Quantum Computing Notes
Prof. Hamzeh Roumani
EECS, York University, Toronto, Ont., Canada

THE GROVER ALGORITHM

1. The Problem

Consider a function *f:* $\{0,1\}^n$ --> {0,1}. Its domain has 2^n elements and its co-domain has 2. You can think of *f* as a 0/1-valued function of n 0/1 variables. In programming terms, *f* is a boolean function that takes n boolean parameters. And if you treat the n arguments of *f* as bits in the binary representation of some integer x, then *f* can be thought of as a function that maps an integer in [0,N-1] to either 0 or 1, where $N=2^n$. We assume that *f* is provided as a black-box U_f (an *oracle*) that implements it in hardware.

Given (the promise) that *f* maps all elements to 0 except for one which gets mapped to 1, i.e.:

$$\forall x \in [0, N-1]: f(x) = \begin{cases} 0 \; if \; x \neq t \\ 1 \; if \; x = t \end{cases}$$

find t, the *target* (find its n bits). This is an example of search in unstructured data because we don't have access to the function's formula or algorithm–all we can do is query the oracle, by sending an x value, and examine its response. It is like searching in un-sorted arrays (a wrong guess does not lead to a better path toward the target).

2. A Classical Algorithm

Since we have no direct access to *f* (its formula or its circuit), all we can do is send arguments to the oracle and examine its returns. To find the target t, we must query the oracle N-1 times, in the worst case, before we stumble on t. Hence, this problem has a $O(2^n)$ query complexity. In fact, one can argue that the complexity of any classical algorithm must be $\Omega(2^n)$.

What if we settle for a *randomized* classical algorithm; one whose conclusion is mostly–but not always–correct? For example, we can pick a sample and hope to find t in it, which will bind the complexity by the sample size. Even such an approach cannot escape the exponential growth.

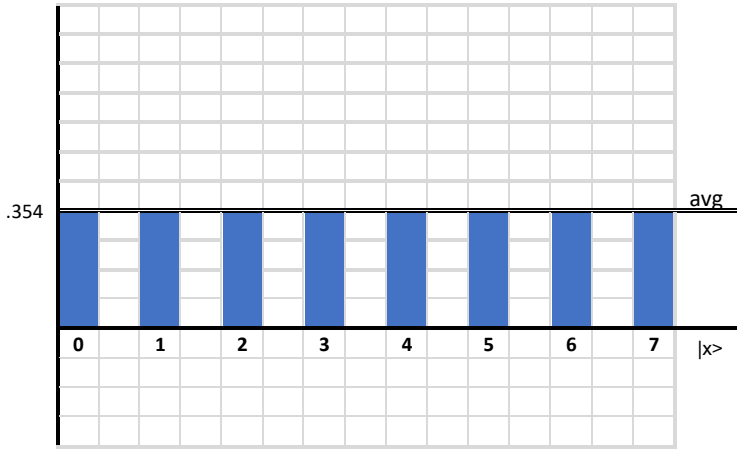
- Some problems (like Sudoku) are hard to solve but easy to verify. Is Grover's like that?
- Argue that the query complexity of any deterministic classical algorithm is $\Omega(2^n)$.
- Show that the sample size in the proposed randomized algorithm has to be ~N/2 for the success probability to reach ½.

3. The Idea

- Prepare an equal superposition of all values of x in [0,N-1], i.e.

