ALGORITMI DI RICERCA: GROVER

Supponiamo di avere una lunga lista di N elementi e, tra questi, supponiamo ci sia un elemento speciale che vogliamo localizzare: utilizzando un algoritmo classico dovremmo analizzare, in media, $\frac{N}{2}$ elementi prima di trovarlo o, nel caso peggiore, tutta la lista. L'algoritmo di Grover velocizza tale ricerca riducendo il numero di passi necessari a $O(\sqrt{N})$. Inoltre, tale algoritmo è estremamente generico, dato che non usa in alcun modo le caratteristiche intrinseche della particolare lista analizzata.

In particolare, noi tratteremo dapprima il caso generico con k elementi segnati su N totali, per poi procedere con esempi e implementazioni specifiche.

0.1 L'algoritmo

Prima di procedere con la descrizione dell'algoritmo, diamo un'idea di come sia impostato e funzioni. Definiamo quindi lo stato "fortunato" $|w\rangle \in \mathbb{C}^N$, le cui entrate sono tali che

$$w_x = \begin{cases} \frac{1}{\sqrt{k}} & \text{se } x \text{ è uno degli elementi cercati} \\ 0 & \text{altrimenti} \end{cases}$$

dove k è il numero di elementi che corrispondono alla nostra ricerca. La denominazione "fortunato" deriva dal fatto che se misurassimo tale vettore otterremmo una soluzione con probabilità 1: l'obiettivo dell'algoritmo sarà dunque costruire un vettore $|w'\rangle$ molto vicino a $|w\rangle$, così che la misurazione possa restituire una soluzione con probabilità elevata. Dato che un qualsiasi vettore inizializzato randomicamente sarà quasi certamente molto lontano da $|w\rangle$, il compito dell'algoritmo sarà quello di modificare lo stato di partenza così da avvicinarlo a $|w\rangle$ utilizzando solo ciò che sappiamo a priori di tale vettore, ossia che tutte le entrate corrispondenti agli elementi cercati hanno lo stesso valore, così come tutti gli indici degli elementi da rifiutare.

Questa proprietà, detta **solution-smoothness**, si estende a tutti i vettori $|v\rangle$ nel sottospazio generato da $|w\rangle$ e $|s\rangle$, che ricordiamo essere lo stato di sovrapposizione equiprobabile per tutte le entrate, dato che

$$v_x = \alpha w_x + \beta s_x = \begin{cases} \alpha\frac{1}{\sqrt{k}} + \beta\frac{1}{\sqrt{N}} & \text{se } x \text{ è uno degli elementi cercati} \\ \beta\frac{1}{\sqrt{N}} & \text{altrimenti} \end{cases}$$

con $\alpha, \beta \in \mathbb{C}$ e $|v\rangle \in \text{Span}(|w\rangle, |s\rangle)$. Questa proprietà ci permette di riflettere uno qualsiasi di questi vettori $|v\rangle$ rispetto a $|w\rangle$ usando l'oracolo di Grover. In realtà, se volessimo riflettere rispetto a $|w\rangle$ dovremmo costruire l'oracolo usando il complementare dell'insieme di

riferimento e, a tal fine, definiamo il vettore "fallimento", corrispondente allo stato $|l\rangle$, come

$$|l\rangle = \frac{1}{\sqrt{N-k}}(\sqrt{N}\,|s\rangle - \sqrt{k}\,|w\rangle)$$

che è solution-smooth dato che le sue entrate sono pari a 0 se x è soluzione e a $1/\sqrt{N-k}$ se non lo è. Inoltre, è importante osservare che tale vettore è ortogonale a $|w\rangle$, e che quindi l'insieme $\mathcal{B} = \{|w\rangle, |l\rangle\}$ è una base per un sottospazio di dimensione 2.

Ora, la riflessione rispetto a $|l\rangle$ utilizza **l'oracolo di Grover**, ossia l'operatore la cui matrice associata $\mathbf{U}_f$ è tale che

$$\mathbf{U}_f[x,x] = (-1)^{f(x)} = \begin{cases} -1 & x \in S \\ 1 & x \notin S \end{cases}$$

dove con S indichiamo l'insieme degli elementi cercati, e zero fuori dalla diagonale. In particolare, f è una funzione binaria che *controlla* se l'elemento in input è uno degli elementi cercati o meno, ossia

$$f(x) = \begin{cases} 1 & x \in S \\ 0 & x \notin S \end{cases}$$

e quindi nel nostro caso è equivalente alla funzione caratteristica dell'insieme S, ossia $f \equiv \mathbb{1}_S$. Dato che verificare che un elemento sia soluzione del nostro problema (i.e. $f(x) = 1$) è classicamente fattibile, l'oracolo di Grover è computazionalmente fattibile, e con esso la riflessione che comporta.

