CSE491/596 (Extra) Grover's Algorithm---and Feynman's Briar Patch?

All of our previous quantum algorithms have been ones where an n-qubit Hadamard transform has been applied once, then an oracle gate or other computation to create a functional superposition

$$\sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle,$$

and then *one* further transform---Hadamard or Fourier---before measuring the entire output. [Footnote: the notation $|xf(x)\rangle$ for the body of the sum is equivalent.] Further iterations are managed by a routine with *classical* control. Grover's algorithm, however, has successive quantum stages that each use two banks of Hadamard gates. The $2^n \times 2^n$ matrices $\mathbf{H}^{\otimes n}$ are just as easy as any other in a "Schrödinger-style" simulation where you multiply matrices. But in a "Feynman-style" simulation where we count nondeterministic witness strings, the repeated Hadamard transforms mushroom the witness space. (This is why the `groverDemo` in my simulator has not been implemented yet.)

Grover's algorithm as originally presented applies only at "witness scale": a space of $N = 2^q$ potential witness strings using $q = q(n)$ qubits, **not** N separate physical locations as commonly talked about. Whether it can apply to N physical sites with $\widetilde{O}\left(\sqrt{N}\right)$ **effort** is IMHO controversial. However, at witness scale, there aren't even $\sqrt{N}$ physical sites, only q qubits with basis vectors $|0^q\rangle$ through $|1^q\rangle$. A solution set $S \subseteq \{0,1\}^q$ is represented by the "hit vector" $\mathbf{h_S}$ defined by

$$\mathbf{h}_S(y) = \begin{cases} \dfrac{1}{\sqrt{|S|}} & \text{if } y \in S \\ 0 & \text{otherwise} \end{cases}.$$

This is just the normalized sum of the basis vectors corresponding to strings in S. *Except:* if S is empty, then this would be the zero vector in $\mathbb{C}^N$, which is not a legal quantum state. There is a further worm in this apple:

- There are $2^N = 2^{2^q}$ different possible subsets S.
- Thus it seems that each hit vector $\mathbf{h}_S$ carries N bits of information.
- However, we are using only $q \ll N$ qubits, and we need to note:

Holevo's Theorem: It is not possible to extract more than q bits of classical information from any q-qubit quantum state.

Thus, like the situation with graph states, the quantum representation of solution sets is inevitably ***lossy***.

This is part reason for Lov Grover's original attention only to singleton sets $S = \{y\}$, whereupon we simply have $\mathbf{h_S} = |y\rangle$. Then distinguishing among the 2^q possibilities (all of them not the empty set) involves only q bits of information. Any setting that allows $|S| > 1$ involves some information smearing. The final point here is that the measurement at the end of the algorithm will give you just *one* witness, not necessarily the whole set of them. When $S = \{y\}$ it is the whole set, but otherwise not.

At witness scale, the running time is not sub-linear but merely quadratically sub-exponential: $\widetilde{O}\left(\sqrt{N}\right) = n^{O(1)}2^{q(n)/2}$, which is still 2-to-the-linear exponential time, not even $2^{q(n)^{1/2}}$. Here the multipler---which is the time per iteration---includes the polynomial gate count $s = s(n) = n^{O(1)}$. As an aside, I am skeptical that this is a true measure of quantum effort. Well, we should examine the quantum circuits, after seeing the idea of the algorithm.

[I will look for a way to simplify the next section. In any event, my lecture will skim over it in order to focus ont he wheel diagram in the enxt section.]

