ارائه ۱۰	پردازش اطلاعات کوانتومی پاییز ۱۴۰۲	دکتر احمد خونساری

مقدمه

در این ارائه به الگوریتم Grover می‌پردازیم. فرض کنید یک آرایه به طول $N = 2^n$ به ما داده شده است و ما میخواهیم مکان یک قلم داده را در این آرایه پیدا کنیم. محتوای این آرایه هیچ نظم و ترتیبی که از قبل از آن اطلاع داشته باشیم ندارد. به این ترتیب در حالت کلاسیک مجبور هستیم که مدخل‌های این آرایه را پشت سر هم بررسی کنیم تا به داده مورد نظر برسیم. بنابراین در حالت کلاسیک پیچیدگی حل این مسئله $O(N)$ است. در این ارائه می‌بینیم که اگر مسئله را به صورت کوانتومی حل کنیم می‌توانیم زمان جستجو را به $O(\sqrt{N})$ کاهش دهیم.

جستجو در اطلاعات بدون ساختار

در این ارائه (بدون از دست دادن کلیت) فرض می‌کنیم که یک سروش [1] به شکل زیر وجود دارد: فرض کنید که z یک رشته دلخواه، از پیش مشخص و ثابت است. سروش به این ترتیب عمل می‌کند که وقتی ورودی $x \neq z$ به آن داده شود، مقدار خروجی با ورودی برابر خواهد بود. اما اگر ورودی z باشد، مقدار خروجی برابر ورودی به اضافه یک فاز منفی می‌شود. به این ترتیب عملگر آن را می‌توان به صورت زیر نشان داد:

$$V|x\rangle = \begin{cases} |x\rangle & \text{اگر } x \neq z \\ -|x\rangle & \text{اگر } x = z \end{cases} \tag{۱}$$

با استفاده از چنین سروشی می‌توان طیف وسیعی از مسائل جستجو را مدل‌سازی کرد. به این ترتیب که کافی است یک تابع $f(x)$ تعریف کنیم که هر زمان $x = z$ است، مقدار آن برابر یک می‌شود. در غیراینصورت مقدار آن صفر است.

[1] Oracle

به این ترتیب می‌توانیم یک سروش به شکل زیر تعریف کنیم (توجه کنید که این ماتریس یکانی است):

$$V = \begin{bmatrix} (-1)^{f(0)} & 0 & \dots & 0 \\ 0 & (-1)^{f(1)} & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \dots & (-1)^{f(N-1)} \end{bmatrix}, \tag{۲}$$

که تساوی زیر را نتیجه می‌دهد:

$$V|x\rangle = (-1)^{f(x)}|x\rangle. \tag{۳}$$

فرض کنید هدف ما این است که $|x\rangle = |z\rangle$ را پیدا کنیم. در ابتدا هیچ اطلاعی از z نداریم. بنابراین احتمال اینکه هر ورودی همان مقدار مورد نظر ما باشد برای ما برابر است و از دید ما با هم فرقی ندارند. بنابراین اگر ورودی را در حالت برهم‌نهاده آماده کنیم و سروش را اعمال کنیم و سپس خروجی را اندازه‌گیری کنیم، سیستم به یکی از ورودی‌های ممکن فرو می‌شکند و دقیقاً معادل این است که یک ورودی را به تصادف انتخاب کرده‌ایم و سپس به ازای آن تابع را ارزیابی کرده باشیم. احتمال اینکه با این روش بتوانیم مقدار مدنظر را پیدا کنیم برابر $\frac{1}{N}$ خواهد بود. برای حل این مسئله از تکنیکی به نام «تقویت دامنه [۲]» استفاده می‌شود. به صورت خاص اگر ورودی را در حالت زیر آماده کرده باشیم:

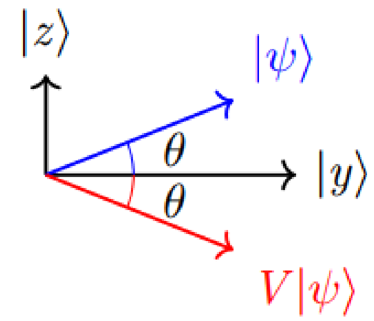
$$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N}|x\rangle, \tag{۴}$$

آنگاه سعی می‌کنیم آن را به حالت زیر ببریم:

$$\sum_{x=0}^{N}\alpha_x|x\rangle, \tag{۵}$$

که در آن α_z به صورت قابل توجهی از بقیه $\alpha_{x\neq z}$ بزرگتر باشد. آنگاه اگر در این حالت اندازه‌گیری را انجام بدهیم، احتمال مشاهده $|z\rangle$ بسیار بیشتر خواهد بود و با حالت حدس تصادفی متفاوت می‌شود. در ادامه بررسی می‌کنیم که چگونه می‌توان این کار را انجام داد.

