Alma Mater Studiorum · Universitá di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

Implementazione di un algoritmo di ricerca quantistico

Relatore:
Prof. Cristian Degli Esposti Boschi

Presentata da:
Stefano Spadano

Anno Accademico 2017/2018

Indice

Abstract

In questa tesi si è provato ad implementare un algoritmo di ricerca quantistico per 4 qubit con l'obiettivo di comprendere a fondo le caratteristiche peculiari che oggigiorno portano la comunità scientifica a voler sviluppare questo tipo di studi e applicazioni anche a livello diffuso.

La tesi presenta quattro sezioni distinte.

Nella prima parte della tesi si trova una introduzione al concetto di computazione quantistica e al concetto di qubit non dimenticando un piccolo background storico.

Successivamente si passa a descrivere il formalismo relativo all'operatore densità dato che questa trattazione è strettamente necessaria per comprendere il rumore quantistico e come questo agisce durante la manipolazione dell'informazione quantistica.

Nella terza parte si è discusso del funzionamento di un algoritmo di ricerca quantistico e di quali vantaggi questo possiede in confronto ad un algoritmo di ricerca classico; il problema computazionale a cui si riferiscono gli algoritmi in questiono è quello della ricerca di elementi in una base di dati (database).

Nell'ultima parte infine si mostrano alcuni risultati ottenuti durante l'esperienza con il software di programmazione per computer quantistici e la relativa esecuzione sia su simulatori equivalenti classici sia su hardware genuinamente quantistici basati su qubit superconduttivi e resi disponibili dalla piattaforma IBM Quantum Experience.

Capitolo 1

Introduzione

1.1 Cenni storici

La teoria dell'informazione e della computazione quantistica può essere inquadrata come uno dei diversi campi in cui applicare i risultati della meccanica quantistica. Per informazione quantistica e computazione quantistica intendiamo, allora, lo studio dei modi per processare informazioni sfruttando sistemi quantomeccanici.

Ripercorriamo alcuni dei passi che hanno contribuito alla formulazione di questa teoria.

Negli anni '80 *Feynman* notó come vi fossero delle difficoltà nella simulazione di sistemi quantomeccanici utilizzando computer classici e suggerí l'idea di costruire computer che fossero basati sulle leggi della meccanica quantistica per aggirare queste difficoltà.

Possiamo dire che questo problema stava emergendo già negli anni '60 quando *Gordon Moore* si rese conto che il numero di transistori utilizzati nei microprocessori stava raddoppiando ogni anno circa e, riducendo progressivamente le dimensioni, effetti quantistici avrebbero iniziato ad interferire nel funzionamento dei dispositivi elettronici; anche in questo caso sembrò opportuno pensare di spostarsi ad un nuovo paradigma per la computazione: la teoria della computazione quantistica, basata sull'idea di sfruttare la meccanica quantistica per la computazione invece della fisica classica.
A questo punto viene naturale chiedersi se, e come, un computer quantistico possa risolvere problemi computazionali che non hanno una soluzione `efficiente` su un computer classico. Ricordiamo che per efficienza computazionale si intende il numero di step utili per la risoluzione del problema rispetto alla dimensione dell'input; un problema che può essere risolto efficientemente richiede un numero di step polinomiale,mentre per non efficienza ci riferiamo ai problemi che richiedono un numero di step esponenziale. Furono queste le idee che portarono alla nozione di Computer quantistico universale introdotta da *David Deutsch* sulla falsa riga di quello che fece *Alan Turing* alla fine degli anni '30 per la computazione classica. In particolare, *Deutsch* cercò di definire uno strumento in grado di simulare,in maniera efficiente, un sistema fisico arbitrario.
Questi primi passi portarono alla dimostrazione, da parte di *Peter Shor*, che un computer quantistico potesse risolvere in maniera efficiente due problemi che non avevano una soluzione efficiente su un computer classico: il problema di trovare i fattori primi per un intero ed il problema del 'logaritmo discreto'. Questi risultati furono le prime indicazioni della potenza di un computer quantistico comparata con quella di un computer classico. Tuttavia rimane ancora da approfondire quali classi di problemi possono essere risolti efficientemente utilizzando un computer quantistico e nel capitolo 3 di questa tesi si presenterà un algoritmo quantistico con i suoi vantaggi.

1.2 Il qubit

La teoria della computazione e dell'informazione quantistica si basa sul concetto di *quantum bit* , o qubit, analogo del bit classico. Come quest'ultimo infatti può avere due stati differenti che indicheremo come $|0\rangle$ e $|1\rangle$ corrispettivi degli stati classici 0 e 1. Faremo uso della notazione di *Dirac*, per indicare un vettore $|\rangle$ ed il suo duale $\langle|$, in quanto è la notazione standard per la meccanica quantistica.
La sorprendente differenza tra i due tipi di bit risiede nel fatto che i qubit possono trovarsi in uno stato che è la combinazione lineare dei due stati $|0\rangle$ e $|1\rangle$, chiamato sovrapposizione:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1.1}$$

con $\alpha, \beta \in \mathbb{C}$.

Per esempio, $|0\rangle$ e $|1\rangle$ possono rappresentare due livelli energetici in un atomo. Consideriamo $|0\rangle$ come il *ground state* e $|1\rangle$ come uno stato eccitato e immaginiamo di irradiare l'atomo con una certa energia e per un certo periodo di tempo in modo da far spostare l'elettrone da uno stato all'altro. Se ora riduciamo il tempo per cui irradiamo l'atomo, un elettrone inizialmente nello stato $|0\rangle$ può finire in uno stato a metà strada tra $|0\rangle$ e $|1\rangle$: questo stato viene di solito indicato con $|+\rangle$ ed ha la seguente forma:

$$|+\rangle = \frac{1}{\sqrt{2}}\,|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle \tag{1.2}$$

Più in generale possiamo dire che lo stato $|\psi\rangle$ rappresenta un vettore in uno spazio complesso bidimensionale ed inoltre i vettori $|0\rangle$ e $|1\rangle$ costituiscono una base ortonormale per questo spazio.

Come sappiamo in meccanica quantistica il concetto di probabilità risiede in una funzione $\psi(x)$ il cui modulo quadrato fornisce la probabilità che una particella si trovi nella posizione x. Allo stesso modo quando andiamo a misurare lo stato di un qubit come quello dell'equazione (1.1) possiamo ottenere il valore 0 con probabilità $|\alpha|^2$ ed il valore 1 con probabilità $|\beta|^2$, ed è per questo motivo che abbiamo parlato di 'metà' strada' per l'elettrone appartenente all'atomo irradiato descritto dallo stato $|+\rangle$ nell'esempio precedente. Da queste considerazioni segue la seguente relazione

$$|\alpha|^2 + |\beta|^2 = 1 \tag{1.3}$$

dal momento che la somma delle probabilità deve valere uno; da un punto di vista geometrico allora possiamo dire che il vettore che rappresenta lo stato del qubit è un vettore unitario in uno spazio complesso bidimensionale data la condizione di normalizzazione in (1.3).

É opportuno sottolineare come non sia possibile conoscere lo stato $|\psi\rangle$ del qubit, e di conseguenza i valori di α e β, bensì quello che otteniamo sarà 0 oppure 1 con le rispettive probabilità $|\alpha|^2$ e $|\beta|^2$.

La relazione (1.3) è utile anche per la rappresentazione geometrica di un singolo qubit e delle operazioni che possono esservi effettuate.

Dal momento che $|\alpha|^2 + |\beta|^2 = 1$ possiamo riscrivere l'equazione generale per lo stato di un singolo qubit nel seguente modo:

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + \exp\{(i\phi)\}\sin\frac{\theta}{2}\,|1\rangle \tag{1.4}$$

con $\phi, \theta \in \mathbb{R}$ che definiscono un punto nella sfera trimensionale con raggio unitario. Questa sfera viene usualmente chiamata *Sfera di Bloch*, presentata in 1.1 ed è uno strumento molto utile per visualizzare lo stato di un singolo qubit. Da un altro punto di

vista c'è da dire che non ci sono generalizzazioni semplici per la sfera di Bloch nel caso di diversi qubit.

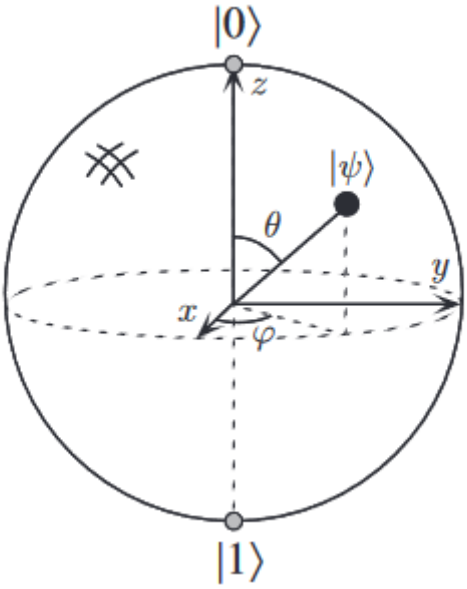

Figura 1.1: Rappresentazione di un qubit mediante la sfera di Bloch.

1.3 Qubit multipli

Proseguiamo questa introduzione sul qubit illustrando cosa succede nel caso di più qubit.

Nel caso classico sappiamo che ad ogni bit sono associate 2 possibilità, quindi avendo a disposizione n bit riusciremmo ad avere 2^n possibili risultati. Se per esempio volessimo utilizzare i bit per contare oggetti sapremmo che con n bit sarebbe possibile contare $2^n - 1$ elementi (partendo da zero).

Allo stesso modo ad un sistema di 2 qubit corrisponde una base computazionale di 4 stati,che nello specifico saranno $|00\rangle, |01\rangle, |10\rangle, |11\rangle$,come sarebbe avvenuto nel caso classico per cui a 2 bit sarebbero corrisposte le possibilità $00, 01, 10, 11$.

Formalmente è quello che ci aspettiamo poiché lo stato di due qubit appartiene ad uno spazio di Hilbert $\mathcal{H}_{AB}$ costruito dal prodotto tensoriale di due spazi di Hilbert $\mathcal{H}_A$ e $\mathcal{H}_B$; indichiamo lo stato complessivo con $|\psi\rangle_{AB} = |\psi\rangle_A \otimes |\psi\rangle_B$. In generale la dimensione di uno spazio di Hilbert che è il prodotto tensoriale di due spazi di Hilbert rispettivamente di dimensione $\texttt{dim}_A$ e $\texttt{dim}_B$ equivale a $\texttt{dim}_A \ \texttt{dim}_B$.

Come abbiamo detto in precedenza dobbiamo considerare la sovrapposizione di questi stati, motivo per cui un vettore che descrive lo stato di due qubit può essere generalmente espresso nel seguente modo:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

con ogni stato dotato della rispettiva ampiezza.

La probabilità per cui il risultato della misura dello stato $|\psi\rangle$ restituisca uno dei quattro stati, $|00\rangle, |01\rangle, |10\rangle, |11\rangle$, verrà fornita dalla rispettiva ampiezza al quadrato $|\alpha_x|^2$ con $x \in (00, 01, 10, 11)$.

La condizione di normalizzazione espressa in (1.3) per un singolo qubit, invece, diventa:

$$\sum_{x\in\{0,1\}^2} |\alpha_x|^2 = 1 \tag{1.5}$$

dove con $\{0,1\}^2$ vogliamo indicare tutte le stringhe di lunghezza 2 dove ogni carattere può valere 0 oppure 1.

Se provassimo a valutare il primo qubit dello stato $|\psi\rangle$ otterremmo 0 con probabilità $|\alpha_{00}|^2 + |\alpha_{01}|^2$ ed 1 con probabilità $|\alpha_{10}|^2 + |\alpha_{11}|^2$. Dalla meccanica quantistica inoltre sappiamo che misurare lo stato del qubit farà precipitare il qubit in quello stato,in questo modo lo stato del qubit preso in considerazione, dopo la misura, varrà:

$$|\psi'\rangle = \frac{\alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}$$

nel caso in cui ottenessimo 0, e possiamo notare come lo stato viene ri-normalizzato dal fattore $\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}$ in modo tale che continui a soddisfare la condizione di normalizzazione di uno stato quantico.

Introduciamo ora alcune considerazioni fondamentali per lo stato di due qubit. Prima di tutto diremo che uno stato si dirà *entangled* se consiste di più qubit, i quali non possono essere espressi come lista degli stati dei singoli qubit. In altre parole, se avessimo un sistema fisico C composto da due sottosistemi A e B, descritto da uno stato quantistico $|\psi\rangle_C$, diremo che questo è *separabile* se può essere espresso come prodotto degli stati dei sottosistemi: $|\psi\rangle_C = |\psi\rangle_A \otimes |\psi\rangle_B$ per una qualche scelta $|\psi\rangle_A, |\psi\rangle_B$. In caso contrario diremo che lo stato è *entangled*.

