle\langle 2| \otimes \sum_x |x \oplus e_2\rangle\langle x| + \ldots + |n\rangle\langle n| \otimes \sum_x |x \oplus e_n\rangle\langle x| \right] \left(\frac{1}{2^n} \sum_{y=0}^{2^{n-1}} |y\rangle \right). \tag{12}$$

Next we move the cat state into the operators on the left. We are precluded from doing this however, since the cat state is a state in the Hilbert space of the line. However, the operator being applied to it is an operator in the tensor product Hilbert space of the ancilla system and the line. So we must augment the cat state with an ancilla. The ancilla can be in the Hilbert space which has the number kets $|1\rangle$ through $|n\rangle$ as its basis. Thus we rewrite the cat state we are operating on as

$$|k\rangle \otimes |\phi_0\rangle \tag{13}$$

for some specific number state labeled by "k." We operate on this state by the step operator S. We obtain,

$$= \left[|1\rangle\langle 1| \otimes \sum_x |x \oplus e_1\rangle\langle x| + |2\rangle\langle 2| \otimes \sum_x |x \oplus e_2\rangle\langle x| + \ldots + |n\rangle\langle n| \otimes \sum_x |x \oplus e_n\rangle\langle x| \right] [|k\rangle \otimes |\phi_0\rangle]. \tag{14}$$

We move each subsystem to its appropriate operator,

$$= \left[|1\rangle\langle 1|k\rangle \otimes \sum_x |x \oplus e_1\rangle\langle x|\phi_0\rangle + |2\rangle\langle 2|k\rangle \otimes \sum_x |x \oplus e_2\rangle\langle x|\phi_0\rangle + \ldots + |n\rangle\langle n|k\rangle \otimes \sum_x |x \oplus e_n\rangle\langle x|\phi_0\rangle \right]. \tag{15}$$

This simplifies since $\langle j|k\rangle = \delta_{jk}$ (the Kronecker delta) and $\langle x|\phi_0\rangle = \frac{1}{2^n}\sum_{y=0}^{2^n-1}\langle x|y\rangle = \frac{1}{2^n}\sum_{y=0}^{2^n-1}\delta_{xy}$ where δ_{xy} is 1 unless x and y labels are distinct, in which case it is 0. We have $\langle x|\phi_0\rangle = \frac{1}{2^n}$. So the simplified expression is,

$$S\left(|k\rangle \otimes |\phi_0\rangle\right) = \left[|k\rangle \otimes \frac{1}{2^n}\sum_x |x \oplus e_k\rangle \right]. \tag{16}$$

In other words, the random walk moves the cat state to a new superposition state. Strictly speaking, to characterize the action of the operator S, we must show its operation on the basis states $\{|k\rangle \otimes |0\ldots 0\rangle, |k\rangle \otimes |0\ldots 1\rangle, |k\rangle \otimes |0\ldots 10\rangle, \ldots\}$. The binary representations inside the line position kets represent the position number in binary. Here e_k flips the k-th binary digit via the direct sum. This means that the final states after the action of S will be $\{|k\rangle \otimes |0\ldots 0 \oplus e_k\rangle, |k\rangle \otimes |0\ldots 1 \oplus e_k\rangle, |k\rangle \otimes |0\ldots 10 \oplus e_k$ Just to reiterate, the n bits represent 2^n line positions and e_k flips the k-th bit. Since there are only n bits, k runs from 1 through n.

In the next sections we will analyze the subsequent application of the reflection and oracle operators.

8 Oracle Operator

We may begin our investigation with the oracle defined in Section 2. There we said,

$$O_f|x\rangle = (-1)^{f(x)}|x\rangle. \tag{17}$$

So we fix a function $f(\cdot)$ which assigns an integer value to every n-bit binary number (state label) that is input to it. In practice, this function will be tailored to the MAXSAT objective (see Section 1). Then, the oracle operator can be characterized by its action on every n-bit line position ket. In general,

$$O_f|0\ldots 0\rangle = (-1)^{f(0)}|0\ldots 0\rangle, \tag{18}$$

$$O_f|0\ldots 01\rangle = (-1)^{f(1)}|0\ldots 01\rangle \tag{19}$$

and so on. Consider the basis state $|k\rangle \otimes |0\ldots 0\rangle$. The step operator transforms it into $|k\rangle \otimes |0\ldots 0 \oplus e_k\rangle$. Since the oracle acts only on the line kets, we can augment it by the identity map I_n on the number kets $|k\rangle$. Thus the overall oracle becomes $I_n \otimes O_f$. The final state after the action of the step and then the oracle therefore is

$$\left(I_n \otimes O_f\right)\left(|k\rangle \otimes |0\ldots 0 \oplus e_k\rangle\right) = |k\rangle \otimes (-1)^{f(0\ldots 0\oplus e_k)}|0\ldots 0 \oplus e_k\rangle. \tag{20}$$