Dato che $|w\rangle, |l\rangle$ generano un sottospazio di dimensione due, utilizzeremo un approccio geometrico per spiegare come l'algoritmo si avvicini al vettore desiderato: pensiamo dunque al vettore $|l\rangle$ come asse delle ascisse, al vettore $|w\rangle$ come asse delle ordinate, e osserviamo che il vettore $|s\rangle$ è nella regione compresa tra questi due vettori, ossia $|s\rangle$ forma un angolo $0 \leq \theta \leq \frac{\pi}{2}$ con $|l\rangle$. In particolare, dato che a priori potremmo scegliere una posizione a caso e sperare contenga uno dei k elementi, inizializzeremo il nostro vettore di partenza a $|s\rangle$, avendo dunque

$$\cos(\theta) = \frac{\langle s \mid l \rangle}{\| |s\rangle \| \cdot \| |l\rangle \|} = \sqrt{\frac{N-k}{N}} \Rightarrow \sin^2(\theta) = \frac{k}{N}$$

ossia, il quadrato del seno del nostro angolo iniziale corrisponde alla probabilità di selezionare l'elemento cercato scegliendone uno randomicamente. Più in generale, possiamo scrivere un qualsiasi stato quantistico come sovrapposizione delle proiezioni sugli assi, ossia

$$|\Psi\rangle = \cos(\theta)\,|l\rangle + \sin(\theta)\,|w\rangle$$

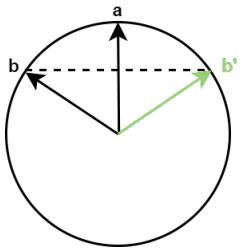
e dunque $\sin^2(\theta)$ sarà sempre la probabiltà di ottenere lo stato $|w\rangle$ misurando lo stato che stiamo analizzando.

L'idea di base è dunque quella di utilizzare le riflessioni per ottenere un angolo che si avvicini il più possibile a $\frac{\pi}{2}$, così da ottenere un risultato ottimale dalla misurazione, sfruttando il seguente risultato geometrico:

Teorema 0.1. *La composizione di due riflessioni* $\mathbf{Ref}_{AC}, \mathbf{Ref}_{AB}$ *per linee non parallele è una rotazione rispetto al punto di intersezione delle due linee pari al doppio dell'angolo formato da esse, ossia*

$$\mathbf{Ref}_{AC} \circ \mathbf{Ref}_{AB} = \mathbf{Rot}_{A,2\angle BAC}$$

Ma come sono definite matematicamente le riflessioni? Data la natura geometrica e generale di tale argomento, per questo breve paragrafo ritorneremo alla notazione vettoriale, costruendo l'operatore unitario $\mathbf{Ref_a}$, il cui effetto è riflettere un qualsiasi vettore $\mathbf{b}$ rispetto al vettore $\mathbf{a}$.
Geometricamente, partendo dalla punta del vettore $\mathbf{b}$ si traccia la retta perpendicolare al corpo di $\mathbf{a}$ continuando per la stessa distanza fino al punto $\mathbf{b'}$, che giace dunque anch'esso sulla sfera unitaria dello spazio di Hilbert. Osserviamo dunque che tale operazione è unitaria, dato che preserva la sfera unitaria ed è la sua stessa inversa.
Sempre in termini geometrici, abbiamo che il punto sul corpo di $\mathbf{a}$ sarà la proiezione di $\mathbf{b}$ su $\mathbf{a}$ e sarà dato da



$$\mathbf{a'} = \mathbf{a}\langle \mathbf{a}, \mathbf{b} \rangle$$

da cui otteniamo

$$\mathbf{b'} = 2(\mathbf{b} - \mathbf{a}\langle \mathbf{a}, \mathbf{b} \rangle) = \mathbf{b} - 2\mathbf{b} + 2\mathbf{a}\langle \mathbf{a}, \mathbf{b} \rangle = -\mathbf{b} + 2\mathbf{a}\langle \mathbf{a}, \mathbf{b} \rangle = (2\mathbf{P_a} - \mathbb{I})\mathbf{b}$$

dove $\mathbf{P_a} = \mathbf{a}^T\mathbf{a} \overset{braket}{\equiv} |a\rangle \langle a|$ è detto **operatore di proiezione**.