  $(1/\sqrt{N})\ \Sigma|x\rangle$

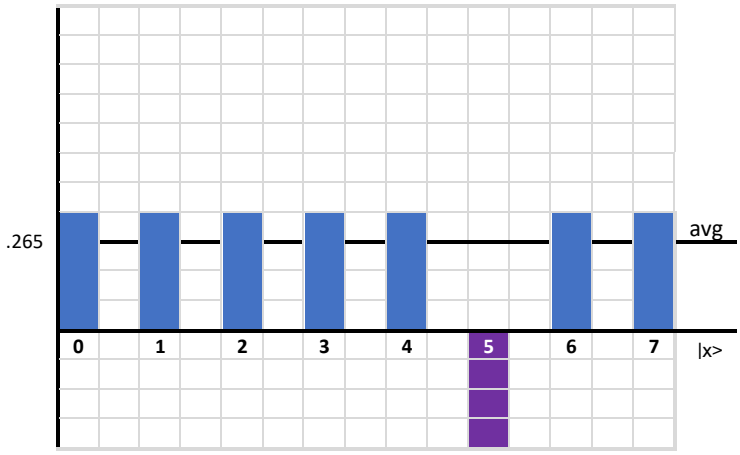
  In this state, the amplitudes of all terms are equal (being $1/\sqrt{N}$) so a measurement is equally likely to yield any of them. In the figure, N=8 so the average of the amplitudes is: $[\ 8 \times 1/\sqrt{8}\ ] / 8 \approx 0.354$



- Next, **flip** the amplitude of the term $|t\rangle$. This is not possible in our world without evaluating *f* at all x values, but it *is* possible (and rather easy) in the quantum world as we shall see in the next section.

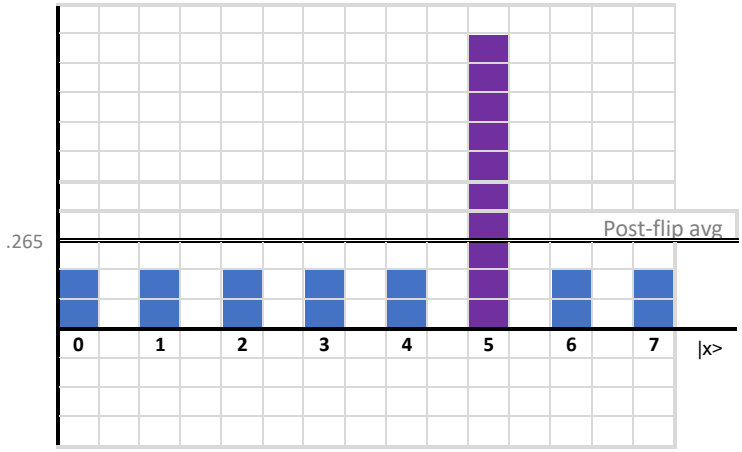
  In the figure, we assumed that t=5. The post-flip amplitude average is now:

  $[\ 7 \times (1/\sqrt{8}) - 1 \times (1/\sqrt{8})\ ] / 8 \approx 0.265$



- Finally, **invert** all amplitudes around the post-flip average, i.e. replace an amplitude 'a' with '2 x avg – a'.

  The x≠t amplitudes were all 0.089 above average so they drop by that much below it to 0.176. As to the t amplitude, it jumps from –0.354 to +0.884. Hence, if we measure now, the outcome will be the sought t with high probability!

