6 ALGORITMI QUANTISTICI

I — ALGORITMO DI GROVER PER LA RICERCA IN DATABASE

6.1.1 Algoritmi con "oracolo" o "Black Box"

Il primo passo per arrivare all'algoritmo per la ricerca in database è introdurre il concetto di *oracolo* o *Black Box* (scatola nera) in maniera analoga (ma più generica) a quella introdotta nella sezione 4.5.3.

Supponiamo di avere un database di N elementi. Invece di classificali e distinguerli in base all'informazione che contengono, a livello astratto, è utile associare ad ogni elemento un numero intero. La ricerca nel database si ridurrà quindi a trovare l'intero che soddisfa le proprietà desiderate. Per descrivere tutto il database avremmo bisogno di n bit o qubit con $N = 2^n$. Gli interi associati saranno nell'intervallo compreso fra 0 e $N-1$.

Tutta l'informazione sul problema che vogliamo risolvere è codificata in una funzione f che ha come input un intero x (o volendo una stringa di n bit) e come output un singolo bit. In altre parole, $f: \{0,1\}^{\otimes n} \to \{0,1\}$. Per definizione, $f(x) = 1$ se x è soluzione del nostro problema (ovvero è uno degli elementi che stiamo cercando) e $f(x) = 0$ se x non è soluzione del nostro problema.

Per fissare le idee, facciamo un esempio. Supponiamo di voler cercare il numero di un utente in un elenco telefonico *non ordinato* che comprende N utenti. L'elenco è costruito in modo tale da codificare per ogni utente le informazioni $\{\mathtt{Nome},\ \mathtt{Cognome}, \mathtt{Numero\ di\ telefono}\}$. Vogliamo trovare il numero di Mario Rossi. Possiamo costruire una funzione f che, per ogni utente, legga il nome e il cognome, lo confronti con il nome (Mario) e il cognome (Rossi) cercato. La funzione f darà come output 1 se nome e cognome sono quelli dell'utente cercato e 0 in tutti gli altri casi.

Lo schema è molto simile nel caso quantistico. Qui dobbiamo supporre di avere un dispositivo quantistico detto *oracolo* o *Black Box* che sia in grado di *riconoscere* la soluzione e di *segnalare o marcare* la soluzione agendo su un qubit addizionale. Prima di andare avanti, è bene chiarire il significato dei termini in italico nella frase precedente.

L'oracolo non conosce la soluzione del problema (del resto è una funzione scritta da noi che non sappiamo quale sia la soluzione) ma sa riconoscerla quando viene interrogato sottoponendogli un elemento del database. Un esempio illustrativo è la fattorizzazione dei numeri interi. Supponiamo che ci sia dato un numero intero m e che ci sia detto che è il prodotto di due numeri primi p e q: $m = p\ q$. Dobbiamo trovare quali sono p e q. L'algoritmo più usato di crittografia classica (RSA) si basa sul fatto che questo è un problema computazionalmente difficile da risolvere e che, fino ad ora, non è stato individuato nessun algoritmo *classico* che sia efficiente [1]. Il problema della fattorizzazione dei numeri primi può essere riformulato come un problema di ricerca in un database. Fra tutti i possibili input [2] dobbiamo trovare quelli per cui m è divisibile. In questo caso la funzione f implementata dall'oracolo non fa altro che prendere un intero x come input, dividere m per x e controllare se la divisione è esatta. Quindi l'oracolo non conosce la soluzione ma è in grado di verificare velocemente (mediante una divisione) se un numero è soluzione o no del problema (in questo caso, è un fattore di m) [3].