L'algoritmo si sviluppa dunque secondo i seguenti step:

1) Inizializza lo stato $|0^n\rangle$ generando la sovrapposizione equiprobabile $|a\rangle := |s\rangle$ tramite un gate Hadamard;

2) Calcola $\theta = \sin^{-1}\left(\sqrt{\frac{k}{N}}\right)$ e $t_k = \left\lfloor \frac{\pi}{4\theta} \right\rfloor$;

3) Ripeti per t_k volte i seguenti comandi:

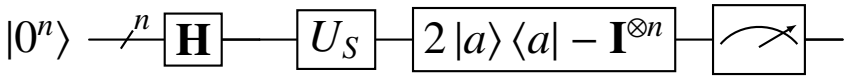
   3.1) Applica $\mathbf{Ref_m}$ allo stato $|a\rangle$ tramite oracolo di Grover $\mathbf{U}_f$ per ottenere lo stato $|a'\rangle$;

   3.2) Applica $\mathbf{Ref_a} = 2|a\rangle\langle a| - \mathbf{I}^{\otimes n}$ allo stato $|a'\rangle$ per ottenere un nuovo stato $|a\rangle$;

4) Misura lo stato finale $|a\rangle$, ottenendo una stringa $x \in \{0,1\}^n$;

5) Se $x \in S$, stop. Altrimenti riparti dal punto 1 dato che il sistema è collassato nella misurazione.

A livello circuitale, tale iterazione può essere tradotta come segue:



Riportiamo di seguito un'iterazione dell'algoritmo in forma grafica, così da poter fissare meglio le idee:

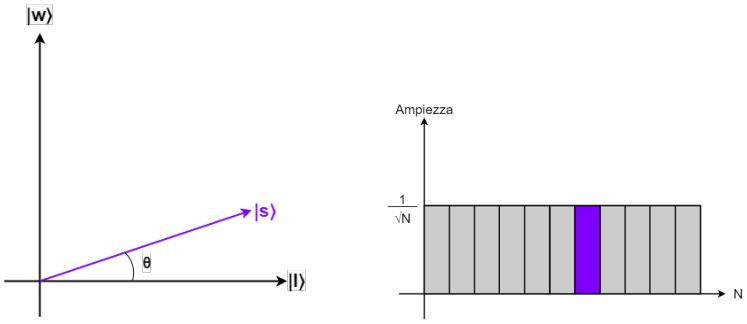


Figura 1: Inizializzazione di $|0^n\rangle$

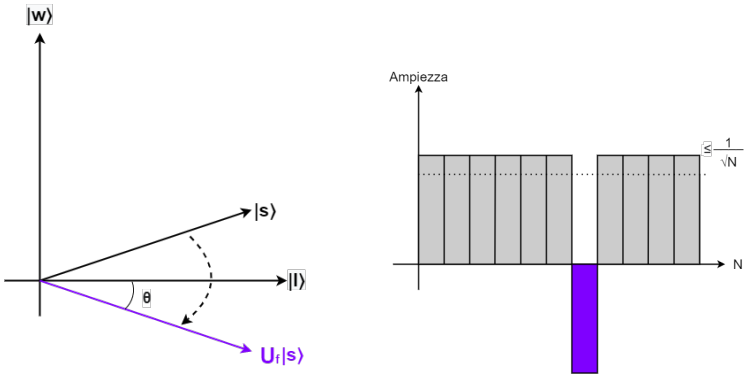


Figura 2: Riflessione rispetto a $|l\rangle$

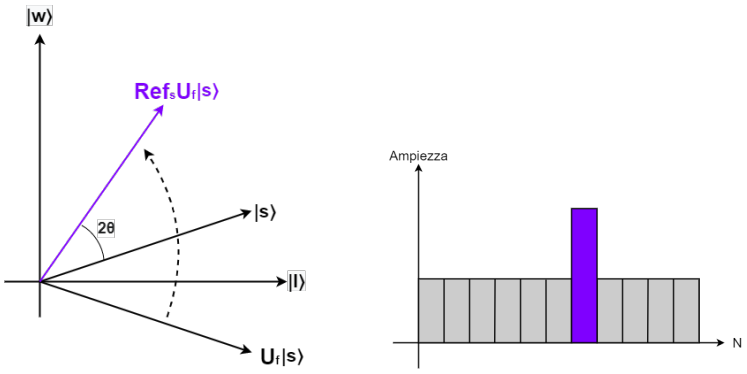


Figura 3: Riflessione rispetto a $|s\rangle$

Osserviamo che se $t_k < 1$, allora il loop interno non viene effettuato e misuriamo immediatamente: dato che questa eventualità accade solo se $\frac{\pi}{4\theta} \approx 1$, ossia quando $\theta > \frac{\pi}{4}$ (o equivalentemente quando $\frac{k}{N} \geq \frac{1}{2}$), misurare subito il vettore ci garantisce una probabilità di successo superiore a $\frac{1}{2}$ senza dover applicare nessuna riflessione. Nella maggior parte dei casi reali però questo accadrà raramente e, dunque, vogliamo dimostrare che la stessa probabilità è raggiunta dall'algoritmo di Grover.