Un esempio di stato non *entangled* può essere $\frac{|00\rangle+|01\rangle}{\sqrt{2}}$ in quanto il primo qubit si trova nello stato sovrapposizione $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ ed il secondo nello stato $|0\rangle$[1]. Prendiamo in considerazione lo stato *entangled*

$$|\psi\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}$$

e osserviamo che nel momento in cui provassimo a misurare uno dei due qubit esso si comporterebbe in maniera casuale nel senso che il risultato della misura potrebbe essere allo stesso modo sia 0 che 1. Infatti, potremmo ottenere il risultato 0 con probabilità 1/2, facendo precipitare lo stato post-misura in $|\psi'\rangle = |01\rangle$, ed il risultato 1 con la stessa probabilità lasciando $|\psi'\rangle = |10\rangle$ come stato successivo alla misura. Il risultato affascinante è che una volta conosciuta la misura di uno dei due qubit saremo in grado di conoscere in maniera esatta anche il risultato della misura del restante qubit!

Torneremo nella sezione relativa alla computazione a discutere di nuovo di questi stati.

Diamo ora un'idea della potenza computazionale della natura. Consideriamo un sistema di n qubit; questo sarà descritto da una base computazione della seguente forma

$$|x_1x_2x_3...x_n\rangle$$

e per questo motivo, in generale, uno stato che descrive questo sistema necessita di 2^n ampiezze. Per $n = 500$ questo numero risulta essere molto più grande del numero stimato di atomi dell'Universo e la Natura riesce a manipolare quantità così grandi di dati per sistemi che contengono solo pochi atomi!

[1]Il primo qubit viene considerato quello più a destra in questa trattazione.

1.4 Computazione quantistica

Come sappiamo all'interno di un computer troviamo tanti circuiti elettrici costituiti da fili e porte logiche, le quali implementano le funzioni logiche elementari AND, OR e NOT. In maniera analoga un computer quantistico sarà realizzato partendo da circuiti quantistici, a loro volta costituiti da fili, o bus, e porte logiche quantistiche; i fili servono a trasportare l'informazione lungo il circuito e le porte logiche per manipolarla.

In generale, per computazione quantistica intendiamo la descrizione dei cambiamenti che avvengono ad un determinato stato.

Per la realizzazione di un elaboratore quantistico, nel 2000, sono stati individuati i *criteri di DiVincenzo* che elenchiamo di seguito:

- Possibilità di inizializzare e leggere gli elementi in cui viene immessa l'informazione quantistica, i qubit;
- esistenza di un insieme universale di porte logiche quantistiche per l'elaborazione dell'informazione, come avveniva per le porte logiche classiche AND,OR e NOT;
- i tempi su cui agiscono le porte devono essere abbastanza piccoli rispetto a quelli di decoerenza, ossia i tempi in cui uno stato perde le proprietà essenziali quantistiche su cui era stato preparato;
- scalabilità, per integrare il maggior numero possibile di memorie, canali e porte in uno spazio limitato.

Tra le porte logiche classiche che agiscono su di un solo bit troviamo unicamente il NOT, che come sappiamo agisce sostituendo 0 con 1 e viceversa. Per ottenere un rispettivo quantistico ci servirebbe qualcosa che convertisse lo stato $|0\rangle$ con lo stato $|1\rangle$. Viene spontaneo chiedersi allora come si comporterebbe nel caso di uno stato che è la sovrapposizione dei due stati $|0\rangle$ e $|1\rangle$. Scopriamo che il quantum NOT gate agisce in maniera lineare, di conseguenza se agisse sullo stato

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

questo restituirebbe

$$|\psi\rangle = \beta\,|0\rangle + \alpha\,|1\rangle$$

invertendo, di fatto, i coefficienti che si trovano davanti i due stati che costituiscono la base computazionale.

Risulta a questo punto utile provare a rappresentare le porte logiche mediante l'uso di matrici in modo da poter esplorare anche sotto quali condizioni algebriche è lecito parlare di porta logica quantistica.

Possiamo rappresentare la porta logica NOT con la seguente matrice

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

allora l'azione dell'operatore X sullo stato $|\psi\rangle$ sarà

$$X\,|\psi\rangle = X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

. Le porte logiche quantistiche che agiscono su di un singolo qubit sono descritte da matrici 2×2 e, in particolare, queste dovranno essere *unitarie*. Ricordiamo che una

matrice si dice unitaria quando vale la condizione $UU^\dagger = \mathrm{I}$, dove $U^\dagger$ indica l'*aggiunto* di U che si ottiene trasponendo righe e colonne della matrice U e poi prendendo il suo complesso coniugato. Questa risulta essere l'unica condizione necessaria per parlare di porta logica quantistica al contrario di quanto accadeva per la computazione classica dove l'unica porta logica per un singolo bit è il NOT.

Tra le porte logiche quantistiche per qubit singoli troviamo anche

$$Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

che ,rispettivamente, cambiano il segno di $|1\rangle$ e trasformano gli stati $|0\rangle$ e $|1\rangle$ in due nuovi stati $|+\rangle$ e $|-\rangle$. Infatti, notiamo come la porta Z lasci invariato $|0\rangle$ e inverta il segno di $|1\rangle$ restituendo $-|1\rangle$; per quanto riguarda H, questa si chiama *Hadamard gate*, e la sua azione consiste nel trasformare lo stato $|0\rangle$ nello stato $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, e trasformare $|1\rangle$ in $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. Dal momento che H è tra le porte logiche più utilizzate possiamo provare a visualizzare il suo comportamento anche mediante la sfera di Bloch.

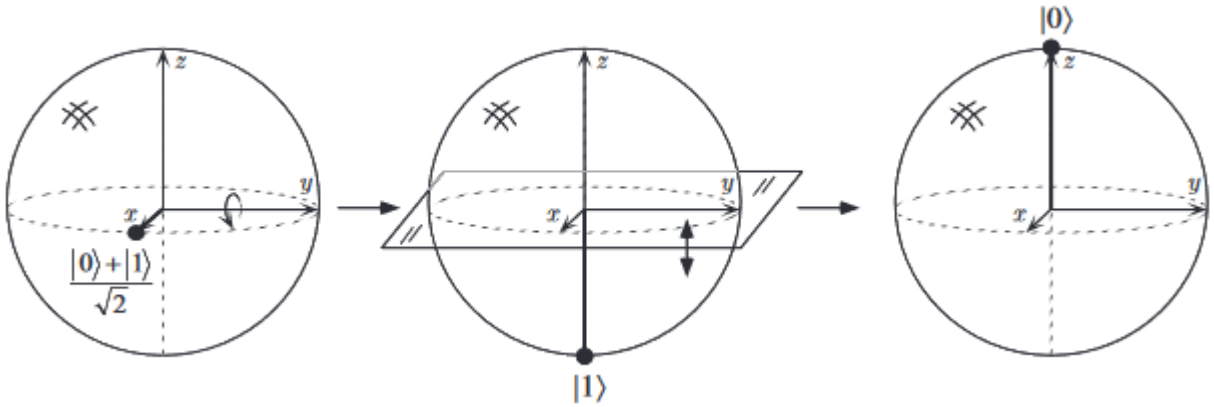

Figura 1.2: Azione di H su un qubit che si trova inizialmente nello stato $|+\rangle$.

In figura 1.2, dobbiamo innanzitutto notare che applicare due volte l'operatore H ad uno stato, non farà altro che lasciarlo invariato dal momento che H^2=I. Successivamente osserviamo che applicare H vuol dire, per prima cosa, ruotare la sfera attorno all'asse $\hat{y}$ del sistema di riferimento di 90° e per seconda, ruotare la sfera attorno all'asse $\hat{x}$ di 180°. In questo modo si ottiene lo stato $|0\rangle$ ed è quello che ci aspettavamo in quanto abbiamo applicato H allo stato $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ che equivale ad applicare due volte H allo stato $|0\rangle$.

1.4.1 Porte logiche per qubit multipli

Il prototipo di porta logica a diversi qubit è il *controlled-not*, o CNOT; questa porta presenta due qubit in input definiti come *control qubit* e *target qubit*. Il CNOT agisce controllando il valore del primo qubit: se questo vale 0 allora il secondo qubit viene lasciato nel suo stato, se invece il primo qubit vale 1 allora al secondo qubit viene applicato il NOT. In pratica:

$$|00\rangle \longrightarrow |00\rangle \; |01\rangle \longrightarrow |01\rangle \; |10\rangle \longrightarrow |11\rangle \; |11\rangle \longrightarrow |10\rangle \, . \tag{1.6}$$

Da un altro punto di vista possiamo dire che il CNOT costituisce una generalizzazione dell'XOR(*exclusive-or*) classico in quanto la sua azione complessivamente si esprime nel seguente modo:

$$|A, B\rangle \longrightarrow |A, B \oplus A\rangle \tag{1.7}$$

dove $\oplus$ indica l'azione dell'XOR. In questo modo si sostituisce al target qubit l'azione dell'XOR tra il control qubit ed il target qubit.

Mostriamo la matrice corrispondente alla porta logica quantistica CNOT notando che anche in questo caso abbiamo una matrice unitaria a rappresentarla (condizione necessaria per conservare la probabilità).

$$U_{cn} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Notiamo che se avessimo un bit bersaglio a 0 in ingresso ed un bit di controllo x, sia il bit di controllo che quello bersaglio verrebbero posti in uscita uguali al valore x dato che $x \oplus 0 = x$ andando di fatto a copiare il bit di controllo. Possiamo chiderci se un simile comportamento possa essere ripetuto in maniera generale per uno stato $|x\rangle = \alpha |0\rangle + \beta |1\rangle$. Di fatti dal momento che anche il CNOT agisce in maniera lineare possiamo scrivere che

$$|x, y = 0\rangle = \alpha |0\rangle |0\rangle + \beta |1\rangle |0\rangle \Longrightarrow \alpha |0\rangle |0\rangle + \beta |1\rangle |1\rangle$$

ed in questo modo abbiamo ottenuto una copia del qubit di controllo nei casi in cui vale 0 ($\alpha = 1, \beta = 0$) e 1 ($\alpha = 0, \beta = 1$).

Questo tuttavia non è un discorso valido in generale e non è possibile copiare lo stato arbitrario di un qubit mediante l'utilizzo del CNOT come afferma il

Teorema 1.4.1 *della non-clonabilità: Non può esistere un dispositivo che produce due copie esatte dello stato generico quantistico* $|\psi\rangle$.

Infine ci soffermiamo sul fatto che non tutte le porte logiche quantistiche possono avere un corrispettivo quantistico in termini di matrici unitarie. Per esempio, l'XOR non può avere una simile rappresentazione in quanto è irreversibile o non invertibile. Ci spieghiamo meglio, se avessimo un output $A \oplus B$ noi non potremmo essere in grado di risalire rispettivamente ai due input A e B; concludiamo che questo processo è accompagnato da una perdita di informazione intrinseca e sappiamo che invece le porte logiche quantistiche rappresentate da matrici unitarie sono sempre invertibili.

1.4.2 Un esempio di circuito

Consideriamo un circuito che ha come input i quattro stati che costituiscono una base computazionale per 2 qubit, rispettivamente $|00\rangle, |01\rangle, |10\rangle, |11\rangle$; a questi viene applicata la porta di Hadamard,trasformando,per esempio

$$|00\rangle \longrightarrow \frac{(|0\rangle + |1\rangle)|0\rangle}{\sqrt{2}}$$

successivamente viene applicato il CNOT, invertendo il secondo qubit solo quando il primo qubit risulta valere 1.

Avremo quattro stati in output, i quali prendono il nome di stati di Bell oppure *EPR pairs* in onore dei primi che studiarono le strane proprietà di questi stati (Bell, Einstein, Podolsky, Rosen).

Concludiamo con una notazione utile per ricordarli, i quattro stati espliciti e la rappresentazione grafica, in figura 1.3, del circuito discusso in precedenza:

$$|\beta_{xy}\rangle \equiv \frac{|0, y\rangle + (-1)^x |1, \overline{y}\rangle}{\sqrt{2}} \tag{1.8}$$

$$|\beta_{00}\rangle \equiv \frac{(|00\rangle+|11\rangle)}{\sqrt{2}}$$
$$|\beta_{01}\rangle \equiv \frac{(|01\rangle+|10\rangle)}{\sqrt{2}}$$
$$|\beta_{10}\rangle \equiv \frac{(|00\rangle-|11\rangle)}{\sqrt{2}}$$
$$|\beta_{11}\rangle \equiv \frac{(|01\rangle-|10\rangle)}{\sqrt{2}}$$

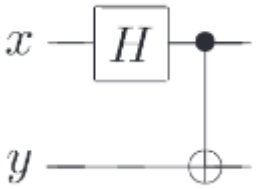

Figura 1.3: Circuito per creare stati di Bell. Troviamo prima una porta di Hadamard e successivamente un CNOT (control qubit in x e target qubit in y).

Capitolo 2

Il problema della decoerenza

Come si è accennato in precedenza nei criteri di DiVincenzo, uno dei temi di cui bisogna tenere conto quando si parla di computazione quantistica è quello della decoerenza. In realtà, il concetto di decoerenza è radicato nella meccanica quantistica ancora prima che nella computazione quantistica nel senso che si inizia a parlare di decoerenza ogni volta che si trattano sistemi non isolati. L'equazione di *Schrödinger* ci spiega come evolve nel tempo uno stato isolato $|\psi\rangle$ da un tempo t_0 fino a t

$$|\psi(t)\rangle = U(t_0, t)\,|\psi(t_0\rangle$$

dove U è un operatore unitario; se questo non fosse isolato, come può avvenire per esempio durante una misura, allora si direbbe entangled con l'ambiente portando ripercussioni sulla coerenza della funzione d'onda. A questo punto non si parlerà più di stati puri ma di miscela statistica.