La seconda proprietà che deve avere l'oracolo è di poter *marcare* la soluzione del problema. Vediamo come questo può essere fatto in maniera relativamente semplice (si veda anche l'algoritmo di Deutch in sec. 4.5.1). Allo stato generico $|x\rangle$ associamo un quit aggiuntivo detto spesso *qubit oracolo* o *ancilla*: $|q\rangle$. Lo stato totale sarà quindi $|x\rangle\,|q\rangle$. Il qubit ancilla non porta informazione logica ma serve solo per immagazzinare l'informazione su $f(x)$, ovvero sul fatto che x sia o no soluzione del nostro problema. In maniera analoga a quanto visto nel problema si Deutch, questa operazione viene fatta usando l'addizione modulo 2 per cui l'effetto dell'oracolo è di applicare la seguente trasformazione

$$|x\rangle\,|q\rangle \xrightarrow{O} |x\rangle\,|q \oplus f(x)\rangle\,. \tag{6.1.1}$$

Nel caso più semplice, $q = 0$. Se x è soluzione, $f(x) = 1$ e $|0 \oplus 1\rangle = |1\rangle$. Al contrario, se x non è soluzione, $f(x) = 0$ e $|0 \oplus 0\rangle = |0\rangle$. Riassumendo abbiamo che

$$|x\rangle\,|0\rangle \xrightarrow{O} \begin{cases} |x\rangle\,|1\rangle & \text{se } x \text{ è soluzione} \\ |x\rangle\,|0\rangle & \text{se } x \text{ non è soluzione.} \end{cases} \tag{6.1.2}$$

In sostanza l'effetto dell'oracolo è quello di *marcare* solo gli stati soluzione associandolo ad un qubit aggiuntivo con valore 1.

L'esempio precedente è stato fatto scegliendo $q = 0$. Questo però è una scelta arbitraria. Potremmo scegliere ad esempio di inizializzare il qubit ancilla nello stato $|q\rangle = |1\rangle$. In questo caso, se x è soluzione, $f(x) = 1$ e $|1 \oplus 1\rangle = |0\rangle$ e se x non è soluzione, $f(x) = 0$ e $|1 \oplus 0\rangle = |1\rangle$. Quindi

$$|x\rangle\,|1\rangle \xrightarrow{O} \begin{cases} |x\rangle\,|0\rangle & \text{se } x \text{ è soluzione} \\ |x\rangle\,|1\rangle & \text{se } x \text{ non è soluzione.} \end{cases} \tag{6.1.3}$$

---

1 Come discusso nel Capitolo 5, esiste un algoritmo *quantistico* (di Shor) che permette di risolvere il problema della fattorizzazione degli interi in modo efficiente. [**nielsen-chuang_book**, **Rieffel2011**, **Yanofsky2008**].

2 In realtà basta cercare fra gli interi compresi fra 2 e $\sqrt{m}$ [**nielsen-chuang_book**, **Rieffel2011**, **Yanofsky2008**].

3 Questo concetto (e quindi quello dell'oracolo o della Black Box) è alla base della teoria della complessità computazionale. Si pensi, ad esempio, alla classe di problemi NP per i quali il trovare una soluzione è difficile mentre il controllo se un dato input (istanza) è soluzione o no può essere fatto velocemente (con risorse polinomiali nella dimensione dell'input) [**nielsen-chuang_book**].

Questa osservazione ci permette di studiare in altri casi dove il qubit ancilla nello stato $|q\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$. Lo stato totale si può scrivere come

$$|x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\Big[\, |x\rangle \otimes |0\rangle - |x\rangle \otimes |1\rangle \,\Big]. \tag{6.1.4}$$

Usando le equazioni (6.1.2) e (6.1.5), vediamo che se x è soluzione il bit ancilla cambia stato mentre rimane ugualse se x non è soluzione. Con questa osservazione otteniamo che l'effetto dell'oracolo è

$$\frac{1}{\sqrt{2}}\Big[\, |x\rangle \otimes |0\rangle - |x\rangle \otimes |1\rangle \,\Big] \xrightarrow{O} \begin{cases} \frac{1}{\sqrt{2}}\Big(\, |x\rangle \otimes |1\rangle - |x\rangle \otimes |0\rangle \,\Big) & \text{se } x \text{ è soluzione} \\ \frac{1}{\sqrt{2}}\Big(\, |x\rangle \otimes |0\rangle - |x\rangle \otimes |1\rangle \,\Big) & \text{se } x \text{ non è soluzione.} \end{cases} \tag{6.1.5}$$

I due stati ottenuti differiscono solo per un segno meno che possiamo fattorizzare come una fase. L'effetto dell'oracolo in questo caso è

$$|x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{O} (-1)^{f(x)}\, |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{6.1.6}$$