How Grover Search Works

Grover's algorithm actually operates completely within a 2-dimensional subspace of $\mathbb{C}^N$. The subspace is spanned by two vectors: $\mathbf{h_S}$ and the vector $\mathbf{j} = \mathbf{H}^{\otimes q}|0^q\rangle$. (Unless $S = \{0,1\}^q$ *in toto*, which makes them equal.) We do not know $\mathbf{h_S}$ in advance, but we do know $\mathbf{j}$. The "miss" vector $\mathbf{m}_S = \mathbf{h}_{\sim S}$ also belongs to the subspace, since it equals

$$\frac{\sqrt{N}\cdot\mathbf{j} - \sqrt{|S|}\cdot\mathbf{h_S}}{\sqrt{N-|S|}}, \quad \text{so that} \quad \mathbf{j} = \frac{\sqrt{N-|S|}}{\sqrt{N}}\mathbf{m}_S + \frac{\sqrt{|S|}}{\sqrt{N}}\mathbf{h}_S.$$

We don't know $\mathbf{m}_S$ either, but provided S is given by a polynomial-time decidable witness predicate $R(x,y)$ of our problem instance x, then we can reflect around it by means of the **Grover oracle**

$$U_R[xy,xy] = (-1)^{R(x,y)} = \begin{cases} -1 & \text{if } R(x,y) \\ 1 & \text{if } \neg R(x,y) \end{cases}.$$

When x is fixed, the Grover oracle drops down to an $N \times N$ diagonal matrix G_x with entry $G_x[y,y] = -1$ if $y \in S$ and $G_x[y,y] = 1$ otherwise. To compute it, we can apply an idea that the textbook calls "flipping a switch" in section 6.5 but might be better called the idea of using an extra qubit as a *catalyst*. The catalyst is that we initialize the extra qubit not to $|0\rangle$ or $|1\rangle$ but to

$$\mathbf{d} = \mathbf{H}|1\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right).$$

We can create a quantum circuit C_0 of deterministic gates only (Toffoli plus constant initializations) for the reversable form of the Boolean function $f_x(y) = R(x, y)$, which is the $(q+1)$-bit function $F_x(yb) = y(b \oplus f_x(y))$. Now define $g_x(y) \;=\; C_0\big(|y\rangle \otimes \mathbf{d}\big)$ using our catalyst. We get

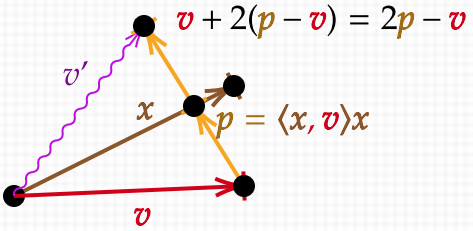
$$g_x(y) = C_0(|y\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \;=\; \frac{C_0|y0\rangle - C_0|y1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\big(|y\rangle|f_x(y)\rangle - |y\rangle|\neg f_x(y)\rangle\big)$$

$$= \begin{cases} \dfrac{|y1\rangle - |y0\rangle}{\sqrt{2}} & \text{if } f_x(y) = 1 \\ \dfrac{|y0\rangle - |y1\rangle}{\sqrt{2}} & \text{if } f_x(y) = 0 \end{cases} = \begin{cases} |y\rangle \otimes (-\mathbf{d}) & \text{if } R(x,y) \\ |y\rangle \otimes \mathbf{d} & \text{if } \neg R(x,y) \end{cases} = (-1)^{R(x,y)}|y\rangle \otimes \mathbf{d}$$

If we "throw away" the last qubit (say by measuring it and ignoring the result) then we get the Grover oracle action on the first q qubits. So for polynomial-time witness predicates $R(x, y)$, the Grover oracle is feasible to compute.

The key next point is that in the geometry of the 2-dimensional space, the Grover oracle represents reflection around the *miss* vector $\mathbf{m}_S$. Note first that $G_x\mathbf{m}_S = \mathbf{m}_S$ because no nonzero entry gets negated. And $G_x\mathbf{h}_S \;=\; -\mathbf{h}_S$ because all the nonzero entries get negated. Therefore the action of G_x in this space is *reflection about* $\mathbf{m}_s$.