0.2 Analisi dell'algoritmo

Sia $\theta_t \in [\theta, \frac{\pi}{2})$, con θ angolo di partenza dell'algoritmo, l'angolo che il vettore $|a\rangle$ forma con $|l\rangle$ nell'iterazione t-esima. Per come è costruito il nostro sottospazio, la prima riflessione

porta il vettore $|a'\rangle$ a un angolo $-\theta_t$ rispetto a $|l\rangle$, mentre la seconda riporta lo stato nel quadrante positivo, ottenendo un nuovo angolo pari a

$$\theta_{t+1} = -\theta_t + 2\theta + 2\theta_t = \theta_t + 2\theta$$

che equivale a effettuare una rotazione positiva di 2θ rispetto all'angolo iniziale dell'iterazione. Da questa osservazione siamo in grado di ricavare il numero di iterazioni necessarie per ottenere un buon risultato: infatti, dopo t_k iterazioni ci troveremo ad un angolo

$$\theta_{t_k} = (2t_k + 1)\theta$$

e vogliamo che tale angolo sia pari a $\frac{\pi}{2}$. Risolvendo tale equazione e tenendo a mente che t_k deve essere un numero naturale, otteniamo

$$t_k = \left\lceil \frac{\pi}{4\theta} - \frac{1}{2} \right\rceil = \left\lfloor \frac{\pi}{4\theta} \right\rfloor$$

Ora, dato che per $N >> 0$ abbiamo $\sqrt{k/N} \ll 1$, possiamo utlizzare la relazione asintotica $\theta \sim \arcsin(\theta)$ per approssimare θ come $\sqrt{k/N}$ e ottenere così che

$$t_k \approx \frac{\pi}{4}\sqrt{\frac{N}{k}}$$

In questo modo, l'angolo finale ψ si troverà nel range $\frac{\pi}{2} \pm \theta$ e, dato che $\theta \leq \frac{\pi}{4}$ per ipotesi di iterazione e la probabilità di successo $\sin^2(\psi)$ sarà sicuramente almeno pari a $\frac{1}{2}$. Abbiamo dunque dimostrato il seguente risultato:

Teorema 0.2. *Siano $f : \{0,1\}^n \longrightarrow \{0,1\}$ una funzione fattibile e $k = |S|$, con $S = \{x | f(x) = 1\}$. Allora l'algoritmo di Grover trova un elemento di S in un numero di mosse dell'ordine di $O(T)$, con $T = \sqrt{\frac{N}{k}} = 2^{\frac{n-\log_2(k)}{2}}$, e termina complessivamente in un tempo $Tn^{O(1)}$.*

0.3 Ottimizzazione dell'algoritmo: la scelta dello stato iniziale

Nel discutere l'algoritmo abbiamo implicitamente assunto di *non sapere nulla* sugli elementi cercati: per questo, abbiamo posto come vettore iniziale della nostra ricerca la sovrapposizione equiprobabile di tutti gli stati dello spazio, dato che all'inizio una qualsiasi scelta vale l'altra.

Può però a volte capitare che qualche informazione sugli elementi cercati sia disponibile e dunque perché non utilizzarla per ridurre lo spazio della ricerca del nostro algoritmo? In questo modo, oltre a velocizzare la ricerca si riducono notevolmente la quantità di gate e di risorse fisiche utilizzate, abbasando il costo computazionale. In generale, possiamo inizializzare il nostro stato usando un qualsiasi **stato simmetrico** (i.e. uno stato invariante per permutazioni dei suoi sottosistemi), come ad esempio lo stato $|GHZ\rangle$ o lo stato $|W\rangle$ generalizzati, oppure uno degli **stati di Dicke** $|Dnk\rangle$ (i.e. una sovrapposizione degli stati con distanza di Hamming pari a k).

Supponiamo, per fissare le idee, di voler risolvere un problema usando 4 qubit e di cui sappiamo che gli stati soluzione hanno una sola entrata pari a 1 in qualsiasi momento. usiamo entrambe le strategie così da confrontare lo spazio di ricerca dell'algoritmo di Grover:

- se inizializziamo tramite Hadamard gate, lo spazio di ricerca avrà dimensione pari a $N = 16$, ossia sarà l'intero spazio di Hilbert di definizione, e ci vorranno 3 iterazioni per trovare una soluzione;

- se invece sfruttiamo l'informazione che abbiamo, possiamo inizializzare lo stato iniziale come uno stato $|W\rangle$, riducendo la dimensione dello spazio di ricerca a 4 e il numero di iterazioni a 1.