Anche in questo caso lo stato soluzione viene *marcato* dall'applicazione dell'oracolo. La differenza consta nella scelta dello stato iniziale del qubit ancilla che si riflette nella maniera in cui l'oracolo agisce. Se usando $|q\rangle = |0\rangle$, lo stato $|x\rangle\,|q\rangle$ viene modificato, usando $|q\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ acquista solo una fase $(-1)^{f(x)}$ mentre la struttura non viene cambiata. In quest'ultimo caso possiamo addirittura dimenticarci del qubit ancilla (visto che non viene modificato ed è uguale per tutti gli stati $|x\rangle$) e scrivere l'effetto solo sui qubit logici

$$|x\rangle \xrightarrow{O} (-1)^{f(x)}\, |x\rangle\,. \tag{6.1.7}$$

6.1.2 Algoritmo di ricerca in database (Grover)

Ora possiamo discutere l'implementazione e le performance dell'algoritmo di Grover per la ricerca in un database. Per semplicità considereremo solo il caso in cui c'è un unico stato che soddisfa i requisiti richiesti fra i possibili N elementi del database. (il problema ha un'unica soluzione). Il caso con M soluzioni è ugualmente trattabile ma richiede l'introduzione di tecniche e algoritmi più complicati [**nielsen-chuang_book**].

L'algoritmo di Grover inizia con la costruzione dello stato quantistico sovrapposizione di tutti i possibili stati logici (sec. 4.1). Se lo spazio di ricerca è costituito da N elementi che possono essere codificati in n qubit (quindi $N = 2^n$), questo può essere costruito partendo dallo stato di soli zeri $|000...0\rangle$ con l'applicazione di n porte di Hadamard (3.7.4). Il nostro stato di partenza sarà

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle\,. \tag{6.1.8}$$

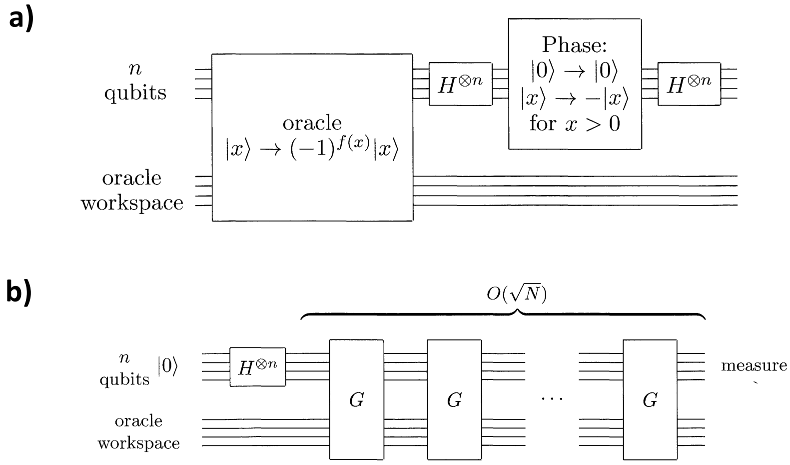

Figure 19: a) Schema circuitale per l'implementazione dell'operatore di Grover. b) Schema circuitale per l'implementazione dell'algoritmo di Grover.

Il cuore dell'algoritmo è nella costruzione di un operatore detto di *Grover* che sfrutta l'idea di oracolo come discussa nella sezione 6.1.1

Implementazione dell'operatore di Grover G**:**

1. Applicare allo stato l'oracolo (6.1.7) che cambia la fase allo stato soluzione:

$$|x\rangle \xrightarrow{O} (-1)^{f(x)} |x\rangle . \tag{6.1.9}$$

2. Applicare n porte di Hadamard.

3. Applicare un cambio di fase a tutti gli stati tranne allo stato $|000...0\rangle$:

$$|x\rangle \to -(-1)^{\delta_{x0}} |x\rangle \tag{6.1.10}$$

(dove δ_{x0} è la delta di Kronecker tale che $\delta_{ij} = 1$ se $i = j$ e $\delta_{ij} = 0$ se $i \neq j$). Ovvero, $|0\rangle \to |0\rangle$ e, per $x \neq 0$, $|x\rangle \to -|x\rangle$.