The other operation we want is reflection about $\mathbf{j}$. In general, reflection of a vector v around a vector x involves first taking the projection of v onto x, which is $\langle v, x\rangle x$. Then we want to move v by twice the difference of that to v:



The matrix operator that creates the projection of an argument v along x is the **outer product** $|x\rangle\langle x|$, whose $[i, j]$ entry is $x_i\overline{x_j}$. The Dirac notation is especially handy here, because we can do

$$|x\rangle\langle x| \cdot |v\rangle \;=\; |x\rangle\langle x|v\rangle \;=\; \langle x, v\rangle|x\rangle.$$

So the operator that creates the reflection is $2|x\rangle\langle x| \;-\; \mathbf{I}$. In the case $x = \mathbf{j}$ this is given by the matrix $2\mathbf{J} - \mathbf{I}$ where each entry of $\mathbf{J}$ is $\frac{1}{N}$ and $\mathbf{I}$ is the $N \times N$ identity matrix.

Because we are talking about exponential-sized matrices, it is relevant to ask about the feasibility of computing their actions. An equation by which to build the reflection about $\mathbf{j}$ is

$$2\mathbf{J} - \mathbf{I} \;=\; \mathbf{H}^{\otimes q}(-1)^{NOR(1..q)}\mathbf{H}^{\otimes q}.$$

The $(-1)^{NOR(1..q)}$ is implemented via a controlled-$\mathbf{Z}$ gate on one qubit with controls on the other $(q-1)$ qubits---it doesn't matter which, as the gate is symmetric. By itself, that gate computes $(-1)^{AND(1..q)}$, so it is sandwiched between two banks of $\mathbf{NOT}$ gates to get the action of NOR. To see why this works, consider first that on any basis input $|x\rangle$, $\mathbf{H}^{\otimes q}|x\rangle = \frac{1}{\sqrt{N}}\Sigma_y(-1)^{x\odot y}|y\rangle$. Applying the $(-1)^{NOR(1..q)}$ gives

$$\frac{1}{\sqrt{N}}\sum\nolimits_{y\neq 0^q}(-1)^{x\odot y}|x\rangle \;+\; \frac{(-1)}{\sqrt{N}}(-1)^{x\odot 0^q}|0^q\rangle \;=\frac{1}{\sqrt{N}}\sum\nolimits_{y}(-1)^{x\odot y}|y\rangle \;-\; \frac{2}{\sqrt{N}}|0^q\rangle$$

Applying $\mathbf{H}^{\otimes q}$ again gives

$$\frac{1}{N}\sum\nolimits_{y}\sum\nolimits_{z}(-1)^{x\odot y}(-1)^{z\odot y}|z\rangle \;-\; \frac{2}{N}\sum\nolimits_{z}(-1)^{z\odot 0^q}|z\rangle \;=\frac{1}{N}\sum\nolimits_{y}\sum\nolimits_{z}(-1)^{(x\oplus z)\odot y}|z\rangle \;-\; \frac{2}{N}\sum\nolimits_{z}|z\rangle$$

Now the outer sum over y in the first term vanishes except when $z = x$, so we get

$$\frac{1}{N}\sum\nolimits_{y}|x\rangle \;-\; \frac{2}{N}\sum\nolimits_{z}|z\rangle \;=\; |x\rangle \;-\; \frac{2}{N}\sum\nolimits_{z}|z\rangle \;=\; (\mathbf{I} \;-\; 2\mathbf{J})|x\rangle.$$

This is (-1) times what we expected, but the global scalar does not matter. The last thing to say is that whenever v belongs to our 2-dimensional subspace, the reflection of v around $\mathbf{j}$ stays within it.