توجه کنید که اگر فرض کنیم $|y\rangle = \frac{1}{\sqrt{N-1}}\sum_{x:f(x)=0}|x\rangle$ باشد (یعنی حاصل برهم‌نهی تمام حالت‌هایی باشد که به ازای آن‌ها تابع مدنظر برابر صفر است)، آنگاه $|y\rangle$ بر $|z\rangle$ عمود است. به خاطر بیاورید که $|x\rangle$ به ازای $x \in \{0,1\}^n$ پایه‌های محاسباتی هستند و بر هم عمود هستند. بنابراین اگر $N-1$ پایه را انتخاب کنیم، هر ترکیب خطی آن‌ها بر آن پایه‌ای که انتخاب نشده است عمود است. با دانستن این موضوع، می‌توان متوجه شد که حاصل برهم‌نهی تمام حالت‌ها یعنی $|\psi\rangle$ نیز بین $|z\rangle$ و $|y\rangle$ قرار می‌گیرد. بنابراین می‌توان آن را در فضای دوبعدی به شکل ۱ نمایش داد.



شکل ۱: وضعیت آغازین حالت‌های $|y\rangle$، $|\psi\rangle$ و $|z\rangle$

حال توجه کنید که اگر سروش را به حالت $|\psi\rangle$ اعمال کنیم، دامنه‌ها تغییر نمی‌کند اما یک فاز منفی به حالت $|z\rangle$ اضافه می‌شود و بقیه نیز بدون تغییر باقی می‌مانند. معنی این عملیات را اینگونه می‌توان تفسیر کرد که این بار $|y\rangle$ و $-|z\rangle$ برهم‌نهاده می‌شوند. نتیجه $V|\psi\rangle$ مشابه حالتی است که $|\psi\rangle$ نسبت به بردار $|y\rangle$ قرینه شده باشد. شکل ۱ را ببینید. به این عملیات «معکوس کردن فاز [3]» می‌گویند. در گام بعدی، حالت بدست آمده نسبت به حالت $|\psi\rangle$ قرینه می‌شود (به آن «معکوس کردن حول میانگین [4]» می‌گویند). در شکل ۲ می‌توانید ببینید که این کار حالت کلی را به سمت $|z\rangle$ می‌برد که حالت مطلوب ما است. بنابراین احتمال اندازه‌گیری z بیشتر می‌شود. برای این کار از یک عملگر یکانی به فرم زیر استفاده می‌شود:

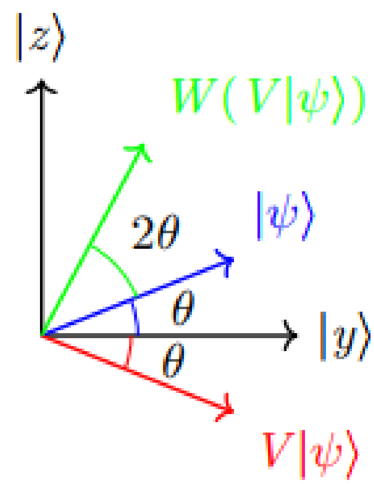
$$W = 2|\psi\rangle\langle\psi| - I. \qquad (۶)$$

برای اینکه ببینید چرا به این عملیات معکوس کردن حول میانگین گفته می‌شود و قرینه‌سازی چگونه انجام می‌شود، اسلاید شماره ۸ را ببینید. به صورت خاص می‌توان ملاحظه کرد که اگر هر بردار دلخواه را به دو مؤلفه «در راستای $|\psi\rangle$» و «عمود بر $|\psi\rangle$» تقسیم کرد و سپس W را اعمال کرد، آنگاه مؤلفه‌ای که در راستای $|\psi\rangle$ است عوض نمی‌شود، اما مؤلفه‌ای که عمود بر $|\psi\rangle$ است در یک منفی ضرب می‌شود. به این ترتیب مشخص است که با اعمال متناوب مدارهای V و W بردار ابتدایی $|\psi\rangle$ به $|z\rangle$ نزدیکتر می‌شود. با استفاده از روابط مثلثاتی ساده می‌توان نشان داد که $O(\sqrt{N})$ بار اعمال این مدارها برای رسیدن به احتمال مطلوب کافی خواهد بود. به صورت خاص، اگر در ابتدا زاویه بردار $|\psi\rangle$ و $|y\rangle$ برابر $\theta \approx \frac{1}{\sqrt{N}}$ باشد، پس از هر بار اعمال به اندازه 2θ به بردار هدف $|z\rangle$ نزدیک می‌شویم. فاصله اولیه $\frac{\pi}{2} - \theta$ است. بنابراین تعداد دوران‌های مورد نیاز به ترتیب زیر می‌شود:

$$\frac{\frac{\pi}{2} - \theta}{2\theta} \approx \frac{\pi\sqrt{N}}{4}. \qquad (۷)$$

اسلایدهای شماره ۱۱ و ۱۳ را ببینید.