4. Applicare n porte di Hadamard.

La sequenza di operazioni per costruire l'operatore di Grover è mostrata in Figura 19 a).

Una volta costruito il circuito logico quantistico per implementare l'operatore di G, l'algoritmo per la ricerca in un database si riduce alla sua applicazione per un numero $\sqrt{N} = 2^{n/2}$ di volte come mostrato in Figura 19 b).

Stati "soluzione" e "non-soluzione".

Lo spazio logico può essere diviso in due sottospazi. Quello generato da $|x\rangle$ con x soluzione del nostro problema (ovvero quelli a cui l'oracolo cambia segno) e quelli che non sono soluzione del nostro problema. Supponiamo che ci siano M stati soluzione e, di conseguenza $N - M$ stati non-soliazione. La sovrapposizione di tutti gli stati soluzione viene indicata da un vettore $|\beta\rangle$ mentre quella degli stati non-soluzione viene indicata con $|\alpha\rangle$. In modo più formale, definiamo

$$\begin{aligned} |\alpha\rangle &= \frac{1}{\sqrt{N-M}} \sum_{x}^{non-sol} |x\rangle \\ |\beta\rangle &= \frac{1}{\sqrt{M}} \sum_{x}^{sol} |x\rangle \end{aligned} \tag{6.1.11}$$

Con questa notazione lo stato iniziale $|\psi\rangle$ in Eq. (6.1.8) si scrive come

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}\,|\alpha\rangle + \sqrt{\frac{M}{N}}\,|\beta\rangle\,. \tag{6.1.12}$$

Questo può essere verificato anche direttamente inserendo le definizioni (6.1.11) nella (6.1.12).

Operatore di Grover

Così come è stato presentato, l'operatore G e la sua azione risultano ancora misteriosi. Per rendere più concreta la trasformazione indotta scriviamo tutto in termini di operatori quantistici sfruttando la distinzione fra stati soluzione e stati non-soluzione in Eq. (6.1.11).

Come detto l'oracolo cambia segno solo agli stati soluzione, quindi cambiarà segno allo stato $|\beta\rangle$ lasciando $|\alpha\rangle$ invariato. Se lo facciamo agire su uno stato generico $a\,|\alpha\rangle + b\,|\beta\rangle$ otteniamo

$$O(a\,|\alpha\rangle + b\,|\beta\rangle) = a\,|\alpha\rangle - b\,|\beta\rangle \tag{6.1.13}$$

Nella notazione braket questo può essere scritto come

$$O = |\alpha\rangle\langle\alpha| - |\beta\rangle\langle\beta|. \tag{6.1.14}$$

Una scrittura ancora più conveniente è quella in termini di matrici nello spazio $\{|\alpha\rangle\,, |\beta\rangle\}$. Abbiamo

$$O = \begin{array}{c c} & \begin{array}{cc} |\alpha\rangle & |\beta\rangle \end{array} \\ \begin{array}{c} \langle\alpha| \\ \langle\beta| \end{array} & \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \end{array}. \tag{6.1.15}$$

Per la rimanente parte dell'operatore di Grover abbiamo l'applicazione di n porte di Hadarmard intermezzate dall'operatore che cambia di segno a tutti gli stati tranne lo stato $|\bar{0}\rangle \equiv |00...0\rangle$. Quest'ultimo nella notazione braket si scrive come

$$U = 2|\bar{0}\rangle\langle\bar{0}| - \mathrm{Id} \tag{6.1.16}$$

(dove Id è l'operatore identità). Possiamo verificare che se $|x\rangle \neq |\bar{0}\rangle$,

$$U\,|x\rangle = (2|\bar{0}\rangle\langle\bar{0}| - \mathrm{Id})\,|x\rangle = -\,|x\rangle \tag{6.1.17}$$

e che

$$U\,|\bar{0}\rangle = (2|\bar{0}\rangle\langle\bar{0}| - \mathrm{Id})\,|\bar{0}\rangle = 2\,|\bar{0}\rangle - |\bar{0}\rangle = |\bar{0}\rangle \tag{6.1.18}$$

come ci aspettavamo.