Cerchiamo di capire cosa significa lasciar agire la decoerenza. Consideriamo uno stato $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ che per quanto detto fino ad ora presenta una probailitá $|\alpha|^2$ per cui lo stato si trovi nello stato 0 ed una probabilitá $|\beta|^2$ per cui si trovi nello stato 1. Una volta che la decoerenza ha agito, ai fini statistici ritroviamo le medesime proabilitá $|\alpha|^2$ e $|\beta|^2$ ma in ogni singola misura, prima della misura stessa, i valori 0 oppure 1 sono assegnati con frequenza classica $|\alpha|^2$, $|\beta|^2$. Possiamo dire che in questo modo si campiona la statistica dei valori di un qubit che però sono sempre o $|0\rangle$ oppure $|1\rangle$ e non una loro sovrapposizione. Gli stati $|0\rangle$ e $|1\rangle$ si manifestano in una base preferenziale che è quella osservata classicamente.

Per un computer quantistico le interazioni non gradite sono costituite principalmente da rumore e questo viene generato dalle interazioni con l'ambiente sia nel caso classico che nel caso quantistico.

Immaginiamo che un bit sia immagazinato su di un hard disk appartenente ad un computer; il bit si trova nello stato 0 oppure nello stato 1 ma dopo diverso tempo un campo magnetico casuale può causare un cambiamento di stato nel bit. Classicamente possiamo modellare questo fenomeno nel seguente modo: assegnamo al bit una probabilità p che esso rimanga nel suo stato, ed una probabilità 1-p che esso cambi il suo stato. In termini di equazioni avremo

$$\begin{bmatrix} q_0 \\ q_1 \end{bmatrix} = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \end{bmatrix}$$

con p_0 e p_1 che rappresentano le probabilità che il bit inizialmente si trovi nello stato 0 oppure nello stato 1 e q_0 e q_1 invece rappresentano le probabilità successive all'azione del rumore. In forma compatta possiamo scrivere

$$\vec{q} = E\vec{p} \tag{2.1}$$

dove E$\vec{p}$ deve essere una distribuzione di probabilità valida, ovvero devono valere due condizioni sulla matrice E: prima, le componenti di E devono essere tutte positive(altrimenti sarebbe possibile ottenere probabilità negative) e seconda, la somma delle componenti di ogni colonna deve valere 1(condizione di completezza).

Nel caso quantistico sarà necessario introdurre l'operatore densità per avere un corrispettivo al caso classico.

2.1 Operatore densità

É possibile ottenere una formulazione della meccanica quantistica equivalente a quella con i vettori di stato sfruttando la matrice densità o operatore densità; in questo modo possiamo estendere le nostre considerazioni anche a quei casi in cui il sistema non sarà descritto da un vettore di stato unico. Supponiamo che un sistema quantistico si trovi in uno stato $|\psi_i\rangle$ con una probabilità rispettiva p_i di trovarsi proprio in quello stato. Chiameremo allora $\{p_i, |\psi_i\rangle\}$ un *ensemble* di stati puri; in questo modo definiamo l'operatore densità mediante l'equazione

$$\rho \equiv \sum_i p_i \, |\psi_i\rangle \, \langle\psi_i| \tag{2.2}$$

Per riformulare i postulati della meccanica quantistica in termini dell'operatore densità dobbiamo prima elencare alcune sue proprietà:

- ρ deve avere una traccia uguale ad 1;
- ρ deve essere un operatore positivo.

Se ρ soddisfa queste due condizioni allora è l'operatore densità associato all'ensemble $\{p_i, |\psi_i\rangle\}$. Inoltre ricordiamo che $\rho = \rho^\dagger$ dato che le $p_i \in \mathbb{R}$.

A questo punto possiamo elencare i postulati della meccanica quantistica formulati mediante l'operatore densità:

`Postulato 1`: Ogni sistema fisico può essere associato ad uno spazio vettoriale complesso dotato di prodotto scalare (uno spazio di Hilbert) chiamato spazio degli stati del sistema fisico. Il sistema è interamente descritto dal suo operatore densità, che a sua volta è un operatore positivo con traccia uguale ad uno, che agisce sullo spazio degli stati del sistema. Se un sistema quantistico si trova nello stato ρ_i con probabilità p_i, allora l'operatore densità per il sistema sarà $\sum_i \rho_i p_i$.

`Postulato 2`: L'evoluzione di un sistema quantistico chiuso(isolato) è descritta da una trasformazione unitaria. Lo stato ρ del sistema al tempo t_1 è collegato allo stato ρ' al tempo t_2 mediante un operatore unitario U che dipende esclusivamente dagli istanti t_1 e t_2,

$$\rho' = U\rho U^\dagger$$

`Postulato 3`: Esistono degli operatori $\{M_m\}$ chiamati operatori di misura che descrivono le operazioni di misura quantistica. Questi operatori agiscono sullo spazio degli stati del sistema misurato. Il pedice m indica il risultato che può scaturire dalla misurazione. Se lo stato del sistema vale ρ prima della misura allora la probabilità che il risultato m avvenga è data dall'equazione

$$p(m) = tr(M_m^\dagger M_m \rho)$$

e lo stato successivo alla misura, invece

$$\frac{M_m \rho M_m^\dagger}{tr(M_m^\dagger M_m \rho)}$$

Inoltre, gli operatori di misura soddisfano la condizione di completezza

$$\sum_m M_m^\dagger M_m = I$$

`Postulato 4`: Lo spazio degli stati di un sistema fisico composto da più sistemi fisici è generato dal prodotto tensoriale degli spazi degli stati dei sistemi fisici che compongono il sistema fisico totale. In questo modo lo stato totale del sistema avrá la struttura

$$\rho_1 \otimes \rho_2 \otimes ...\rho_n$$

oppure loro combinazioni lineari.

2.1.1 Operatore densità ridotto

La vera potenza dell'operatore densità si osserva durante l'analisi di sistemi composti da più sottosistemi; in questi casi si parla allora di operatore densità ridotto.

Supponiamo di avere due sistemi fisici A e B, il cui stato è descritto dall'operatore densità ρ_{AB}. Definiamo l'operatore densità ridotto nel seguente modo

$$\rho_A \equiv tr_B(\rho_{AB}) \tag{2.3}$$

dove tr_B indica traccia parziale operata sul sistema B. Applicare la traccia parziale significa calcolare su ogni singolo termine

$$tr_B(|a_1\rangle \langle a_2| \otimes |b_1\rangle \langle b_2|) \equiv |a_1\rangle \langle a_2| \, tr(|b_1\rangle \langle b_2|)$$

con $|a_1\rangle, |a_2\rangle$ vettori appartenti allo spazio degli stati A, e $|b_1\rangle, |b_2\rangle$ vettori appartenenti allo spazio degli stati B.

Ricordiamo che dalla meccanica quantistica abbiamo che $tr(|b_1\rangle \langle b_2|) = \langle b_1|b_2\rangle$.

Ritorniamo a sfruttare lo stato di Bell $|+\rangle$ e mostriamo come agisce l'operatore densità ridotto su questo stato. L'operatore densità per $|+\rangle$ vale

$$\rho = (\frac{|00\rangle + |11\rangle}{\sqrt{2}})(\frac{\langle 00| + \langle 11|}{\sqrt{2}}) = \frac{(|00\rangle \langle 00| + |11\rangle \langle 00| + |00\rangle \langle 11| + |11\rangle \langle 11|)}{2}$$

Se adesso calcoliamo la traccia del secondo qubit, troviamo l'operatore densità ridotto per il primo qubit,

$$\rho^1 = tr_2(\rho)$$

$$\frac{tr_2(|00\rangle \langle 00|) + tr_2(|11\rangle \langle 00|) + tr_2(|00\rangle |11\rangle) + tr_2(|11\rangle \langle 11|)}{2}$$

$$\frac{|0\rangle \langle 0| \langle 0|0\rangle + |1\rangle \langle 0| \langle 0|1\rangle + |0\rangle \langle 1| \langle 1|0\rangle + |1\rangle \langle 1| \langle 1|1\rangle}{2}$$

a questo punto dato che $\langle i|j\rangle = 0$ per $i \neq j$ rimane

$$\frac{|0\rangle \langle 0| + |1\rangle \langle 1|}{2} = \frac{I}{2}$$

In casi come questi parliamo di stati *misti* dato che $tr((\frac{I}{2})^2) = \frac{1}{2} < 1$ e invece parleremo di stati *puri* nel caso in cui $tr(\rho^2) = tr(\rho) = 1$ dato che $\rho^2 = \rho$.

Abbiamo trovato un risultato davvero interessante allora in quanto dai conti svolti in precedenza risulta che lo stato dei qubit insieme costituisce uno stato puro, cioè uno stato che siamo in grado di conoscere perfettamente; lo stato del primo qubit dall'altro lato invece si trova in uno stato misto che quindi non conosciamo del tutto. Questo risultato è una delle tante conseguenze del fenomeno dell'*entanglement*.

2.2 Esempi di rumore quantistico

Descriviamo gli stati quantistici in termini di operatore densità e, come avveniva classicamente per l'equazione (2.1), gli stati si trasformano seguendo l'equazione:

$$\rho' = \xi(\rho) \tag{2.4}$$

dove con ξ indichiamo una *operazione quantistica*. Esempi di operazioni quantistiche possono essere la trasformazione unitaria, per cui $\xi(\rho) = U\rho U^\dagger$, e la misura, per cui $\xi_m(\rho) = M_m \rho M_m^\dagger$. In generale, diremo che un'operazione quantistica mostra l'evoluzione di uno stato ρ soggetto a trasformazioni fisiche, portandolo nello stato finale $\xi(\rho)$.

Possiamo immaginare una trasformazione unitaria come un box che accetta uno stato di input e restituisce uno stato di output, senza preoccuparci di quello che accade dentro al box. Per il momento abbiamo parlato di sistemi chiusi, o isolati, e di come essi evolvono mediante trasformazioni unitarie; un modo conveniente per trattare sistemi aperti risulta quello di considerarli come il risultato dell'interazione del sistema, che ci interessa studiare, con l'ambiente. Se a questo punto consideriamo i due sistemi insieme, otteniamo un sistema chiuso: abbiamo quindi uno stato di input costituito dal prodotto tensoriale dei due stati $\rho \otimes \rho_{amb}$.

Dopo che la trasformazione U interna al box ha agito, possiamo operare la traccia parziale sull'ambiente in modo da ottenere lo stato ridotto, relativo al sistema che ci interessa, da solo:

$$\xi(\rho) = tr_{amb}[U(\rho \otimes \rho_{amb})U^\dagger] \tag{2.5}$$

Assumere che l'ambiente ed il nostro sistema si trovino in uno stato che è il prodotto tensoriale dei due stati è un'assunzione ragionevole in diversi casi di interesse. Per esempio, se decidessimo di preparare uno stato quantistico in uno stato specifico ci dovremmo preoccupare di eliminare le correlazioni tra il sistema e l'ambiente in modo da ottenere uno stato puro.

Generalizziamo la definizione di operazione quantistica in modo da poter utilizzare diversi spazi di input e output. Immaginiamo di avere un qubit singolo, che indichiamo con A, in un certo stato ρ accompagnato da un altro qubit, che indichiamo con B, nello stato $|0\rangle\langle 0|$. I due sistemi interagiscono mediante una trasformazione unitaria U, facendo evolvere il sistema nello stato $U(\rho \otimes |0\rangle\langle 0|)U^\dagger$; escludiamo ora il sistema A, lasciando il sistema B in un certo stato ρ'. L'operazione quantistica, per definizione, che descrive questa trasformazione è la seguente:

$$\xi(\rho) = \rho' = tr_a(U(\rho \otimes |0\rangle\langle 0|)U^\dagger) \tag{2.6}$$

Definiamo allora le operazioni quantistiche come la classe di funzioni che trasformano ρ in ρ' mediante il processo appena descritto.

Possiamo riscrivere ulteriormente l'equazione (2.5) in termini di operatori che agiscono solo sulla spazio di Hilbert del sistema principale. Supponiamo di avere una base

ortonormale per lo spazio finito dimensionale dell'ambiente $|e_k\rangle$ ed uno stato iniziale per l'ambiente $\rho_{amb} = |0\rangle\langle 0|$[1].

L'equazione (2.5) allora diventa

$$\xi(\rho) = \sum_k \langle e_k| U \Big[\rho \otimes |e_0\rangle\langle e_0|\Big] U^\dagger |e_k\rangle \tag{2.7}$$

$$= \sum_k E_k \rho E_k^\dagger \tag{2.8}$$

dove $E_k \equiv \langle e_k|U|e_0\rangle$ è un operatore sullo spazio degli stati del sistema principale,gli operatori E_k sono chiamati elementi per l'operazione quantistica ξ e questa equazione definisce l'*operator-sum representation* per ξ. Gli elementi dell'operazione ξ soddisfano una condizione di completezza per cui la traccia di $\xi(\rho)$ deve valere uno:

$$1 = tr(\xi(\rho)) = tr(\sum_k E_k^\dagger \rho E_k) = tr(\sum_k E_k^\dagger E_k \rho)$$

e dal momento che questa relazione è valida per tutte le ρ possiamo affermare che

$$\sum_k E_k E_k^\dagger = I.$$

In conclusione, l'*operator-sum representation* costituisce un modo per descrivere la dinamica del sistema principale senza dover considerare proprietà del sistema ambiente.