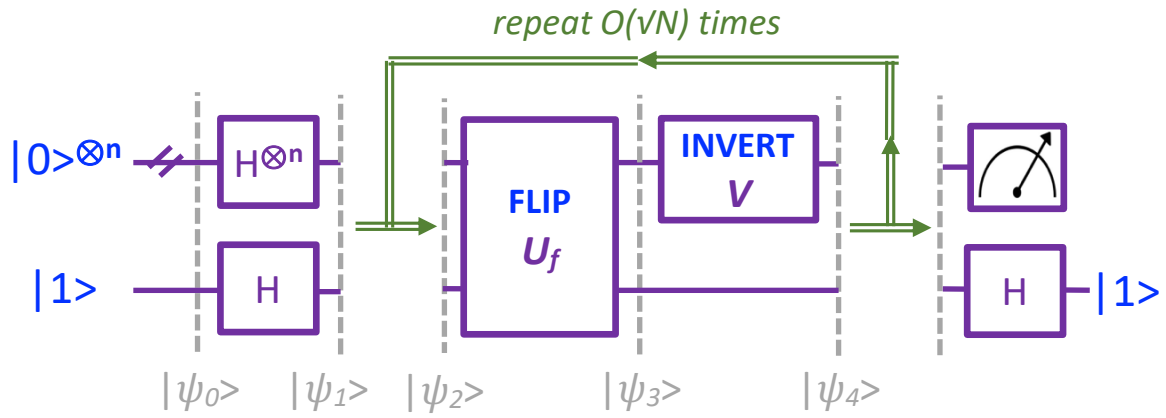


Hence, the **Flip-Invert** technique amplifies the amplitude of the x we are searching for, thereby increasing the probability of measuring it. In the above, the increase was from 12% to 78% (the probability is the square of the amplitude). In fact, the Flip-Invert technique can be iterated in order to increase the probability even further.

- In the above n=3 example, show that the average after the Invert is still 0.265.
- Prove that the post-Invert average is always equal to the post-Flip average.
- In the above n=3 example, show that repeating the Flip-Invert technique one more time (for a total of 2 iterations) would boost the probability of measuring t to about 95%.
- In the above n=3 example, show that repeating the Flip-Invert technique two more time (for a total of 3 iterations) would *lower* the probability of measuring t to about 33%.
- Work out the general case (not just n=3) and show that the amplitude of |t> increases from 1/√N to (3 - 4/N)/√N after one Flip-Invert iteration.

4. The Quantum Algorithm

The circuit diagram starts the same as in the Deutsch problem:



$$|\psi_0> = |0> \otimes |0> \otimes |0> \otimes \ldots \otimes |0> \otimes |1>$$

$$|\psi_1> = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k> \otimes |-> = \frac{1}{\sqrt{N}} \left(|t> + \sum_{k \neq t} |k> \right) \otimes |->$$

The Flip-Invert block that follows is to be repeated. We can think of this logically as looping the output of *V* back to the input of *Uf* (but this is not what physically happens--see the remarks at the end). Hence:

$$|\psi_2> = \begin{cases} |\psi_1>, & in\ the\ first\ iteration \\ |\psi_4>, & in\ subsequent\ iterations \end{cases}$$

After a number of iterations, we stop repeating and pass $|\psi_4>$ to the final stage. In it, the lower register (consisting of 1 qubit) gets uncomputed via a Hadamard gate, and the upper (consisting of n qubits) is measured to yield an outcome equal to |t> with high probability.

Note that the upper register of $|\psi_1>$ consists of an equal superposition of all x values in [0,N-1] with all amplitudes being 1/√N. But since the Flip-Invert block seeks to amplify the amplitude of |t> and suppress the remaining (N-1) amplitudes equally, let us rewrite $|\psi_1>$ in a way that accommodates unequal amplitudes so it would be valid in all iterations, not just the first:

$$|\psi_1> = \left(\alpha|t> + \beta \sum_{k \neq t} |k>\right) \otimes |->, \text{ where } \alpha^2 + (N-1)\beta^2 = 1$$

It should be noted that both α and β are real numbers because they start of being real, and FLIP and INVERT merely change their signs and magnitudes; they don't introduce phases. This is why the normalization condition was written for real, not complex, numbers (α^2 rather than $|\alpha|^2$). It should also be noted that there is only one β for all the N-1 non-target kets. This is because the non-target amplitudes start off being equal and INVERT amplify / suppress them equally.