A questo punto possiamo scrivere la parte rimanente dell'operatore di Grover come

$$H^{\otimes n}UH^{\otimes n} = H^{\otimes n}(2|\bar{0}\rangle\langle\bar{0}| - \mathrm{Id})H^{\otimes n} = 2|\psi\rangle\langle\psi| - \mathrm{Id}. \tag{6.1.19}$$

L'ultimo passaggio deriva dal fatto che $H^{\otimes n}\,|\bar{0}\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle = |\psi\rangle$ e che $H^{\otimes n}\mathrm{Id}H^{\otimes n} = \mathrm{Id}$. Infatti, $H^{\otimes n}\mathrm{Id}\;H^{\otimes n} = \left(H^2\right)^{\otimes n} = \mathrm{Id}$ dato che $H^2 = \mathrm{Id}$.

Ne consegue che l'operatore di Grover può essere scritto come

$$G = (2|\psi\rangle\langle\psi| - \mathrm{Id})O. \tag{6.1.20}$$

6.1.3 Interpretazione geometrica dell'algoritmo

Lo stato $|\psi\rangle$ in Eq. (6.1.12) è normalizzato e può essere riscritto in termini di funzioni seno e coseno come

$$|\psi\rangle = \cos\frac{\theta}{2}\,|\alpha\rangle + \sin\frac{\theta}{2}\,|\beta\rangle \tag{6.1.21}$$

con

$$\begin{aligned} \cos\frac{\theta}{2} &= \sqrt{\frac{N-M}{N}} \\ \sin\frac{\theta}{2} &= \sqrt{\frac{M}{N}} \end{aligned} \tag{6.1.22}$$

Questa riscrittura ci permette di rappresentare lo stato del sistema in uno spazio bidimensionale e legarlo agli angoli come in figura 20.

I casi più importanti a livello computazionale (e più difficili da risolvere) sono quelli in cui lo spazio di ricerca è molto grande e le soluzioni sono poche; ovvero, $N \gg M$.

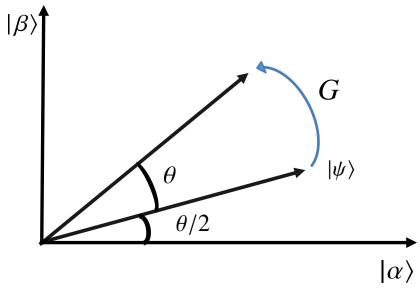

Figure 20: Rappresentazione geometrica dell'algoritmo di Grover. Il sistema è descritto in un piano bidimensionale in cui gli assi sono le proiezioni sul subspazio delle soluzioni $|\beta\rangle$ e delle non-soluzioni $|\alpha\rangle$. L'operatore di Grover induce una rotazione di un angolo θ.

Riscriviamo ora l'operatore $2|\psi\rangle\langle\psi| - \mathrm{Id}$ in termini delle funzioni seno e coseno. Dall'Eq. (6.1.21), abbiamo che $\langle\psi|\alpha\rangle = \cos\frac{\theta}{2}$ e $\langle\psi|\beta\rangle = \sin\frac{\theta}{2}$. Quindi,

$$\begin{aligned}(2|\psi\rangle\langle\psi| - \mathrm{Id})\,|\alpha\rangle &= 2\,|\psi\rangle\,\langle\psi|\alpha\rangle - |\alpha\rangle = 2\cos\frac{\theta}{2}\,|\psi\rangle - |\alpha\rangle \\ &= 2\cos\frac{\theta}{2}\left(\cos\frac{\theta}{2}\,|\alpha\rangle + \sin\frac{\theta}{2}\,|\beta\rangle\right) - |\alpha\rangle \\ &= \cos\theta\,|\alpha\rangle + \sin\theta\,|\beta\rangle\,. \qquad (6.1.23)\end{aligned}$$

Dove abbiamo usato le relazioni $2\sin\frac{\theta}{2}\cos\frac{\theta}{2} = \sin\theta$ e $2\cos^2\frac{\theta}{2} - 1 = \cos\theta$.