Possiamo visualizzare il comportamento di alcune operazioni quantistiche in termini della loro azione sulla sfera di Bloch. Una matrice densità per uno stato misto può essere sempre scritta come

$$\rho = I + \frac{\vec{r}\cdot\vec{\sigma}}{2}$$

con $\vec{r}$ che indica un vettore tridimensionale tale che $||\vec{r}|| \leq 1$. Questo vettore è noto come vettore di Bloch per lo stato ρ. Notiamo che in questo modo uno stato si dirà puro se e solo se $||\vec{r}|| = 1$; questa conclusione deriva dal fatto che se andassimo a cercare gli autostati di ρ troveremmo autovalore 1 ed autovettore $|\psi_r\rangle$ tale che $(\hat{r}\cdot\vec{\sigma})|\psi_r\rangle = |\psi_r\rangle$.

Esplicitiamo allora l'equazione precedente per ottenere

$$\rho = \frac{1}{2}\begin{bmatrix} 1+r_z & r_x - ir_y \\ r_x + ir_y & 1-r_z \end{bmatrix}. \tag{2.9}$$

2.2.1 Canali di bit flip & phase flip

Parliamo di bit flip quando vengono scambiati tra di loro gli stati $|0\rangle$ e $|1\rangle$ con probabilità 1-p. In questo caso gli elementi di questa operazione quantistica sono

$$E_0 = \sqrt{p}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$E_1 = \sqrt{1-p}\sigma^x = \sqrt{1-p}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

[1]Non perdiamo di generalità nel considerare lo stato iniziale dell'ambiente in uno stato pure, dato che se si trovasse in uno stato misto, potremmo sempre inserire un ulteriore sistema in grado di riportare il sistema in uno stato puro.

Per phase-flip invece intendiamo un canale con i seguenti elementi

$$E_0 = \sqrt{p}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

$$E_1 = \sqrt{1-p}\sigma^z = \sqrt{1-p}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$$

Come caso speciale di phase-flip possiamo considerare quello per cui p=1/2 ed andare a scrivere l'operazione quantistica come

$$\rho \longrightarrow \xi(\rho) = P_0\rho P_0 + P_1\rho P_1$$

Con $P_0 = |0\rangle\langle 0|$ e $P_1 = |1\rangle\langle 1|$ ci accorgiamo che corrisponde a compiere una misura sul qubit nella base $|0\rangle, |1\rangle$ con il risultato della misura sconosciuto. In questo senso si tratta di proiettare il vettore di Bloch lungo l'asse z della sfera, perdendo le componenti relative ad x ed y.

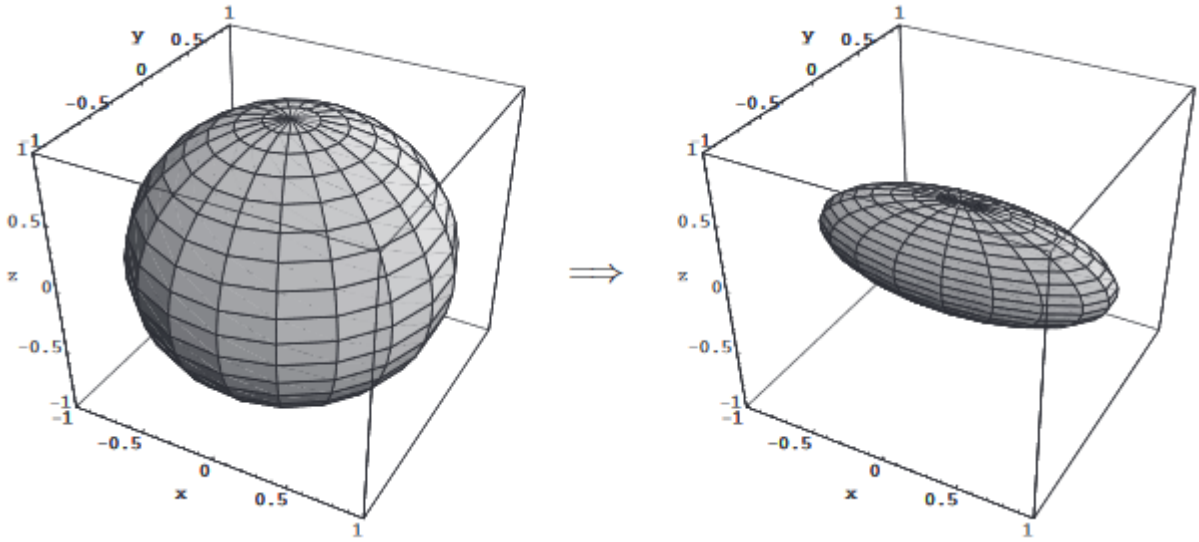

Figura 2.1: Effetto di un canale di bit flip sulla sfera di Bloch per p=0.3. C'è la proiezione del vettore lungo l'asse $\hat{x}$ con la contrazione del piano $\hat{y} - \hat{z}$.

Esiste infine anche il bit-phase flip dotato dei seguenti elementi:

$$E_0 = \sqrt{p}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

$$E_1 = \sqrt{1-p}\sigma^y = \sqrt{1-p}\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}$$

Come indica il nome in questo caso abbiamo una combinazione delle due precedenti situazioni, di fatto se notiamo $\sigma^y = i\sigma^x\sigma^z$.

Le figure 2.1,2.2 e 2.3 visualizzano in termini geometrici l'azione sulla sfera di Bloch dotata di operatore densità $\rho = \frac{1+\hat{r}\cdot\vec{\sigma}}{2}$ con operazione quantistica $\xi(\rho) = \frac{1+\vec{\xi}\cdot\vec{\sigma}}{2}$, quindi $\xi(\rho)$ parametrizzato da $\vec{r}$, $\vec{\xi} = \vec{\xi}(\vec{r})$

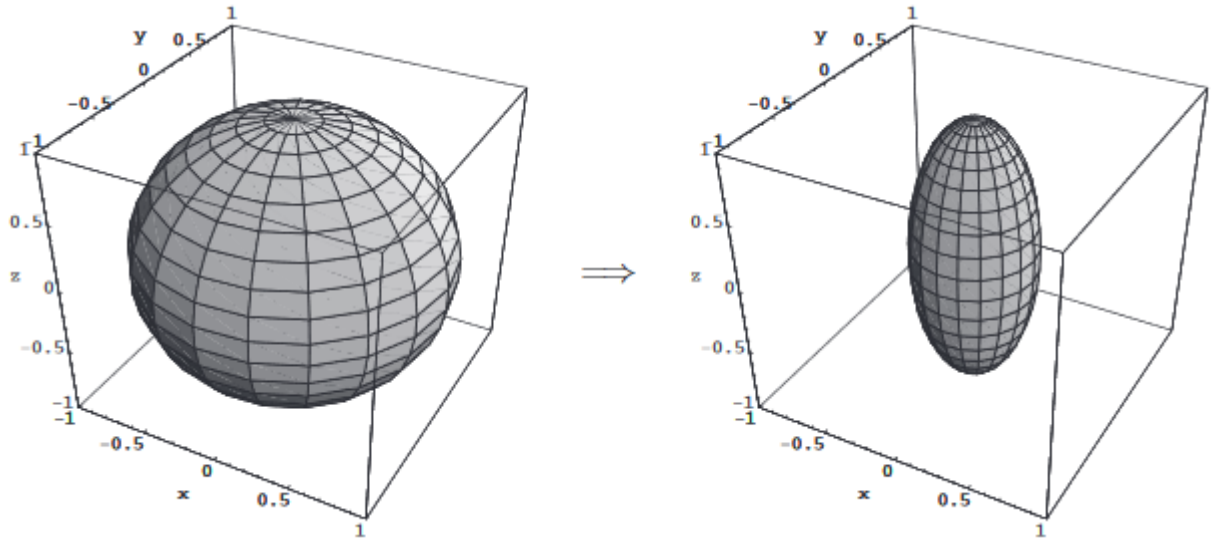

Figura 2.2: Effetto di un canale di phase flip per p=0.3.

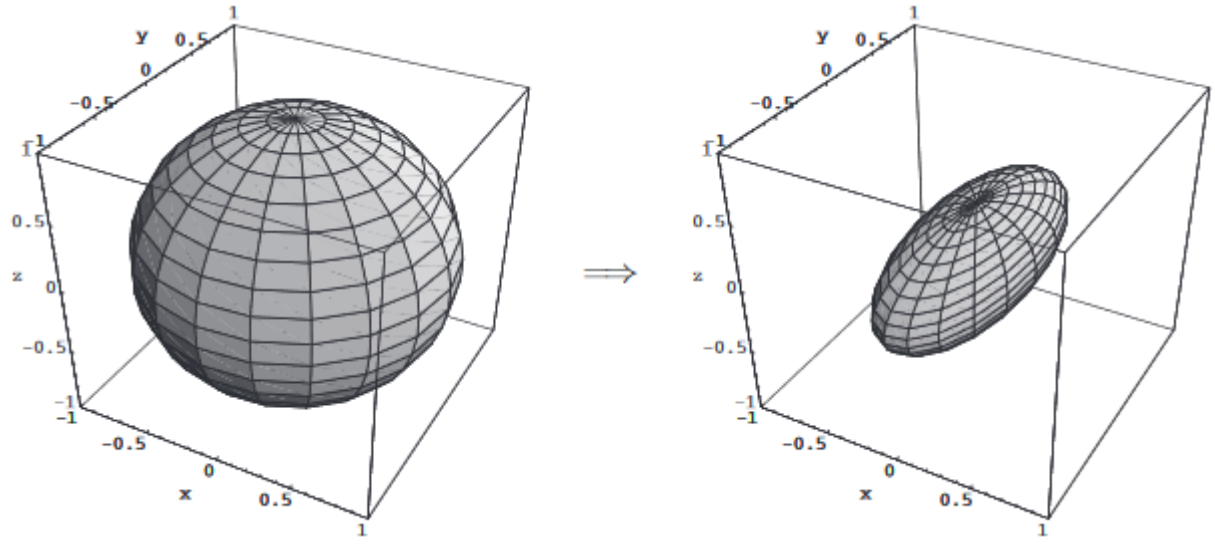

Figura 2.3: Effetto di un canale di bit-phase flip per p=0.3.

2.2.2 Canali depolarizzanti

Parliamo di depolarizzazione quando il nostro qubit con una probabilità p viene rimpiazzato da uno stato misto e senza informazioni di rilievo come I/2, con una probabilità 1-p invece viene lasciato così com'è. Dopo che il rumore ha agito abbiamo il seguente stato:

$$\xi(\rho) = \frac{pI}{2} + (1-p)\rho$$

Abbiamo un circuito, in figura 2.4, che può darci l'idea di come funziona il canale di depolarizzazione: In questo circuito sulla prima linea troviamo il nostro sistema di input

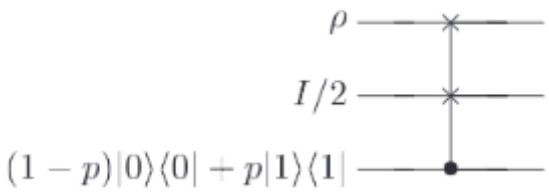

Figura 2.4: Implementazione di un circuito per realizzare depolarizzazione.

e le altre due linee rappresentano invece il canale. Il terzo qubit si trova in uno stato misto di $|0\rangle$ ed $|1\rangle$ ed agisce come control per scambiare o meno il primo qubit con il secondo.

Anche in questo caso osserviamo in figura 2.5 l'azione, in termini geometrici, sulla sfera di Bloch.

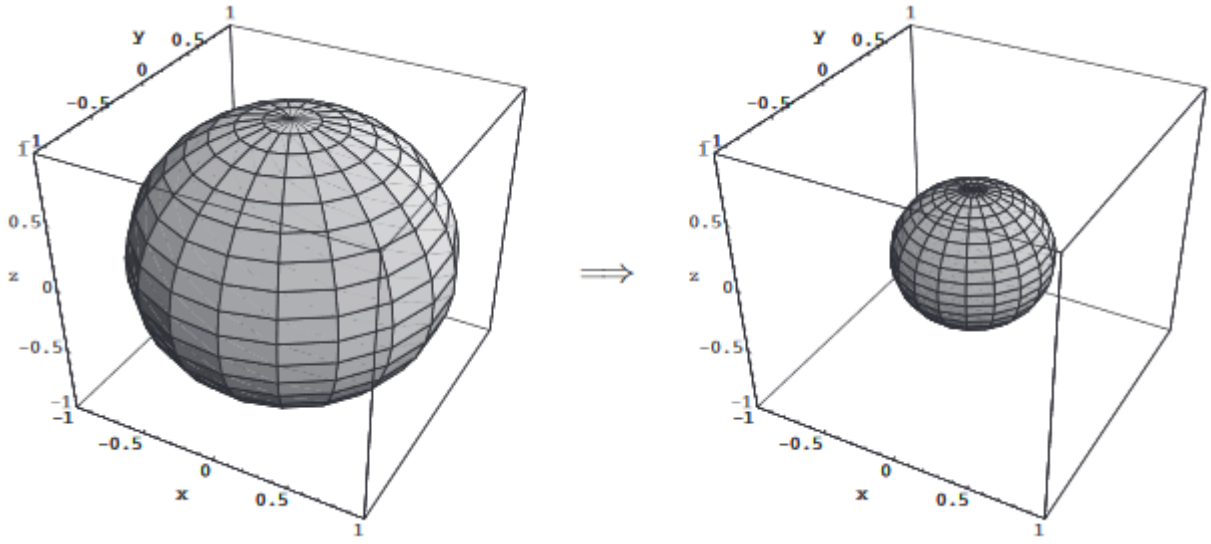

Figura 2.5: L'effetto del canale depolarizzante per p=0.5.