The Search Process

Let α stand for the angle between $\mathbf{j}$ and $\mathbf{m}_S$. Then $\alpha = \cos^{-1}\langle \mathbf{j}, \mathbf{m}_S\rangle = \sin^{-1}\langle \mathbf{j}, \mathbf{h}_S\rangle$. When $|S| = o(N)$ we can estimate

$$\alpha \;=\; \sin^{-1}\langle \mathbf{j}, \mathbf{h}_S\rangle \;\sim\; \langle \mathbf{j}, \mathbf{h}_S\rangle \;=\; \frac{\sqrt{|S|}}{\sqrt{N}}.$$

The number of iterations (each a pair of reflections) we will need is about $\frac{\pi/2}{2\alpha} = \frac{\pi}{4\alpha} \approx \frac{\pi}{4}\sqrt{\frac{N}{S}}$. This is always about the square root of the expected time for guessing uniformly at random and verifying. If

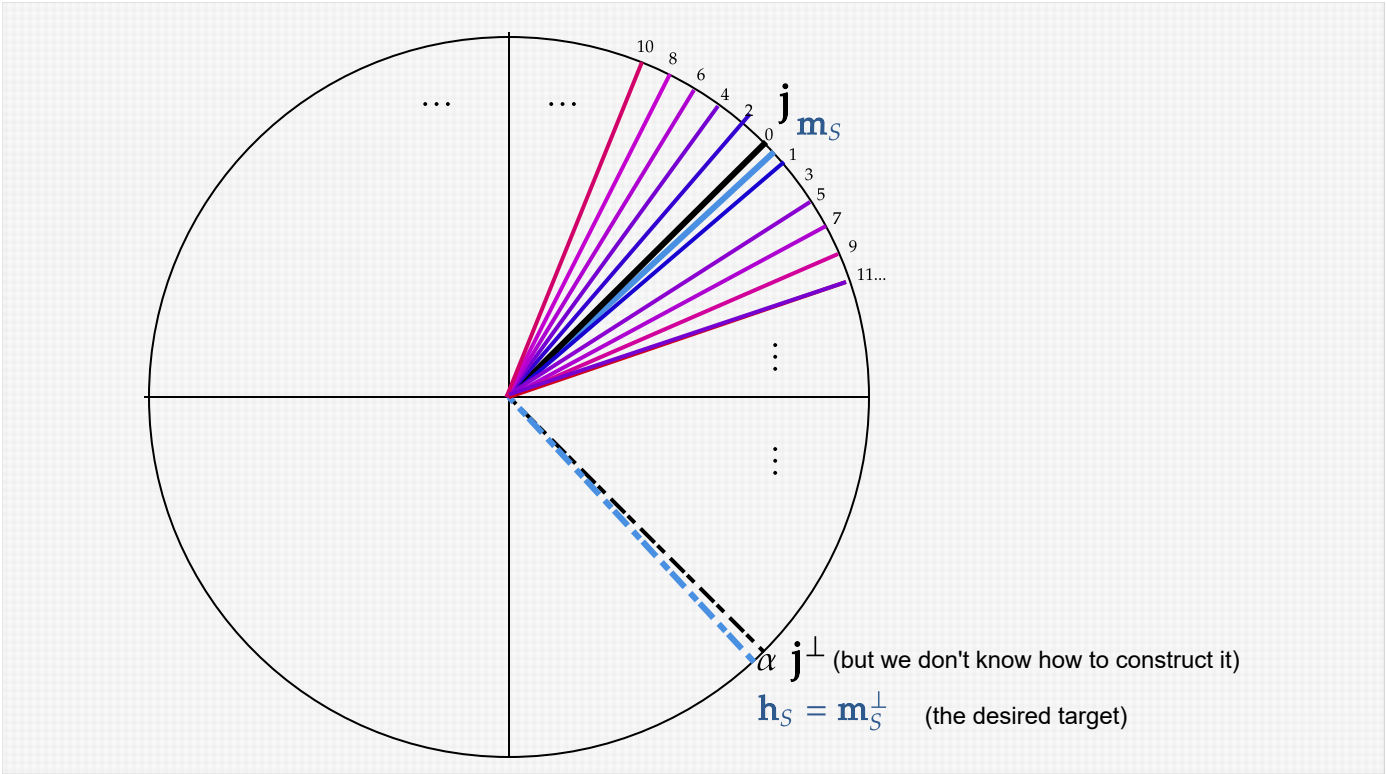
we know $|S|$, then we know how many iterations to make before measuring; if we don't know $|S|$, then there are further tradeoffs discussed later. In any event, unless $|S| = \Omega(N)$, we have $\alpha = o(1)$, so that the angle α is best pictured as very small. When $|S| \leq \sqrt{N}$, we have