In maniera del tutto analoga abbiamo

$$\begin{aligned}(2|\psi\rangle\langle\psi| - \mathrm{Id})\,|\beta\rangle &= 2\,|\psi\rangle\,\langle\psi|\beta\rangle - |\alpha\rangle = 2\sin\frac{\theta}{2}\,|\psi\rangle - |\beta\rangle \\ &= 2\sin\frac{\theta}{2}\left(\cos\frac{\theta}{2}\,|\alpha\rangle + \sin\frac{\theta}{2}\,|\beta\rangle\right) - |\beta\rangle \\ &= \sin\theta\,|\alpha\rangle - \cos\theta\,|\beta\rangle\,. \qquad (6.1.24)\end{aligned}$$

Questo ci permette di scrivere l'operatore U in forma matriciale come

$$U = \begin{array}{c c} & \begin{array}{cc} |\alpha\rangle & |\beta\rangle \end{array} \\ \begin{array}{c} \langle\alpha| \\ \langle\beta| \end{array} & \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix} \end{array}. \qquad (6.1.25)$$

Ne consegue che l'operatore di Grover nello spazio $\{|\alpha\rangle, |\beta\rangle\}$ e in forma matriciale si scrive come

$$G = U\,O = \begin{array}{c c} & \begin{array}{cc} |\alpha\rangle & |\beta\rangle \end{array} \\ \begin{array}{c} \langle\alpha| \\ \langle\beta| \end{array} & \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \end{array}. \qquad (6.1.26)$$

6.1.4 Effetto dell'operatore di Grover

L'operatore di Grover nella rappresentazione (6.1.26) è immediatamente associabile ad una rotazione nel piano definito dagli stati $\{|\alpha\rangle, |\beta\rangle\}$ [4].

Per capire meglio questo punto, supponiamo di applicarlo allo stato $|\phi\rangle = \cos\delta\,|\alpha\rangle + \sin\delta\,|\beta\rangle$;

$$G\ |\phi\rangle = \begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\cdot\begin{bmatrix}\cos\delta\\ \sin\delta\end{bmatrix} = \begin{bmatrix}\cos(\theta+\delta)\\ \sin(\theta+\delta)\end{bmatrix} \tag{6.1.27}$$

dove abbiamo usato le formule $\sin(a\pm b) = \sin a\cos b \pm \cos a \sin b$ e $\cos(a\pm b) = \cos a\cos b \mp \sin a \sin b$. Quindi lo stato descritto dall'angolo δ viene ruotato di un angolo θ.

Ne consegue che se applichiamo k volte l'operatore di Grover, genereremo nello spazio $\{|\alpha\rangle, |\beta\rangle\}$ una rotazione di un angolo $k\theta$. Se lo stato iniziale è $|\psi\rangle$ in Eq. (6.1.21) (associato ad un angolo $\theta/2$) dopo k applicazioni dell'operatore di Grover avremo

$$G^k\ |\psi\rangle = \begin{bmatrix}\cos\left(\frac{2k+1}{2}\theta\right)\\ \sin\left(\frac{2k+1}{2}\theta\right)\end{bmatrix} = \cos\left(\frac{2k+1}{2}\theta\right)|\alpha\rangle + \sin\left(\frac{2k+1}{2}\theta\right)|\beta\rangle\,. \tag{6.1.28}$$

6.1.5 Performance dell'algoritmo di Grover

L'Eq. (6.1.28) ci da lo stato del sistema dopo k applicazioni dell'operatore di Grover. Affinchè l'algoritmo sia efficace deve aumentare la probabilità di misurare uno degli stati soluzione; quindi deve aumentare il coefficiente dello stato $|\beta\rangle$. Per avere la certezza di misurare uno degli stati in $|\beta\rangle$ dobbiamo avere

$$\sin\left(\frac{2k+1}{2}\theta\right) \approx 1 \tag{6.1.29}$$

che equivale ad richiedere che $\frac{2k+1}{2}\theta \approx \frac{\pi}{2}$ e quindi

$$k = \frac{1}{2}\left(\frac{\pi}{\theta} - 1\right)\,. \tag{6.1.30}$$

In altre parole, l'Eq. (6.1.30) ci fornisce il numero di iterazioni necessarie all'algoritmo di Grover per rendere molto probabile la misura di uno degli stati soluzione.