2.2.3 Amplitude damping

Con amplitude damping si intende la dissipazione di energia per un sistema quantistico. Gli elementi che caratterizzano questa operazione sono:

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}$$

$$E_1 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}$$

E_1 corrisponde al processo fisico di perdere un quanto di energia nell'ambiente dato che cambia lo stato $|1\rangle$ nello stato $|0\rangle$; E_0 invece lascia $|0\rangle$ nel suo stato ma riduce l'ampiezza dello stato $|1\rangle$. Fisicamente la seconda operazione corrisponde al caso in cui è più probabile che il sistema si trovi nello stato $|0\rangle$ dato che non c'è stata perdita di energia.

Possiamo visualizzare l'effetto di questa operazione sulla sfera di Bloch con la seguente trasformazione sulle componenti del vettore di Bloch:

$$(r_x.r_y, r_z) \longrightarrow (r_x\sqrt{1-\gamma}, r_y\sqrt{1-\gamma}, \gamma + r_z(1-\gamma))$$

se ora γ viene sostituito con la funzione $1 - e^{-\frac{t}{T_1}}$, dove t indica il tempo e T_1 il tempo caratteristico del processo, tutti i punti della sfera di Bloch vengono spostati verso lo stato $|0\rangle$ come esemplificato in figura 2.6.

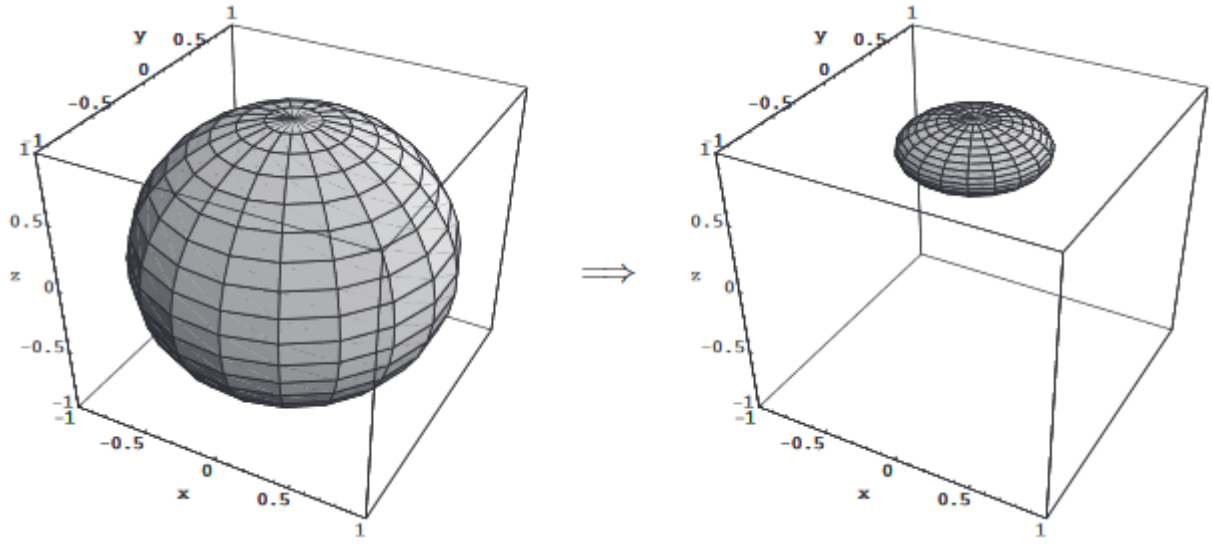

Figura 2.6: Amplitude damping sulla sfera di Bloch per p=0.8.

2.2.4 Phase damping

Con phase damping si intende un processo del tutto quantomeccanico poiché indica la perdita di informazione non accompagnata da perdita di energia. Fisicamente il processo risulta simile al modo in cui sono perturbati stati elettronici in un atomo quando interagiscono con cariche elettriche distanti.

Per modellare questo tipo di rumore possiamo immaginare di avere un qubit inizialmente nello stato $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ a cui applichiamo l'operatore di rotazione attorno all'asse $\hat{z}$, $R_z(\theta)$, con l'angolo θ di rotazione scelto in maniera casuale. Assumiamo allora che l'angolo θ sia rappresentato da una variabile che ha una distribuzione gaussiana di media 0 e deviazione standard 2λ.

Il risultato di questa operazione allora è l'operatore densità ottenuto mediando su θ

$$\rho = \frac{1}{\sqrt{4\pi\lambda}} \int_{-\infty}^{\infty} R_z(\theta)\,|\psi\rangle\,\langle\psi|\,R_z^{\dagger}(\theta) \exp\left(-\frac{\theta^2}{4\lambda}\right) d\theta \tag{2.10}$$

$$= \begin{bmatrix} |a|^2 & ab^* e^- \lambda \\ a^* b e^- \lambda & |b|^2 \end{bmatrix} \tag{2.11}$$

e vediamo come, dopo aver applicato la rotazione, i valori fuori dalla diagonale vanno a zero in maniera esponenziale. Se λ è un parametro che caratterizza la dispersione delle rotazioni random attorno a z e cresce con il tempo, si vede dalla (2.11) che a tempi molto lunghi gli elementi fuori dalla diagonale sono soppressi esponenzialmente e di fatto si riproducono i pesi sulla diagonale che corrisponde alla base preferenziale classica ma sono stati cancellati effetti quantistici di interferenza tra i due.

Il tempo caratteristico di questo processo si indica con T_2.

Non viene riportata un'immagine che mostra l'azione del *phase damping* sulla sfera di Bloch in quanto il suo effetto è simile a quello visto per il phase flip. Si proverà a dare una dimostrazione di questa ultima conclusione nell'appendice A.

Capitolo 3

Algoritmo di Grover

In questa sezione verrà discusso un algoritmo di ricerca quantistico: l'algoritmo di Grover. Per comprendere la sua importanza, in termini di efficienza computazionale, introdurremo prima i concetti legati alla complessità computazionale.

3.1 Complessità computazionale

Per complessità computazionale si intende lo studio dei requisiti, in termini di spazio e tempo, per risolvere alcuni problemi computazionali, inoltre fornisce una classificazione dei problemi tenendo conto della difficoltà necessaria per risolverli.

I problemi che prevedono una risposta nella forma 'si' oppure 'no' vengono chiamati *problemi decisionali*. Un esempio di problema decisionale può essere quello di decidere se un numero è primo oppure no.

Prendiamo in considerazione il seguente esempio. Se consideriamo $\Sigma = \{0, 1\}$ come l'insieme delle stringhe che possono essere realizzate combinando 0 ed 1, indicheremo con L il sottoinsieme delle stringhe pari $L = \{0, 10, 100, 110, ...\}$. In questo contesto chiamiamo L linguaggio e Σ alfabeto. Nel problema della primalità, prendendo come alfabeto $\Sigma = \{0, 1\}$, L sarebbe costituito dall'insieme delle stringhe che corrispondono ad un numero primo.

La domanda che dobbiamo porci a questo punto è la seguente: quanto velocemente sarebbe possibile capire se un numero è primo o meno? Si dice che un problema è di tempo $time(f(n))$ se esiste una macchina di Turing[1] che decide se il candidato x si trova nel linguaggio in tempo $O(f(n))$, con n lunghezza di x. Come si può intuire, il comportamento generale di un algoritmo viene descritto dal numero di step temporali che sono necessari per far funzionare l'algoritmo stesso. Abbiamo tre modi per classificare il comportamento di una funzione:

- Notazione O(O grande) per avere limiti superiori per la funzione. Per esempio, $f(n)$ è $O(g(n))$ se esistono due costanti c ed n_0 tali che per tutti i valori $n > n_0$ si ha $f(n) < cg(n)$. In questo modo per n molto grandi ci accorgiamo subito che $g(n)$ costituisce un limite superiore per la funzione $f(n)$;

- notazione Ω(Ω grande) per individuare limiti inferiori. Sulla falsa riga di quanto detto sopra diremo che una funzione $f(n)$ è $\Omega(g(n))$ se esistono due costanti c e n_0 tali per cui $f(n) \geq cg(n)$ per $n > n_0$ molto grandi;

[1] Per la trattazione esposta in questa tesi possiamo riferirci ad una macchina di Turing come un modello astratto per la computazione identificato come un computer avente un set base di istruzioni ed una memoria illimitata.

- l'ultima notazione Θ(Θ grande) si usa quando si vogliono descrivere funzioni che hanno lo stesso comportamento.

Dopo aver introdotto i precedenti concetti possiamo affermare che un problema può essere risolto in un tempo polinomiale se è in $time(n^k)$ per qualche k finito. La collezione di tutti i linguaggi L tali da essere in tempo $time(n^k)$, per qualche k, viene indicata con `P`. `P` rappresenta una classe di complessità.

Strettamente collegata alla classe `P` troviamo la classe `NP`(non deterministic polynomial time) che è anch'essa una classe di complessità per problemi decisionali. La differenza con `P` risiede nel fatto che una eventuale risposta 'si' può essere verificata in un tempo polinomiale. Anche in questo caso potremmo riferirci alla classe `NP` come la classe di quei problemi che possono essere risolti con l'ausilio di una macchina di Turing non deterministica.

Osserviamo che `P` è una sottoclasse di `NP`, ci basti pensare al fatto che se un problema può essere risolto in un tempo polinomiale allora una soluzione può essere verificata nello stesso tempo semplicemente risolvendo il problema. Tra i problemi aperti attinenti alla scienza dei computer c'è quello di determinare se esistono problemi che sono in `NP` ma non in `P`, brevemente indicato problema `P`$\neq$`NP`.

Un ruolo particolare all'interno della classe `NP` lo svolgono i problemi detti `NP`-completi. Quest'ultimo tipo di classe di complessità appartiene a quei problemi che se risolti in un tempo t garantiscono la possibilità a tutti gli altri problemi appartenenti alla classe `NP` di essere risolti in un tempo polinomiale in t. Tra i problemi `NP`-completi ricordiamo:

- Il problema del commesso viaggiatore(*travelling salesman problem*): dato un insieme di città, e note le distanze tra ciascuna coppia di esse, trovare il tragitto di minima percorrenza che un commesso viaggiatore deve seguire per visitare tutte le città una ed una sola volta;

- Il problema della soddisfacibilitá booleana(Satisfiability o SAT)): data una espressione booleana, ci si chiede se è possibile sostituire le variabili con i valori TRUE e FALSE in modo tale che l'espressione sia vera.

3.2 L'oracolo

Ci troviamo a cercare in uno spazio di ricerca composto da N elementi; piuttosto che concentrarci sugli elementi veri e propri, considereremo i loro indici, che per N elementi, sarà un numero compreso tra 0 ed $N-1$. A questo punto, per convenienza, supponiamo proprio che $N = 2^n$, in modo tale che il nostro indice può essere espresso utilizzando esattamente n bits. La nostra ricerca ammetterà delle soluzioni, M soluzioni, con $1 \leq \times M \leq N$. Supponiamo di disporre ora di una funzione $f(x)$ con $x \in [0, N-1]$, tale che $f(x) = 1$ se x è una soluzione al nostro problema di ricerca, e $f(x) = 0$ se invece non è soluzione. L'oracolo, per semplicità, possiamo immaginarlo come una scatola nera, di cui non discuteremo il funzionamento interno, in grado di riconoscere le soluzioni al problema di ricerca.

Come in altri casi, pensiamo all'oracolo come ad un operatore unitario O, che agisce nel seguente modo sulla base computazionale:

$$|x\rangle\,|q\rangle \longrightarrow |x\rangle\,|q \oplus f(x)\rangle$$

in cui chiamiamo $|q\rangle$ qubit oracolo e $|x\rangle$ sarà l'indice del registro. Ci accorgiamo che in questo modo il qubit oracolo sarà invertito se $f(x) = 1$ e lasciato così com'è se invece

$f(x) = 0$. In questo senso per sapere se uno $|x\rangle$ è soluzione del sistema, ci basterà preparare il qubit oracolo nello stato $|0\rangle$ e vedere se questo viene invertito in $|1\rangle$.