Ovviamente in questo esempio a dimensioni ridotte il vantaggio non è molto spettacolare, ma ci permette di capire facilmente come inizializzare correttamente il nostro vettore iniziale possa ridurre notevolmente il costo computazionale dell'algoritmo di Grover per dimensioni più elevate.

0.4 Esempio: 2 qubit

Supponiamo di avere $N = 4$, ossia 2 qubit. In questo caso particolare ci basta una sola rotazione per trovare l'elemento cercato, dato che

$$\theta = \arcsin\left(\frac{1}{2}\right) = \frac{\pi}{6}$$

e che dopo t step si ha

$$(\mathbf{U}_j\mathbf{U}_f)^t\,|s\rangle = \sin(\theta_t)\,|w\rangle + \cos(\theta_t)\,|s'\rangle$$

con $\theta_t = (2t+1)\theta$. Quindi, poiché vogliamo $\theta_t = \frac{1}{2}$, ci basta avere $t = 1$ per ottenere il risultato voluto.

Per vedere meglio il funzionamento dell'oracolo, supponiamo che l'elemento speciale da cercare sia $|w\rangle = |11\rangle$. Allora l'oracolo di Grover corrispondente sarà

$$\mathbf{U}_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

che corrisponde al gate **CZ**. Per completare il circuito ci serve anche la riflessione rispetto alla sovrapposizione equiprobabile, ossia

$$\mathbf{Ref}_s = 2\,|s\rangle\langle s| - \mathbf{I}_4 = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}$$

Dunque, l'algoritmo proseguirà in questo modo:

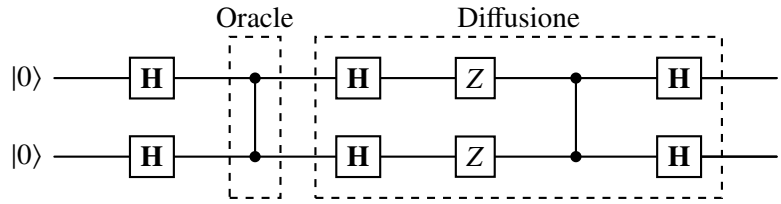
$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{\mathbb{H}} \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \xrightarrow{\mathbf{U}_f} \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix} \xrightarrow{\mathbf{Ref}_s} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

o, analogamente, in notazione braket:

$$|00\rangle \xrightarrow{\mathbb{H}} \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \xrightarrow{\mathbf{U}_f} \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle) \xrightarrow{\mathbf{Ref}_s} |11\rangle$$

proprio come ci aspettavamo.

A livello di circuiti, possiamo computare tali operazioni tramite i seguenti gate quantistici:



In particolare, la riflessione viene simulata seguendo il seguente ragionamento: supponendo di voler effettuare la nostra riflessione rispetto allo stato $|s\rangle$, ci riportiamo dapprima al vettore più semplice $|0\rangle$ e riflettiamo rispetto a quest'ultimo (applicazione della prima matrice di Hadamard $\mathbb{H}_4$). A questo punto, vogliamo che le componenti ortogonali a $|0\rangle$ del vettore che vogliamo riflettere siano invertite di segno e otteniamo ciò applicando $\mathbf{Z} \otimes \mathbf{Z}$ seguito da un **CZ**. Infine, dato che la nostra riflessione era rispetto al vettore $|s\rangle$, riapplichiamo la matrice di Hadamard per ottenere il risultato voluto. Infatti, tramite calcolo diretto otteniamo:

$$(\mathbb{H} \otimes \mathbb{H})\mathbf{CZ}(\mathbf{Z} \otimes \mathbf{Z})(\mathbb{H} \otimes \mathbb{H}) =$$

$$= \begin{bmatrix} 1/2 & 1/2 & 1/2 & 1/2 \\ 1/2 & -1/2 & 1/2 & -1/2 \\ 1/2 & 1/2 & -1/2 & -1/2 \\ 1/2 & -1/2 & -1/2 & 1/2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/2 & 1/2 & 1/2 & 1/2 \\ 1/2 & -1/2 & 1/2 & -1/2 \\ 1/2 & 1/2 & -1/2 & -1/2 \\ 1/2 & -1/2 & -1/2 & 1/2 \end{bmatrix} =$$

$$= \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} =$$

$$= 2\,|s\rangle\langle s| - \mathbf{I}_4$$

proprio come volevamo.