9 Reflection Operator

Consider the reflection operator defined as

$$R = 2|\ \varphi_0\rangle\ \langle\varphi_0|\ - I_n. \tag{21}$$

Consider a generic state $|\psi\rangle$ which we can decompose into a part along $|\varphi_0\rangle$ and a part orthogonal to $|\varphi_0\rangle$. We write,

$$|\psi\rangle = \left(|\varphi_0\rangle\langle\varphi_0| + |\varphi_0^\perp\rangle\langle\varphi_0^\perp|\right)|\psi\rangle = \langle\varphi_0|\psi\rangle|\varphi_0\rangle + \langle\varphi_0^\perp|\psi\rangle|\varphi_0^\perp\rangle \tag{22}$$

which can be operated upon by R to obtain,

$$R|\psi\rangle = [2|\varphi_0\rangle\langle\varphi_0| - I]\left(|\varphi_0\rangle\langle\varphi_0| + |\varphi_0^\perp\rangle\langle\varphi_0^\perp|\right)|\psi\rangle \tag{23}$$

which simplifies to

$$R|\psi\rangle = 2|\varphi_0\rangle\langle\varphi_0|\psi\rangle - |\varphi_0\rangle\langle\varphi_0|\psi\rangle - |\varphi_0^\perp\rangle\langle\varphi_0^\perp|\psi\rangle \tag{24}$$

which in turn simplifies to

$$R|\psi\rangle = \langle\varphi_0|\psi\rangle|\varphi_0\rangle - \langle\varphi_0^\perp|\psi\rangle|\varphi_0^\perp\rangle \tag{25}$$

which should be compared with Equation (22). It is immediate that R has reflected the input state about the cat state $|\varphi_0\rangle$.

We take the final state after the action of the random walk step operator and oracle operator and pass it through the reflection operator. Omitting the intermediate steps which are straightforward in the interest of brevity, we may write down the final state as,

$$|k\rangle \otimes \left[(-1)^{f(0\ldots0\oplus e_k)}\left(2^{1-n} - 1\right)|0\ldots0 \oplus e_k\rangle\right] \tag{26}$$

10 POVM

10.1 Single Round

As the final operation, we must measure the final state after the first round. Consider a projection measurement as a first step. The state to be measured resides in a state space of size 2^n. So we want at least 2^n measurement outcomes which will tell us the basis-subspace to which the state to be measured belongs and with what probability. The probability will be in terms of the function $f(\cdot)$.

We write down the measurement operators as the projectors onto the basis states. Thus M_0 is the projector onto $|0\ldots0\rangle$, M_1 is the projector onto $|0\ldots01\rangle$, and $M_{2^{n-1}}$ is the projector onto $|1\ldots1\rangle$. So we can construct the operators as follows:

$$M_0 = |0\ldots0\rangle\langle0\ldots0|, \tag{27}$$

$$M_1 = |0\ldots1\rangle\langle0\ldots1| \tag{28}$$

and so on

$$M_{2^{n-1}} = |1\dots1\rangle\langle1\dots1|. \tag{29}$$

Assuming the final state is given by Equation (26) with $k = 1$, the probability of getting the result m is

$$p(m) = (-1)^{f(0\dots0\oplus e_1)} \left(2^{1-n} - 1\right) \langle 0\dots0\oplus e_1| M_m (-1)^{f(0\dots0\oplus e_1)} \left(2^{1-n} - 1\right) |0\dots0\oplus e_1\rangle. \tag{30}$$

Next we collect the coefficient factors up front to get,

$$p(m) = \left[(-1)^{f(0\dots0\oplus e_1)} \left(2^{1-n} - 1\right)\right]^2 \langle 0\dots0\oplus e_1|m\rangle\langle m|0\dots0\oplus e_1\rangle \tag{31}$$

which can be simplified to

$$p(m) = \left[(-1)^{f(0\dots1)} \left(2^{1-n} - 1\right)\right]^2 \langle 0\dots1|m\rangle\langle m|0\dots1\rangle. \tag{32}$$

Clearly, $p(m)$ is zero unless m is 1. In general the input state will not be given by Equation (26) but some superposition

$$\sum_{j=0}^{2^{n-1}} |k\rangle \otimes \left[(-1)^{f(j\oplus e_k)} \left(2^{1-n} - 1\right) |j\oplus e_k\rangle\right]. \tag{33}$$

Then we find that

$$p(m) = \left[(-1)^{f(j'\oplus e_k)} \left(2^{1-n} - 1\right)\right]^2 \langle j'\oplus e_k|m\rangle\langle m|j'\oplus e_k\rangle \tag{34}$$

where j' is the j such that

$$j' \oplus e_k = m \tag{35}$$

so that

$$p(m) = \left[(-1)^{f(m)} \left(2^{1-n} - 1\right)\right]^2. \tag{36}$$

Recall that here n is the dimension of the ancilla system. We must ensure normalization of the result above.