Nell'algoritmo di ricerca quantistico risulta utile avere il qubit oracolo inizialmente nello stato $\frac{(|0\rangle-|1\rangle)}{\sqrt{2}}$; allo stesso modo di prima, se x non è una soluzione, lo stato del qubit oracolo rimarrà invariato, se invece x è una soluzione allora $|0\rangle$ e $|1\rangle$ saranno invertiti dall'azione dell'oracolo, lasciando uno stato $-|x\rangle\,(\frac{|0\rangle-|1\rangle}{\sqrt{2}})$. Riassumiamo allora l'azione dell'oracolo:

$$|x\rangle\,(\frac{|0\rangle - |1\rangle}{\sqrt{2}}) \longrightarrow (-1)^{f(x)}\,|x\rangle\,(\frac{|0\rangle - |1\rangle}{\sqrt{2}})$$

da questo punto di vista allora lo stato del qubit oracolo rimane invariato, e possiamo ometterlo in modo da semplificare la nostra discussione. Riscriviamo di nuovo l'azione dell'oracolo:

$$|x\rangle \longrightarrow (-1)^{f(x)}\,|x\rangle$$

questa formulazione risulta molto più chiara e diretta. Infatti, in questo modo risulta evidente come l'azione dell'oracolo corrisponda a quella di evidenziare la soluzione al nostro problema, *shiftando* la fase della soluzione.

3.3 Procedimento dell'algoritmo

Prima di cercare nella lista di elementi non sappiamo dove si trovi l'elemento che cerchiamo; in questo senso ogni indice può essere un buon candidato per la nostra soluzione. Costruiamo allora il nostro stato iniziale in modo che presenti le stesse probabilità per ogni indice della lista. Il nostro sistema iniziale si troverà nello stato $|0\rangle^{\otimes n}$. Successivamente applichiamo la trasformazione di Hadamard H, in modo da ottenere una sovrapposizione uniforme. Lo stato dopo l'azione di H allora sarà:

$$|\psi\rangle = \frac{1}{N^{1/2}} \sum_{x=0}^{N-1} |x\rangle\,.$$

Se ora andassimo a misurare lo stato $|\psi\rangle$ nella base computazionale standard $\{|x\rangle\}$, in accordo con i postulati della meccanica quantistica collasserebbe in uno degli stati che costituiscono la base computazionale con una probabilità che vale $\frac{1}{N} = \frac{1}{2^n}$; in questo modo in media ci servirebbero $N = 2^n$ tentativi per indovinare la posizione dell'elemento che stiamo cercando. L'algoritmo di ricerca quantistico prevede l'applicazione ripetuta dell'*operatore di Grover* che chiamiamo G. L'iterazione di Grover può essere suddivisa in quattro step:

1. Applicazione dell'oracolo O;
2. applicazione della prima trasformata di Hadamard $H^{\otimes n}$;
3. applicazione di uno spostamento di fase tale per cui tutti gli stati ricevono uno shift di -1 eccetto $|0\rangle$
$$|x\rangle \longrightarrow -(-1)^{\delta_{x0}}\,|x\rangle\,;$$
4. seconda applicazione della trasformata di Hadamard $H^{\otimes n}$.

Risulta a questo punto utile osservare che gli effetti combinati degli step 2, 3 e 4 può essere scritta nel seguente modo

$$H^{\otimes n}(2\,|0\rangle\,\langle 0| - I)H^{\otimes n} = 2\,|\psi\rangle\,\langle\psi| - I$$

con $|\psi\rangle$ che rappresenta la sovrapposizione uniforme degli stati. In forma più compatta possiamo infine scrivere

$$G = (2\,|\psi\rangle\,\langle\psi| - I)O.$$

3.3.1 Una utile interpretazione geometrica

In questa sezione mostreremo come l'iterazione di Grover può essere in realtà interpretata come una doppia riflessione in uno spazio bidimensionale che è lo *span* del vettore costituito dalla sovrapposizione uniforme delle soluzioni al problema di ricerca, ed il vettore iniziale $|\psi\rangle$.

Indicheremo con $\sum_x'$ la somma sugli stati che sono soluzione al problema di ricerca e con $\sum_x''$ la somma sugli stati che invece non sono soluzione del nostro problema. Fatte queste premesse possiamo scrivere il vettore $|\psi\rangle$ come la somma di due contributi:

$$|\alpha\rangle \equiv \frac{1}{\sqrt{N-M}} \sum_x{}'' \,|x\rangle$$

$$|\beta\rangle \equiv \frac{1}{\sqrt{M}} \sum_x{}' \,|x\rangle$$

A questo punto risulta conveniente riscrivere il nostro stato $|\psi\rangle$ nel seguente modo:

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

così lo stato iniziale $|\psi\rangle$ appartiene allo spazio realizzato dallo span di $|\alpha\rangle$ e $|\beta\rangle$.

L'azione dell'oracolo O può essere visualizzata come una riflessione del vettore $|\alpha\rangle$ nello piano spazzato da $|\alpha\rangle$ e $|\beta\rangle$. Di fatti possiamo scrivere $O(a\,|\alpha\rangle + b\,|\beta\rangle) = a\,|\alpha\rangle - b\,|\beta\rangle$.

In maniera simile anche $2\,|\psi\rangle\,\langle\psi| - I$ opera una riflessione sempre nello stesso piano ma questa volta attorno al vettore $|\psi\rangle$. Due riflessioni corrispondono ad una rotazione, e questa riflessione ci assicura che lo stato $G^k\,|\psi\rangle$[2] appartiene sempre allo spazio realizzato dallo span di $|\alpha\rangle$ e $|\beta\rangle$. Proviamo ora a dare un valore agli angoli che caratterizzano queste rotazioni e poniamo $\cos\frac{\theta}{2} = \sqrt{\frac{N-M}{N}}$ in modo tale che $|\psi\rangle = \cos\frac{\theta}{2}\,|\alpha\rangle + \sin\frac{\theta}{2}\,|\beta\rangle$.

La figura 3.1 mostra in maniera più chiara il procedimento appena discusso. Una iterazione della *subroutine* di Grover trasforma allora lo stato $|\psi\rangle$ nel seguente modo

$$|\psi\rangle \longrightarrow G\,|\psi\rangle = \cos\frac{3\theta}{2}\,|\alpha\rangle + \sin\frac{3\theta}{2}\,|\beta\rangle\,.$$

Di conseguenza, ripetute applicazioni dell'operatore di Grover portano ad uno stato

$$G^k\,|\psi\rangle = \cos\frac{(2k+1)\theta}{2}\,|\alpha\rangle + \sin\frac{(2k+1)\theta}{2}\,|\beta\rangle$$

In conclusione, l'iterazione di Grover ruota il vettore di stato $|\psi\rangle$ in modo da farlo avvicinare all'asse $|\beta\rangle$; dopo che diverse iterazioni sono state applicate, una misura nella base computazionale produrrà con una probabilità più alta uno dei risultati utili sovrapposti nello stato $|\beta\rangle$, ovvero la soluzione al nostro problema.

[2]Con questa notazione vogliamo indicare l'applicazione dell'iterazione di Grover allo stato $|\psi\rangle$ k volte.

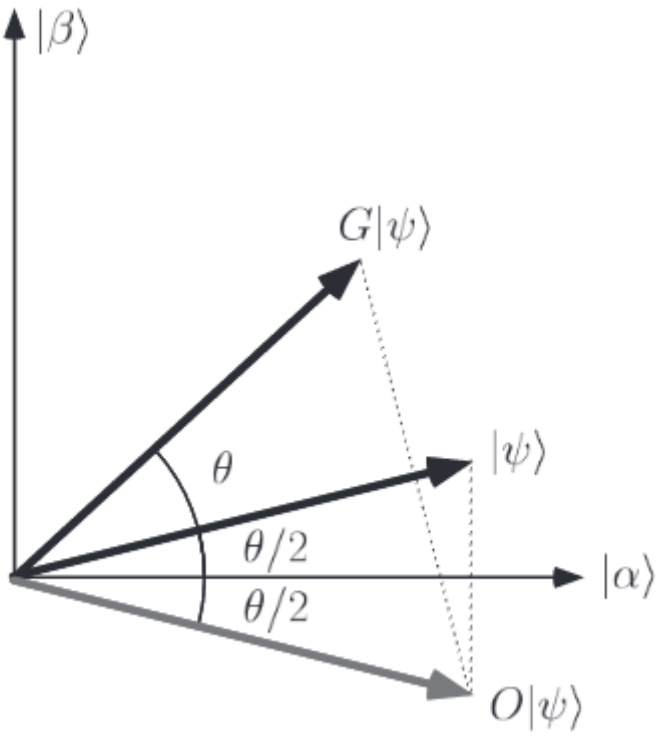

Figura 3.1: In figura troviamo l'azione di una singola iterazione di Grover: l'azione complessiva è quella di ruotare $|\psi\rangle$ verso $|\beta\rangle$, sovrapposizione delle soluzioni del problema di ricerca, di un angolo θ. Vediamo in dettaglio cosa accade. Inizialmente $|\psi\rangle$ si trova inclinato di un angolo $\frac{\theta}{2}$ rispetto allo stato $|\alpha\rangle$ perpendicolare a $|\beta\rangle$. L'azione dell'oracolo O invece riflette $|\psi\rangle$ attorno a $|\alpha\rangle$, successivamente l'operazione $2\,|\psi\rangle\,\langle\psi| - I$ opera una seconda riflessione attorno allo stato $|\psi\rangle$. Dopo ripetute iterazioni dell'algoritmo di Grover, il vettore di stato $|\psi\rangle$ si avvicina a $|\beta\rangle$ e a quel punto una misura nella base computazionale resituisce una soluzione al problema con molta probabilitá. L'algoritmo risulta particolarmente efficace perchè θ è di ordine $\Omega(\sqrt{\frac{M}{N}})$, in questo modo saranno necessarie solo $O(\sqrt{\frac{N}{M}})$ applicazioni dell'algoritmo per ruotare $|\psi\rangle$ vicino a $|\beta\rangle$.

3.3.2 Performance

Ci si domanda a questo punto quante volte è necessario applicare l'iterazione di Grover in modo da avvicinare $|\psi\rangle$ a $|\beta\rangle$. Lo stato iniziale del sistema, come abbiamo accennato nella sezione precedente, vale $|\psi\rangle = \sqrt{(N-M)/N}\,|\alpha\rangle + \sqrt{M/N}\,|\beta\rangle$, quindi ruotando il sistema di $arccos\sqrt{M/N}$ radianti porta il sistema in $|\beta\rangle$. Definiamo R come segue

$$R = CI(\frac{arccos\sqrt{M/N}}{\theta}) \tag{3.1}$$

dove $CI(x)$ sta per *closest integer* ed è una funzione che restituisce il numero intero più vicino ad x ed approssima in difetto. Possiamo dire che se ripetiamo l'iterazione R volte ruotiamo $|\psi\rangle$ verso $|\beta\rangle$ entro un angolo $\theta/2 \leq \pi/4$. Notiamo che una misurazione nella base computazionale ora restituisce la soluzione corretta con una probabilità di almeno un mezzo.

Prendiamo in considerazione ora il caso in cui $M \ll N$ per cui $\theta \approx \sin\theta \approx 2\sqrt{M/N}$, perciò l'errore angolare vale al massimo $\theta/2 \approx \sqrt{M/N}$ con una probabilità di errore che vale al più M/N.

Il risultato ottenuto in (3.1) per R risulta una espressione esatta per il numero di applicazioni dell'oracolo per far funzionare l'algoritmo di ricerca quantistico ma adesso proviamo a ricavare una soluzione più elegante e pratica che descriva il comportamento di R.

Dalla (3.1) si evince che $R \leq \lceil \pi/2\theta \rceil$[3], di conseguenza un limite inferiore per θ corrisponderà ad un limite superiore per R. Se assumiamo che $M \leq N/2$ consegue che

$$\frac{\theta}{2} \geq \sin\frac{\theta}{2} = \sqrt{\frac{M}{N}}$$

e sostituendo abbiamo

$$R \leq \left\lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rceil \tag{3.2}$$

Concludiamo allora che sono necessarie $R = O(\sqrt{N/M})$ iterazioni di Grover(e quindi applicazioni dell'oracolo) per ottenere una soluzione al problema di ricerca con alta probabilità. Classicamente erano necessarie $O(N/M)$ applicazioni dell'oracolo di conseguenza c'è stato un miglioramento di tipo quadratico.

3.4 Ricerca in un database non strutturato

Supponiamo di avere una lista contenente 1000 nomi di fiori, e volessimo cercare il fiore 'tulipano' all'interno della lista. Se il fiore appare una sola volta e la lista non è ordinata in ordine alfabetico, in media dovremo ispezionare 500 nomi prima di trovare il tulipano. In questa sezione ci occuperemo di mostrare come e se è possibile velocizzare questo processo utilizzando un algoritmo di ricerca quantistico.

Immaginiamo di avere un database contenente $N \equiv 2^n$ oggetti, ognuno lungo l bits e indicheremo questi oggetti con $d_1, d_2, ..., d_N$. Vogliamo allora determinare la posizione di una stringa di l bit, che chiameremo s. Mostreremo prima l'approccio per un computer classico e successivamente quello per un computer quantistico.

In un computer classico, per risolvere questo problema, sfruttiamo due parti: la CPU(central processing unit) che è il posto dove vengono manipolati i dati sfruttando una piccola quantità di memoria temporanea; la seconda parte invece è una memoria che memorizza il database in una stringa di 2^n blocchi da l bit. La memoria non è in grado di manipolare dati ma può inserire dati nella CPU e memorizzare i dati manipolati dalla CPU.