$$\frac{1}{\sqrt{N}} \leq \alpha \leq \frac{1}{\sqrt[4]{N}}$$

as the most relevant range of angles. Now to summarize what we know and don't know:

1. We know a vector $\mathbf{j}$ in the two-dimensional subspace H generated by the hit vector $\mathbf{h}_S$ and its orthogonal complement, the miss vector $\mathbf{m}_S$.
2. The goal is to build a quantum state ϕ whose vector is within ϵ of $\mathbf{h}_S$, so that measuring ϕ will with probability $\approx 1 - \epsilon$ yield a member of S.
3. We know that $\mathbf{j}$ is close to $\mathbf{m}_S$, so that $\mathbf{j}^\perp$ is close to $\mathbf{h}_S$ (or opposite to $\mathbf{h}_S$---either way, measuring $\mathbf{j}^\perp$ would yield a solution whp.), but we have no idea how to construct $\mathbf{j}^\perp$ within H.
4. What we do have are feasible circuit components computing reflection around $\mathbf{m}_S$ and reflection aound $\mathbf{j}$ that stay within H.
5. If we know $|S|$, then we know the number of iterations that produces a vector ϕ closest to $\mathbf{h}_S$. Moreover, ϕ will be within angle α of $\mathbf{h}_S$.

Here is a diagram of the iteration process. It is different from most other diagrams by emphasizing the smallness of α and not giving the impression that $\mathbf{j}^\perp$ is knowable by aligning it with vertical or horizontal axes. The iteration starts by reflecting the known vector $\mathbf{j}$ around $\mathbf{m}_S$. The next five iterations (each a rotation by 2α effected by two reflections) are shown and color-coded.



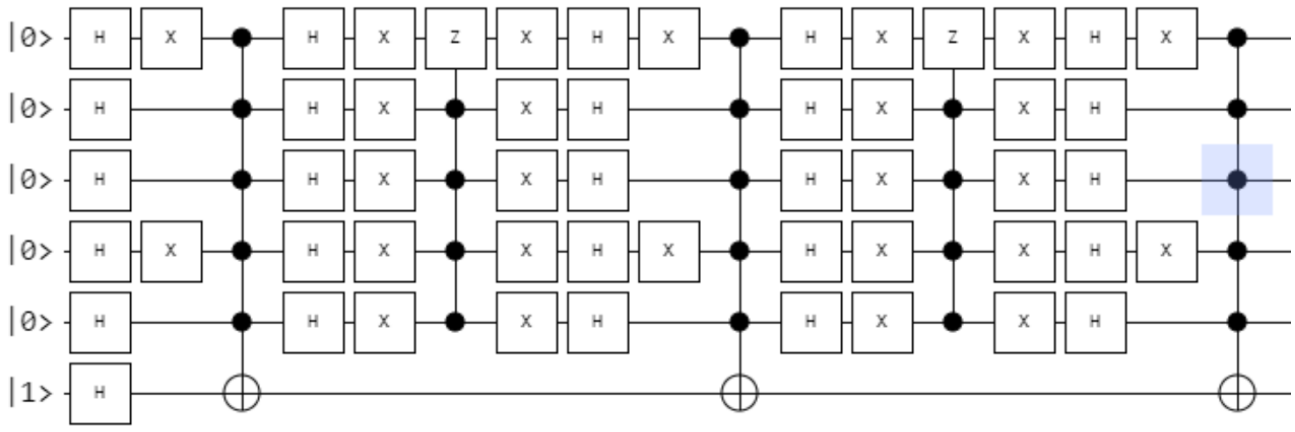
It may seem strange that we cannot jump straight to $\mathbf{j}^{\perp}$ from $\mathbf{j}$ or otherwise leverage the initial proximity to $\mathbf{m}_S$ in a way that would at least allow taking bigger steps toward $\mathbf{h}_S$ than repeated rotation by 2α. It looks even more enticing upon realizing that getting within 45° of $\mathbf{h}_S$, that means anywhere in the lower-right quadrant shown, gives at least a $\sin^2\left(\frac{\pi}{4}\right) = \frac{1}{2}$ chance of the measurement giving a string in S.