Che valore assume θ? Dall'Eq. (6.1.22) abbiamo che $\sin\frac{\theta}{2} = \sqrt{\frac{M}{N}}$. Nei casi più interessanti e difficili dove ci sono poche soluzioni, $M \ll N$. Di conseguenza, anche l'angolo $\theta/2$ dovrà essere piccolo e otteniamo

$$\sin\frac{\theta}{2} \approx \frac{\theta}{2} = \sqrt{\frac{M}{N}}. \tag{6.1.31}$$

4 Si veda, ad esempio, https://en.wikipedia.org/wiki/Rotation_matrix.

Usando questa relazione nell'Eq. (6.1.30), otteniamo

$$k = \frac{1}{2}\left(\frac{\pi}{2}\sqrt{\frac{N}{M}} - 1\right) \approx \frac{\pi}{4}\sqrt{\frac{N}{M}}. \tag{6.1.32}$$

Ricordando che per un sistema a n bit $N = 2^n$, abbiamo ottenuto che l'algoritmo di Grover riesce a trovare una soluzione corretta con $\sqrt{N} = 2^{\frac{n}{2}}$ chiamate alla funzione f. Questo è da confrontare alle $N = 2^n$ chiamate alla funzione f, per la ricerca classica in un database non strutturato. Ne consegue che l'algoritmo di Grover dà una velocizzazione (*speed-up*) quadratico rispetto agli analoghi classici.

6.1.6 Applicazione alla ricerca in un database di 4 elementi

Consideriamo la ricerca in un database di 4 elementi. Supponiamo che fra i 4 elementi ce ne sia solo uno indicato con $\bar{x}$ che soddisfi le condizioni richieste. Quindi avremo che $f(\bar{x}) = 1$ e $f(x) = 0$ se $x \neq \bar{x}$.

Per l'implementazione dell'algoritmo di Grover, è necessario specificare le caratteristiche dell'elemento che stiamo cercando e la funzione f [5]. Tuttavia per capire in senso astratto come funziona l'algoritmo di Grover e calcolare quante applicazioni sono necessarie per risolvere il problema, questo non è necessario dato che si può usare il formalismo più astratto usato nella sezione precedente.

In questo caso, abbiamo che $N = 4$ e $M = 1$. Di conseguenza,

$$\begin{aligned} \cos\frac{\theta}{2} &= \frac{N-M}{N} = \frac{\sqrt{3}}{2} \\ \sin\frac{\theta}{2} &= \frac{M}{N} = \frac{1}{2} \end{aligned} \tag{6.1.33}$$

che corrisponde ad un angolo $\theta = \pi/3$ (ovvero $\theta/2 = \pi/6$)

Possiamo scrivere lo stato iniziale in Eq. (6.1.21) come

$$|\psi\rangle = \frac{\sqrt{3}}{2}|\alpha\rangle + \frac{1}{2}|\beta\rangle = \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix} \tag{6.1.34}$$

Quindi dall'Eq. (6.1.26)

$$G = \frac{1}{2}\begin{bmatrix} 1 & -\sqrt{3} \\ \sqrt{3} & 1 \end{bmatrix}. \tag{6.1.35}$$

Applicando una sola volta l'operatore di Grover allo stato iniziale $|\psi\rangle$ otteniamo

$$G\ |\phi\rangle = \frac{1}{2}\begin{bmatrix} 1 & -\sqrt{3} \\ \sqrt{3} & 1 \end{bmatrix} \cdot \frac{1}{2}\begin{bmatrix} \sqrt{3} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |\beta\rangle\,. \tag{6.1.36}$$

Quindi con una sola applicazione dell'operatore G siamo arrivati a trovare la soluzione del problema.

5 Si noti che, come discusso in sezione 6.1.1, questo non significa conoscere la soluzione del problema ma saperla riconoscere.