Based on this, we can now write $|\psi_2>$ in a way that is valid in all iterations:

$$|\psi_2> = \left(\alpha|t> + \beta \sum_{k \neq t} |k>\right) \otimes |->$$

When this is fed to *Uf*, we know (from the Deutsch problem) that the two registers will get de-entangled by kicking the value of *f(x)* from a ket |*f(x)*> in the lower register to a phase $(-1)^{f(x)}$ in the upper. Hence:

$$|\psi_3> = \left(\alpha(-1)^{f(t)}|t> + \beta \sum_{k \neq t} (-1)^{f(k)}|k>\right) \otimes |->$$

And since *f*(t)=1 and *f*(k)=0:

$$|\psi_3> = \left(-\alpha|t> + \beta \sum_{k \neq t} |k>\right) \otimes |->$$

At this stage, the post-flip average '*avg*' of all N amplitudes is:

$$avg = \frac{-\alpha + (N-1)\beta}{N}$$

Upon feeding $|\psi_3>$ to *V*, all amplitudes would be inverted about this '*avg*':

$$|\psi_4> = \left((2 \times avg + \alpha)|t> + (2 \times avg - \beta) \sum_{k \neq t} |k>\right) \otimes |->$$

Comparing $|\psi_4\rangle$ to $|\psi_2\rangle$ shows how the amplitudes change from iteration *i* and iteration *i+1*:

$$\alpha_{i+1} = 2 \times avg + \alpha_i$$

$$\beta_{i+1} = 2 \times avg - \beta_i$$

These recurrence relations capture the essence of the Flip-Invert block. Solving them allows us to determine how many iterations are needed to boost the amplitude of t and thus increase the probability of measuring it.

Let us focus on the α recurrence. Replace '*avg*' with its value above and use the normalization condition to eliminate β:

$$\alpha_{i+1} = (1 - \frac{2}{N})\alpha_i + \frac{2\sqrt{N-1}}{N}\sqrt{1-\alpha_i^2}$$

To solve this recurrence, we note that the two N-dependent factors add up to 1 when squared, which suggests a sin/cos trigonometric substitution. Indeed, define the angle φ as follows:

$$cos(\varphi) = (1 - \frac{2}{N}) \Rightarrow sin(\varphi) = \frac{2\sqrt{N-1}}{N}$$

The recurrence becomes:

$$\alpha_{i+1} = cos(\varphi)\alpha_i + sin(\varphi)\sqrt{1-\alpha_i^2}$$

The presence of both α and $\sqrt{(1-\alpha^2)}$ together with the trigonometric identity for *sin(a+b)* give us a strong hint that the solution is:

$$\alpha_i = sin(i\varphi + \delta)$$

where δ is determined by the initial condition $\alpha=1/\sqrt{N}$ in iteration $i=0$:

$$\alpha_0 = sin(\delta) = 1/\sqrt{N} \Rightarrow \delta = \varphi/2$$

Hence, the solution of our recurrence is:

$$\alpha_i = sin(i\varphi + \varphi/2)$$

We conclude from this that α is a periodic function of the iteration count, and hence, it doesn't always increase! It is therefore important that we don't iterate too many times. We know that sin reaches its maximum first when the angle is π/2, at which point α becomes 1, the highest possible amplitude. Using this, and the fact that $\varphi \approx sin(\varphi) \approx 2/\sqrt{N}$ when N is large, we find:

$$i\varphi + \varphi/2 = \pi/2 \Rightarrow i \approx \pi/2\varphi \approx \pi/2sin(\varphi) \approx (\pi/4)\sqrt{N}$$

This tells us that the query complexity of Grover algorithm is $O(\sqrt{N}) = O(2^{n/2})$; a quadratic speed-up over the classical algorithm.

- Verify that sin(iφ + δ) is indeed the solution of the α recurrence.
- Prove that $\delta = \varphi / 2$.
- Solve the β recurrence along the lines done above for α. Verify that the amplitudes of the non-target kets approach zero when the target amplitude approaches 1.