[2]Amplitude Amplification
[3]Phase Inversion
[4]Inversion About Mean



شکل ۲: وضعیت پس از قرینه‌سازی نسبت به $|\psi\rangle$

مدار

برای اینکه مدار کوانتومی الگوریتم Grover را بسازیم، فرض می‌کنیم که سروش V به ما داده شده است. کافی است مدار W را بسازیم. برای دیدن حالت کلی به اسلایدهای ۹ و ۱۰ مراجعه کنید. اما به عنوان مثال، یک سیستم دو کیوبیتی را در نظر بگیرید:

$$W = 2|++\rangle\langle++| - I. \tag{۸}$$

سپس این مدار را به صورت زیر با مدار هادامارد ترکیب می‌کنیم:

$$\begin{aligned} H^{\otimes 2} W H^{\otimes 2} &= H^{\otimes 2}(2|++\rangle\langle++| - I)H^{\otimes 2} && (۹)\\ &= 2H^{\otimes 2}\langle++|\langle++|H^{\otimes 2} - H^{\otimes 2} I H^{\otimes 2} && (۱۰)\\ &= 2|00\rangle\langle 00| - I. && (۱۱) \end{aligned}$$

سپس، از مدار X استفاده می‌کنیم:

$$\begin{aligned} X^{\otimes 2}(2|00\rangle\langle 00| - I)X^{\otimes 2} &= 2|11\rangle\langle 1| - I && (۱۲)\\ &= -\begin{bmatrix} 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1 \end{bmatrix}, && (۱۳) \end{aligned}$$

که ماتریس اخیر مدار Z کنترل‌شده است که در یک منفی ضرب شده است. به خاطر بیاورید که دریچه Z حالت «صفر» را تغییر نمی‌دهد اما به حالت «یک» فاز منفی اضافه می‌کند. ابتدا مدار هادامارد اعمال می‌شود، سپس مدار X اعمال می‌شود، پس از آن مدار Z کنترل‌شده اعمال می‌شود. سپس مدارهای X و H اعمال می‌شوند. برای پیاده‌سازی مدار Z کنترل‌شده نیز می‌توان از مدار هادامارد و دریچه Toffoli استفاده کرد که در اسلایدها به آن پرداخته شده است.

شبیه‌سازی

برای شبیه‌سازی، ابتدا کتابخانه‌های مورد نیاز را وارد می‌کنیم:

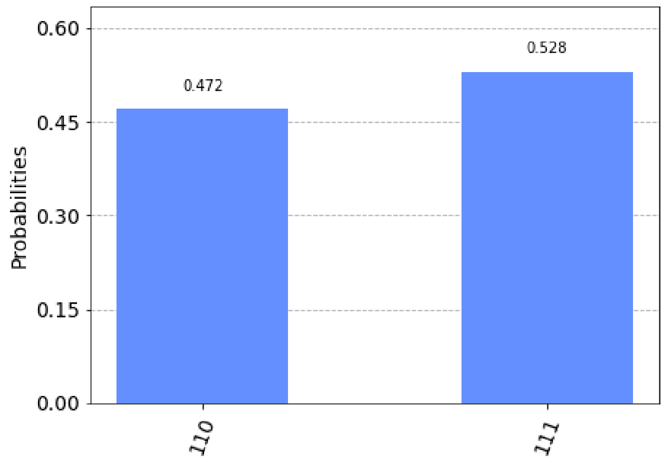
```
from qiskit import Aer, QuantumCircuit, execute
from qiskit.visualization import plot_histogram
```

در اینجا ما یک سروش را طراحی می‌کنیم که به ازای $\langle 110|$ و $\langle 111|$ یک فاز منفی اضافه می‌کند و در غیراینصورت خروجی را تغییر نمی‌دهد:

```
def get_oracle(n):
    qc = QuantumCircuit(n)
    qc.cz(1, 2) # -> V|110> = -|110>, V|111> = -|111>
    oracle_ex3 = qc.to_gate()
    oracle_ex3.name = "V"
    return oracle_ex3
```

سپس، مدار W را طراحی می‌کنیم:

```
def get_W(n):
    qc = QuantumCircuit(n)
    for qubit in range(n):
        qc.h(qubit)
    for qubit in range(n):
        qc.x(qubit)
    qc.h(n-1)
    qc.mct(list(range(n-1)), n-1)
    qc.h(n-1)
    for qubit in range(n):
       qc.x(qubit)
    for qubit in range(n):
       qc.h(qubit)
    W = qc.to_gate()
    W.name = "$W$"
    return W
```



شکل ۳: حاصل شبیه‌سازی و مشاهده دو خروجی مطلوب

با توجه به اینکه $N = 2^3 = 8$ و دو مقدار هدف وجود دارند، یک بار اعمال مدارها کافی است. به ترتیب زیر می‌توان مدار Grover را ایجاد کرد:

```
qc = QuantumCircuit(3)
for i in range(n):
    qc.h(i)
my_oracle = get_oracle(n)
W = get_W(n)
qc.append(my_oracle, [0,1,2])
qc.append(W, [0,1,2])
qc.measure_all()
qc.draw("mpl")
```

به صورت زیر می‌توانیم آن را شبیه‌سازی کنیم:

```
backend = Aer.get_backend('qasm_simulator')
results = execute(qc, backend=backend, shots=1024).result()
answer = results.get_counts()
plot_histogram(answer)
```

حاصل شبیه‌سازی در شکل ۳ آمده است. می‌بینیم که تنها احتمال مشاهده دو حالت مورد نظر وجود دارد.