Assumiamo inoltre che la CPU sia abbastanza spaziosa da poter contenere $n \equiv [logN]$ indici di bit dal momento che sarà necessario inizializzare un indice a n bit per gli elementi del database all'interno della CPU. L'algoritmo funziona nel seguente modo: l'indice parte da zero e ad ogni iterazione viene aumentato di uno; ad ogni iterazione vengono confrontati l'ingresso corrispondente all'indice attuale e la stringa che stiamo cercando. Se c'è corrispondenza, l'algoritmo si ferma e fornisce il valore dell'indice, altrimenti prosegue. Notiamo subito che nel peggiore dei casi sarà necessario caricare oggetti in ingresso dalla memoria alla CPU 2^n volte.

Supponiamo ora che anche il nostro computer quantistico sia costituito da due componenti sulla falsa riga di un computer classico, una memoria ed una CPU. In questo caso la nostra CPU sarà costituita da 4 registri: (1) un indice per n qubit inizializzato a $|0\rangle$; (2) un registro per l qubit inizializzato come $|s\rangle$ che rimarrà immutato durante tutta

[3]Con la notazione $\lceil x \rceil$ si intende il primo intero più grande di x.

la computazione; (3) un registro per i dati di l qubit inizializzato a $|0\rangle$; (4) un registro a 1 qubit inizializzato come $\frac{(|0\rangle-|1\rangle)}{\sqrt{2}}$.

L'unità di memoria può essere implementata in due modi, la più semplice delle quali è una memoria contenente $N = 2^n$ celle di lunghezza l qubit ognuna, contenenti gli elementi del database $|d_x\rangle$. La seconda implementazione invece realizza la memoria come se fosse una memoria *classica* con $N = 2^n$ celle di l bit ognuna contenenti gli ingressi del database d_x. Tuttavia, a differenza di una memoria classica può essere indirizzata da un indice x che può essere in una sovrapposizione di diversi valori. Abbiamo quindi un indice quantistico che consente di caricare dalla memoria alla CPU sovrapposizioni dei valori delle celle.

Vediamo ora il funzionamento della memoria: se il registro d'indice della CPU si trova nello stato $|x\rangle$ ed il registro dei dati si trova nello stato $|d\rangle$, allora il contenuto d_x della *x-esima* cella di memoria viene aggiunto al registro dei dati $|d\rangle \longrightarrow |d \oplus d_x\rangle$. É possibile dimostrare che un indirizzamento di questo tipo è fisicamente realizzabile[1].

Per realizzare un algoritmo di ricerca quantistico bisognerà realizzare prima un oracolo che dovrà applicare un phase-flip all'indice che localizza s all'interno della memoria. Prendiamo la CPU nel seguente stato

$$|x\rangle\,|s\rangle\,|0\rangle\,\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

allora caricare i dati dalla memoria alla CPU ci porta nel seguente stato

$$|x\rangle\,|s\rangle\,|d_x\rangle\,\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

a questo punto il secondo ed il terzo registro vengono messi a confronto e se sono uguali allora un bit-flip viene applicato al registro 4; se invece non sono uguali rimane tutto invariato.

$$|x\rangle\,|s\rangle\,|d_x\rangle\,\frac{|0\rangle-|1\rangle}{\sqrt{2}} \longrightarrow \begin{cases} -\,|x\rangle\,|s\rangle\,|d_x\rangle\,\frac{|0\rangle-|1\rangle}{\sqrt{2}} & d_x = s \\ \;\;|x\rangle\,|s\rangle\,|d_x\rangle\,\frac{|0\rangle-|1\rangle}{\sqrt{2}} & d_x \neq s \end{cases}$$

Il registro dei dati successivamente viene ristabilito a $|0\rangle$ caricando di nuovo dalla memoria. Vediamo che l'azione dell'oracolo lascia invariati i registri 2, 3 e 4 non entangled con il registro 1. L'effetto complessivo allora è quello di portare lo stato $|x\rangle$ nello stato $-\,|x\rangle$ solo se $d_x = s$.

Sfruttando un oracolo implementato in questo modo possiamo usare l'algoritmo di ricerca quantistico per determinare la posizione di s all'interno del database utilizzando solo $O(\sqrt{N})$ caricamenti, comparato al caso classico in cui erano necessari N caricamenti.

Concludiamo questo capitolo con la consapevolezza che non sembra produttivo utilizzare un algoritmo di ricerca quantistico per cercare un fiore all'interno di una lista di fiori. I database di questo tipo di solito vengono strutturati in ordine alfabetico e per cercare un nome al suo interno sarebbero richieste solamente $O(logN)$ operazioni. Dall'altro punto di vista per realizzare un algoritmo di ricerca quantistico sarebbe necessario implementare una memoria con un indirizzamento di tipo quantistico con conseguente utilizzo di hardware non economicamente vantaggioso rispetto a quello classico già esistente.

Ci si aspetta che algoritmi di tipo quantistico , quindi, non vengano utilizzati per cercare elementi all'interno di database classici, bensì potrebbero essere utilizzati per cercare soluzioni a problemi più complicati come potrebbero essere quelli esposti nella sezione 3.1 ovvero il problema del *commesso viaggiatore* oppure i problemi di *soddisfacibilitá*.

Capitolo 4

Qiskit e la IBM Q Experience

QISKit[2] sta per Quantum Information Science Kit ed è un'interfaccia di programmazione, basata sul linguaggio Python, per programmare computer quantistici. L'interfaccia garantisce accesso diretto ai computer quantistici della IBM dal momento che l'azienda mette a disposizione i propri device per entrare in contatto con il mondo della computazione quantistica; è inoltre possibile utilizzare anche alcuni tipi di simulatori i quali spesso risultano utili per preparare il codice e quindi confrontare i risultati ottenuti con i dispositivi reali ed osservare, per esempio, il modo in cui il rumore quantistico vada ad influenzare i dispositivi reali.

I qubit realizzati dalla IBM[3] sono dei qubit transmon superconduttivi a frequenza fissata, ovvero dei qubit basati sulla giunzione di Josephson non sensibili al rumore di carica; sono stati utilizzati dei qubit a frequenza fissa poiché sono meno sensibili alle fluttuazioni del campo magnetico esterno che potrebbero danneggiare l'informazione quantistica. Questi sono realizzati su wafers di silicio con metalli superconduttivi come l'alluminio e il niobio.

Tra le caratteristiche di questi qubit rivestono particolare importanza due tempi caratteristici T_1 e T_2 che sono i tempi caratteristici dei processi descritti nelle sezioni 2.2.3 e 2.2.4.

Sono a disposizione le porte logiche più comuni tra cui ricordiamo la porta di Hadamard H, il *controlled-not* CNOT e le porte di Pauli. Ci sono inoltre anche altre porte logiche che non erano state introdotte precedentemente in questa relazione tra cui troviamo la porta T che applica una fase $\pi/4$ e ha la seguente rappresentazione:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

Nell'interfaccia QISKit esiste una funzione che restituisce tutte le informazioni tecniche riguardanti il dispositivo che si utilizza in quel momento. Tra le informazioni presentate si notano delle porte logiche di base per quel dispositivo e tra queste ci sono tre porte logiche $\{u_1, u_2, u_3\}$ che sono quelle fisicamente realizzabili sui qubit all'interno del dispositivo della IBM. Riportiamo cosa si intende allora per u_1, u_2, u_3. Esprimiamo la trasformazione unitaria U che porta un qubit nello stato $|0\rangle$ nello stato definito dall'equazione (1.4) come segue, tenendo a mente che deve valere anche $U^\dagger U = I$:

$$U = \begin{bmatrix} \cos\frac{\theta}{2} & -e^{i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i\lambda+i\phi}\cos\frac{\theta}{2} \end{bmatrix}$$

e fatta questa precisazione notiamo che $u_1 = U(0, 0, \lambda)$, $u_2 = U(\pi/2, \phi, \lambda)$ ed infine $u_3 = U(\theta, \phi, \lambda)$.

In questo lavoro di tesi si è prevalentemente voluto prendere confidenza con i temi della computazione quantistica e di conseguenza la piattaforma è stata utilizzata per la maggior parte del tempo per visualizzare i risultati di eventuali esperimenti realizzati con qualche qubit e la loro realizzazione circuitale.

Sul sito della IBM Q Experience si trovano diverse guide per avvicinarsi a questi temi con la possibilità di avere accesso ad esempi che vanno dalla visualizzazione degli strani comportanti legati all'entanglement quantistico fino ad arrivare anche alla possibilità di vedere come potrebbe essere scritto un algoritmo quantistico tipo.

É inoltre disponibile una piattaforma grafica chiamata *quantum composer*[4] in cui è possibile realizzare circuiti che sono risultati utili per un apprendimento più immediato dei concetti legati alla misura in una determinata base computazionale oppure un'altra.

In sostanza, creare un account sul loro sito permette di utilizzare il loro dispositivo a Melbourne dotato di 14 qubit, il dispositivo a Tenerife da 5 qubit, il dispositivo a Yorktown da 5 qubit nonché il simulatore *qasm* da 32 bit. Una volta creati e lanciati gli esperimenti sui loro dispositivi reali, i dati vengono processati e successivamente restituiti all'utente in un lasso di tempo che può essere influenzato dalla coda che c'è per un determinato dispositivo e dalla complessità dell'esperimento stesso. Nella mia esperienza per test di algoritmi semplici, come quello che verrà esposto tra poco nella sezione 4.1, ho aspettato circa $15 minuti$, mentre per algoritmi più articolati come quello che verrà discusso nella sezione 4.2 sono state necessarie anche delle ore per ottenere i risultati.

4.1 Stato di Bell per 3 qubit

In una delle esperienze con il QISKit ho provato a realizzare stati di Bell per 3 qubit con conseguenti misure nelle basi computazionali passando per la visualizzazione del loro operatore densità ρ fino ad arrivare al confronto dei risultati ottenuti sfruttando il simulatore ed i risultati ottenuti sfruttando *ibmqx4*, uno dei dispositivi a 5 qubit.

Elenchiamo di seguito alcuni istogrammi ottenuti con QISKit ed infine anche il circuito che implementa la realizzazione di uno stato di Bell per 3 qubit.

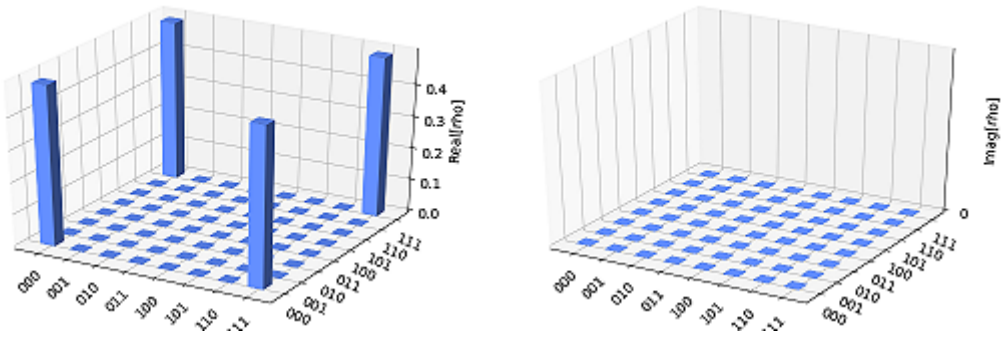

Figura 4.1: Nell'istogramma sono presenti parte immaginaria e reale perché, come abbiamo accennato più volte, stiamo lavorando con vettori appartenenti allo spazio dei numeri complessi $\mathbb{C}$. La parte immaginaria tuttavia è nulla.

In figura 4.1 c'è la rappresentazione, tramite operatore densità, dello stato $|\psi\rangle = \frac{|000\rangle+|111\rangle}{\sqrt{2}}$. Di fatto per questo stato la ρ corrispondente vale

$$\rho = |\psi\rangle\langle\psi| = \frac{(|000\rangle + |111\rangle)(\langle 000| + \langle 111|)}{2}$$

$$= \frac{1}{2}(|000\rangle\langle 000| + |111\rangle\langle 111| + |000\rangle\langle 111| + |111\rangle\langle 000|)$$

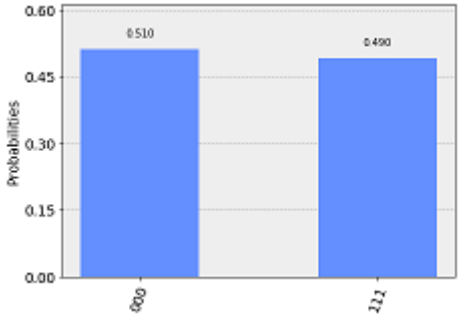

Figura 4.2: Nell'istogramma troviamo le probabilità che lo stato si trovi nello stato $|000\rangle$ oppure nello stato $|111\rangle$ ottenute con il simulatore, ci accorgiamo che sono approssimativamente identiche e, cosa più importante, esistono solo due colonne all'interno dell'istogramma di conseguenza non ha agito alcun tipo di rumore durante la misura ed è il risultato che ci aspettiamo dato che stiamo utilizzando un simulatore.

In particolare nella figura 4.2 è presentato l'istogramma per la misura dello stato di Bell per 3 qubit dopo aver ripetuto la misura 8192 volte; lo stato '111' è stato contato 4014 volte mentre lo stato '000' 4178 volte.