5. Implementing the INVERT Gate

Building a unitary gate V that takes an arbitrary superposition and inverts its terms about the average of its amplitudes sounds complex because the transformation:

$$V \times \sum_{k=0}^{N-1} a_k|k> \; = \sum_{k=0}^{N-1} (2 \times avg \; - \; a_k)|k>$$

requires computing the average:

$$avg = \sum_{k=0}^{N-1} a_k/N$$

which seems difficult. But in fact, computing the average is "natural" when a matrix of 1's multiplies a column vector. Here is an example for n=3:

$$(1/3)\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = (1/3)\begin{pmatrix} a_0 + a_1 + a_2 \\ a_0 + a_1 + a_2 \\ a_0 + a_1 + a_2 \end{pmatrix} = \begin{pmatrix} avg \\ avg \\ avg \end{pmatrix}$$

We can therefore build V easily using an nxn matrix of 1's, M, and the identity matrix:

$$V = (2/N)M - I_n$$

It is easy to verify that V is unitary. In fact, it can be easily constructed from a π phase shift gate acting on |0>. See the exercises.

- Write a program in your favorite programming language to compute and print out the components of $|\psi_4>$ after each iteration in the n=3 case. Do this for 10 iterations and observe the periodicity of the target and non-target amplitudes.
- Show that for n=1 (i.e. N=2) that V=–HPH where H is the 2x2 Hadamard gate and P is a π phase shift gate that flips the phase of the |0> component.
- As above but for n=2. Note that H (aka Walsh-Hadamard) is given by: $H_{i,j} = (-1^{)i.j} / \sqrt{2^n}$.

Appendix: A Geometric Interpretation

We start the algorithm by preparing qubits in a superposition that has only two amplitudes: one for the target ket and one for the rest, and both start as real numbers equal to $1/\sqrt{N}$. The Flip-Invert block involves operations that modify these amplitudes, but it keeps them real. Moreover, these operations treat all the non-target kets the same, and hence, there are only two distinct amplitudes. Because of this, the system always remains in a two-dimensional, real Hilbert subspace (a hyperplane) with two axes, one for all the non-target kets, which we will treat as the horizontal x-axis, and one for the target ket, our vertical y-axis.

Viewing the algorithm within this subspace, we see that the state starts as a unit vector in the first quadrant with an angle $\varphi/2$ with the x-axis (because $\sin(\varphi/2) = 1/\sqrt{N}$). Let's call this vector the *equal* vector $|e\rangle$ because in it, all amplitudes are equal. We note that:

- The flip stage involves flipping the sign of the target amplitude and keeping the rest unchanged. We interpret this as a reflection of the state vector about the x-axis.
- The Invert stage can also be interpreted as a reflection about $|e\rangle$. This can be seen in two different ways: (1) $V = -HPH$ is a π phase flip of the 0 state (a reflection about the y-axis) with a basis change to $|e\rangle$ (via the two H's). (2) Invert can be expressed as the operator $2|e\rangle\langle e| - I$, which effectively reflects about $|e\rangle$.

Hence, the Flip-Invert block is a composition of two reflections, one about the x-axis and one about $|e\rangle$. It is known in geometry that the composition of two reflections about two non-parallel axes is equivalent to a rotation about the intersection point of the two axes by double the angle between them, which $\varphi/2$.

Hence, the overall effect is a contraclockwise rotation of the state vector about the origin by an angle φ. This is consistent with what we found algebraically via the amplitude recurrences.

Remarks

- This algorithm was proposed by Lov Grover c. 1996.
- The loop in the circuit diagram of this algorithm is not physical. Qubits do not move in wires from one gate to the next. They stay in place, and we subject them to varying conditions, e.g. microwave pulses of varying frequencies and durations. Repetitions are effected by re-applying these waves.
- The Flip-Invert operation V is known as the Grover *Diffusion Operator*.
- The algorithm can be easily extended to handle multiple targets; i.e. more than one x value at which $f(x)=1$.
- In addition to speeding up brute-force search (e.g. to find the key of a symmetric cipher), the Grover algorithm can be used to attack cryptographic hash functions (e.g. to find collisions or determine a pre-image.