The picture makes it look like we could hit that quadrant quickly just by throwing darts at it. But the point is that the "dartboard" H is hidden inside a vastly higher dimensional space, and we have no direct information besides the $\mathbf{j}$ vector of how it lies. In fact, the above process is tightly optimal.

If $|S|$ is unknown, we can guess a stopping time $t \leq \sqrt{N}$ uniformly at random. Now the "dartboard" reasoning works in our favor since everything happens within the subspace H, and the expected time to find a solution is only a constant factor greater than when $|S|$ is known.

Circuit Implementation and Problematic Aspects

The Grover oracle is deterministic except for the single Hadamard gate used to initialize the catalyst qubit to the difference state $\mathbf{d}$. We do not have to re-initialize it, however, because the output after the evaluation remains $(-1)^{R(x,y)}|y\rangle \otimes \mathbf{d}$. The issue is the reflection about $\mathbf{m}_S$. Done straightforwardly, it is heavy on the $\mathbf{H}$ gates, as evinced by the following example in Davy Wybiral's quantum web applet:



(show all)

0.11132812+0.00000000i	|011000>	1.2394%
-0.11132812+0.00000000i	|011001>	1.2394%
0.33007813+0.00000000i	|011010>	10.8952%
-0.33007813+0.00000000i	|011011>	10.8952%
0.11132813+0.00000000i	|011100>	1.2394%

Here the Grover oracle is $\bar{x}_1 \wedge x_2 \wedge x_3 \wedge \bar{x}_4 \wedge x_5$ giving $S = \{01101\}$. This is implemented as a multi-controlled flip of the catalyst line (where a single $\mathbf{H}$ follows the initial 1) with $\mathbf{X}$ gates to make $\bar{x}_1$ and

$\overline{x}_4$. The initial bank of Hadamards on the first five qubits is to create the $\mathbf{j}$ vector on them. The four other banks, however, are for the two reflections about $\mathbf{j}$. The angle α is $\sin^{-1}\left(1/\sqrt{32}\right) = 0.1777...$ radians. The desired number of iterations is $\frac{\pi}{4\alpha} = 4.42$; the diagram counts as 2.5 iterations. This is close enough to show more probability accumulating on the string 01101 on the first five qubits.

If we make a polynomial simulation out of this, however, the Hadamard gates for the reflections give rise to 20 new variables. The number of Feynman paths grows by a factor of more than 1,000 per iteration. (This also causes major branching in the witness space for problem 3 on assignment 4.) This growth quickly chokes the path-counting simulation written in C++ which I've demo'ed.

The multi-controlled Z gate has its own element of excess. Yes, OK, the Grover oracle in this case is also multi-controlled, but one expects to expend more effort on it---and it could be a larger network of gates with only one control each. The reflection about $\mathbf{j}$, however, really uses all the controls. IBM researchers have found even the double-controlled Toffoli gate to be difficult to engineer, which is why their preferred basis consists of $\mathbf{H}$, $\mathbf{CNOT}$, and the $\mathbf{T}$ gate.