10.2 Two Rounds

We start with the state

$$|k\rangle \otimes \left[(-1)^{f(0\dots0\oplus e_k)} \left(2^{1-n} - 1\right) |0\dots0\oplus e_k\rangle\right] \tag{37}$$

instead of the cat state and subject it to the step operator. We obtain,

$$|k\rangle \otimes \Big[(-1)^{f(0\ldots0\oplus e_k\oplus e_k)}\left(2^{1-n}-1\right)|0\ldots0\oplus e_k\oplus e_k\rangle\Big]\,. \tag{38}$$

Since flipping the same bit two times yields no change to the bit, we have,

$$|k\rangle \otimes \Big[(-1)^{f(0\ldots0)}\left(2^{1-n}-1\right)|0\ldots0\rangle\Big]\,. \tag{39}$$

Next we subject this last result to the oracle and obtain

$$|k\rangle \otimes \Big[(-1)^{2\cdot f(0\ldots0)}\left(2^{1-n}-1\right)|0\ldots0\rangle\Big]\,. \tag{40}$$

The last operator to be applied is the reflection and we obtain,

$$|k\rangle\otimes\Big[2\cdot(-1)^{2\cdot f(0\ldots0)}\left(2^{1-n}-1\right)|\phi_0\rangle\langle\phi_0|0\ldots0\rangle-(-1)^{2\cdot f(0\ldots0)}\left(2^{1-n}-1\right)|0\ldots0\rangle\Big] \tag{41}$$

which simplifies to

$$|k\rangle\otimes\Big[2^{(1-n)}\cdot(-1)^{2\cdot f(0\ldots0)}\left(2^{1-n}-1\right)|0\ldots0\rangle-(-1)^{2\cdot f(0\ldots0)}\left(2^{1-n}-1\right)|0\ldots0\rangle\Big] \tag{42}$$

and finally collecting terms,

$$|k\rangle \otimes \Big[(2^{(1-n)}-1)\cdot(-1)^{2\cdot f(0\ldots0)}\left(2^{1-n}-1\right)|0\ldots0\rangle\Big]\,. \tag{43}$$

Finally, we just need to measure this state. The post-measurement probability can be written down by inspection as,

$$p(m)=\Big[(-1)^{f(m)}\left(2^{1-n}-1\right)\Big]^4 \tag{44}$$

10.3 Multiple Rounds (Even Number)

If we apply an even number r of rounds to the cat state, we can make some observations:

1. Since the number of rounds is even, an even number of e_k will be direct summed with the zero ket, leaving it unchanged.

2. Since the oracle will have been applied r times, there will be a coefficient of r instead of the 2 that appears in Equation (40).

3. Since the ket will be the zero ket, once the R operator is applied, which is based on the cat state, only the zero ket of the cat state will survive along with a coefficient of $\frac{1}{2^n}$.

Finally, therefore we will obtain a state of the form,

$$|k\rangle \otimes \Big[(2^{(1-n)}-1)^r\cdot(-1)^{r\cdot f(0\ldots0)}|0\ldots0\rangle\Big] \tag{45}$$

which would then be subject to measurement, to obtain

$$p(m)=\Big[(-1)^{f(m)}\left(2^{1-n}-1\right)\Big]^{2r} \tag{46}$$

10.4 Multiple Rounds (Odd Number)

This would be a slightly more complicated case than the previous ones. We just need to observe that an odd number of flips of the same bit is equal to a single flip of the bit. Thus the final state before measurement can be written as,

$$|k\rangle \otimes \left[(2^{(1-n)} - 1)^r \cdot (-1)^{r \cdot f(0\ldots01)} |0\ldots01\rangle \right]. \tag{47}$$

When the measurement is performed, we obtain,

$$p(m) = \left[(-1)^{f(m)} \left(2^{1-n} - 1\right) \right]^{2r}. \tag{48}$$

This can then be treated as the general expression.

Acknowledgment

The author would like to thank Prof. Kolin Paul, CSE, IIT Delhi, for suggesting this problem.

References

[1] Mario Krenn, Manuel Erhard, and Anton Zeilinger. Computer-inspired quantum experiments. *Nature Reviews Physics*, 2(11):649–661, 2020.

[2] Anthony Chefles. Distributed implementation of standard oracle operators. *Physical Review A Atomic, Molecular, and Optical Physics*, 78(6):062304, 2008.

[3] Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via quantum walk. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 575–584, 2007.

[4] Anirban Narayan Chowdhury, Yigit Subasi, and Rolando D Somma. Improved implementation of reflection operators. *arXiv preprint arXiv:1803.02466*, 2018.

[5] Avatar Tulsi. Faster quantum searching with almost any diffusion operator. *Physical Review A*, 91(5):052307, 2015.

[6] Julia Kempe. Quantum random walks: an introductory overview. *Contemporary Physics*, 44(4):307–327, 2003.