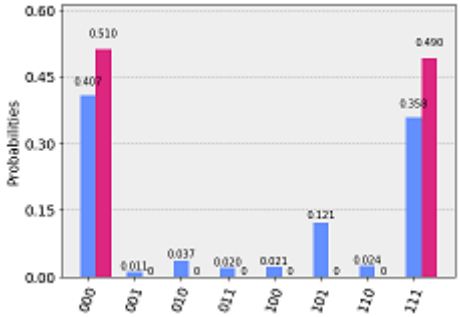

Figura 4.3: Istogrammi delle misurazioni sul dispositivo reale e sul simulator sovrapposti; con il color prugna ritroviamo i risultati precedentemente visti per il simulatore e con il celeste invece i risultati del dispositivo reale.

Nella figura 4.3 è evidente l'azione del rumore nei dispositivi reali. Si nota subito come non vi siano più solamente due stati pienamente definiti ed unici ma bensì compaiono anche alcuni degli altri stati che si potrebbero ottenere per combinazioni di 3 qubit che non ci aspetteremmo considerando il circuito che abbiamo tentato di realizzare.

Concludiamo con la figura 4.4 del circuito che realizza uno stato di Bell per 3 qubit, ottenuta sempre mediante QISKit. Se si fa attenzione al circuito, i qubit sono preparati prima della computazione sempre nello stato $|0\rangle$, di conseguenza prima di iniziare, il sistema si trova nello stato $|\psi\rangle = |00000\rangle$.

4.2 Una implementazione dell'algoritmo di Grover per 4 qubit

Tenendo a mente le considerazioni fatte sul funzionamento dell'algoritmo di Grover e di un oracolo in grado di riconoscere la soluzione al nostro problema di ricerca si è tentato

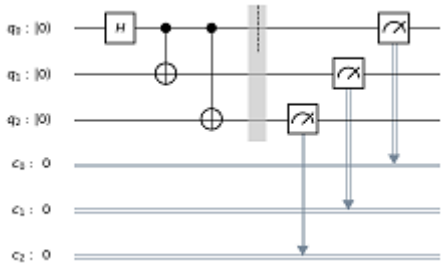

Figura 4.4: Notiamo che lo stato di Bell per 3 qubit viene realizzato in tre passi: prima si crea lo stato sovrapposizione per un qubit mediante la porta di Hadamard, successivamente si applicano due CNOT avendo come control qubit sempre il primo qubit, nello stato di Hadamard, e come target qubit i due qubit successivi. Per realizzare delle misure sui qubit sarà necessario proiettare i risultati delle loro misure rispettivamente su 3 bit classici che vengono istanziati solamente a questo scopo.

di simulare il funzionamento di un algoritmo di ricerca tra N=16 elementi, sfruttando di conseguenza 4 qubit[4]. Anche in questo caso c'è stata l'implementazione circuitale[5] che è molto più articolata di quella mostrata in precedenza in quanto in qualche modo bisogna preoccuparsi di realizzare l'oracolo ed anche la componente relativa all'amplificazione del modulo della soluzione. All'interno della parte di codice relativa all'amplificazione è stata implementata anche la porta `cccZ` che agisce come una porta logica di tipo *controlled* ma a differenza di quelle viste in precedenza questa ha come control qubit i primi 3 qubit e come target qubit solo l'ultimo, a cui viene applicata la trasformazione Z.

In questo problema di ricerca si è supposto la soluzione fosse unica e si è implementato l'oracolo in modo da applicare lo shift di phase a solo una delle soluzioni possibili, quindi solo uno dei sedici stati che si possono ottenere da 4 qubit. Supponiamo che l'oracolo riconosca la soluzione 0010 e riportiamo di seguito due istogrammi ottenuti per il caso del simulatore ed il caso del dispositivo reale rispettivamente.

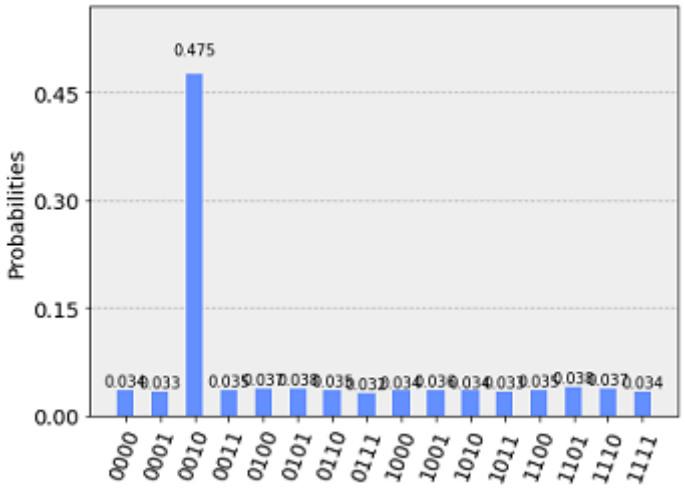

Figura 4.5: In questo istogramma sono riportate le probabilità relative agli stati per 4 qubit, notiamo come quella relativa allo stato 0010 sia molto più elevata delle altre, indicazione di una buona riuscita del test dell'algoritmo sul simulatore.

Dall'istogramma presente in figura 4.5 ci possiamo accorgere che quando andremo a misurare lo stato dei 4 qubit nella base computazionale otterremo con una probabilità

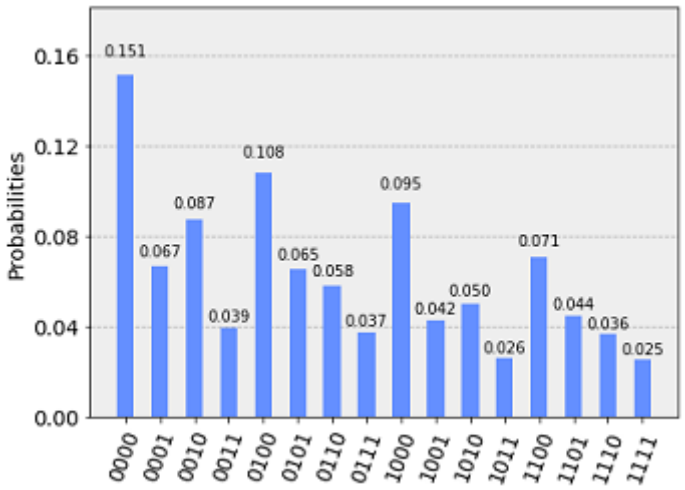

Figura 4.6: Istogramma per algoritmo di Grover con 4 qubit sul dispositivo reale *ibmq_16_melbourne*.

molto più alta lo stato 0010 che è lo stato corrispondente all'implementazione dell'oracolo e quindi la soluzione al nostro problema di ricerca.

Presentiamo ora l'istogramma relativo al test dell'algoritmo sul dispositivo *ibmq_16_melbourne* il quale è un dispositivo fisico reale a 14 qubit.

Risulta evidente in figura 4.6 come in questo caso la soluzione al nostro problema di ricerca non solo non abbia la probabilità più alta di essere individuata ma addirittura esistono anche stati che hanno probabilità significativamente maggiori.

Quando si va a testare uno dei computer quantistici collegandosi al sito della IBMQ Experience bisogna anche tenere conto del tempo di attesa che c'è da aspettare prima di poter ottenere i propri risultati. Sulla piattaforma QISKit esistono delle funzioni che restituiscono i nomi dei terminali meno occupati in termini di esperimenti in attesa di essere eseguiti. In questo senso i primi tentativi sono stati fatti sulla piattaforma *ibmq_16_melbourne* dal momento che in quel giorno era quella meno affollata. I secondi tentativi sono stati svolti successivamente e questa volta la piattaforma più disponibile era la *ibmqx4*.

Come si evince dalla figura 4.7 anche in questo caso i risultati non sembrano paragonabili a quelli ottenuti mediante il simulatore.

Sia nel caso del simulatore che nel caso del dispositivo reale sono state ripetute le misure 8192 volte in quanto per il dispositivo reale è il numero massimo di *shots* che possono essere tentati alla volta. Per il simulatore la risposta corretta è stata trovata nel $\frac{3888}{8192} \approx 47,46\%$ dei casi, mentre per il dispositivo reale si ottiene solo il $\frac{711}{8192} \approx 8,68\%$ dei casi.

Sarebbe interessante poter incrementare la statistica considerevolmente per capire come si abbatte l'errore statistico dovuto alla decoerenza e alle eventuali imperfezioni dei gates nel caso dei dispositivi reali.

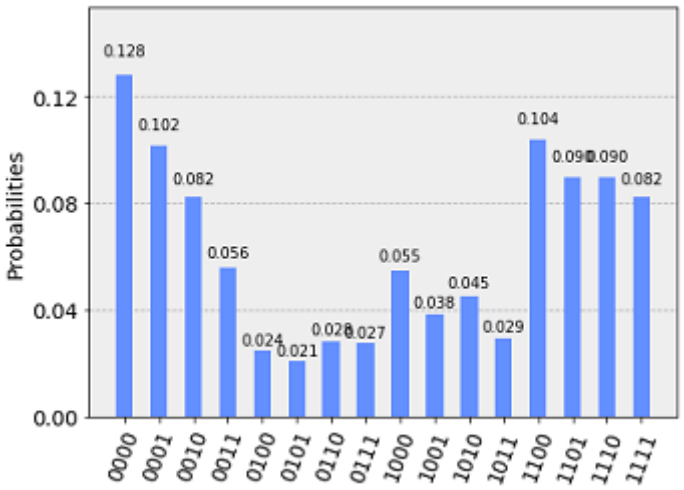

Figura 4.7: Istogramma per algoritmo di Grover con 4 qubit sul dispositivo reale *ibmqx4*.

Capitolo 5

Conclusioni

Come si può dedurre dagli istogrammi del capitolo precedente, in tutti i casi in cui è stato utilizzato il dispositivo reale della IBM c'è stato un netto peggioramento dei risultati ottenuti rispetto a quelli attesi teoricamente e riscontrati nella fase di simulazione preliminare usando il codice per l'algoritmo di Grover. Questo deriva dal fatto che in questo momento il rumore quantistico gioca ancora un ruolo davvero importante e per nulla trascurabile.

Dobbiamo considerare che, sempre allo stato attuale delle cose, i dispositivi disponibili per provare degli esperimenti quantistici sono anch'essi limitati in quanto non è possibile eseguire un esperimento più di 8192 tentativi per volta e anche per quanto riguarda la realizzazione circuitale c'è un numero massimo di 75 componenti utilizzabili insieme.

In questo senso forse si può anche dire che al momento della scrittura di questa tesi l'hardware a disposizione non è ancora adatto per provare algoritmi come quello di ricerca quantistico, o almeno per aspettarsi dei risultati che siano coerenti con le previsioni teoriche entro intervalli di confidenza ragionevoli.

Una volta migliorato il livello di decoerenza e/o aumentata la statistica potrebbe diventare decisivo sfruttare i dispositivi con un numero maggiore di qubits, già esistenti o in fase di sviluppo, per vedere l'effettiva potenza dell'approccio quantistico.

Ringraziamenti

Per primo vorrei ringraziare il Prof. Cristian Degli Esposti Boschi per avermi aiutato e seguito negli ultimi mesi, nonché per avermi insegnato diverse cose.

Successivamente ringrazio i miei genitori perché mi hanno dato la possibilità di portare avanti i miei studi,i miei fratelli ed in particolare mio fratello Michele che mi ha accompagnato a distanza in questo percorso di studi.

Ringrazio anche Serena perché mi è rimasta vicino anche nei momenti che sembravano più complicati, ed infine ringrazio i miei colleghi Inti, Paola ed Enrico con i quali ho condiviso diverse esperienze universitarie.

Appendice A

Nel libro non era inserita una dimostrazione diretta del perché, partendo dalla ρ vista nell'equazione (2.11), si potesse arrivare a dire che la deformazione di phase damping subita da uno stato puro che si trova sulla sfera di Bloch fosse simile a quella ottenuta nel caso del phase-flip, per questo motivo in questa appendice riportiamo i passaggi necessari per giustificare questa conclusione

Ripartiamo quindi dalla ρ vista nell'equazione (2.11) e ricordiamoci che in generale un vettore sulla sfera di Bloch può essere espresso anche mediante la forma (2.9). Inoltre bisogna tenere a mente l'equazione (1.4). Fatte queste considerazioni possiamo asserire che

$$(\hat{r} \cdot \vec{\sigma}) \left|\psi_r\right\rangle = \left|\psi_r\right\rangle$$

per uno stato puro sulla sfera di Bloch. Questo ci porta a poter scrivere

$$\begin{bmatrix} r_z & r_x - ir_y \\ r_x + ir_y & r_z \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

da cui tenuto conto delle equazioni riprese all'inizio dell'appendice possiamo, senza ulteriori conti, concludere che

$$a = \cos\frac{\theta}{2}$$

$$b = e^{i\beta}\sin\frac{\theta}{2}$$

Ora riscriviamo la ρ che abbiamo trovato nell'equazione (2.11) con questi valori.

$$\xi(\rho) = \frac{1 + \vec{\xi} \cdot \vec{\sigma}}{2} = \begin{bmatrix} \cos^2\frac{\theta}{2} & \cos\frac{\theta}{2}\sin\frac{\theta}{2}e^{-i\beta}e^{\lambda} \\ \cos\frac{\theta}{2}\sin\frac{\theta}{2}e^{i\beta}e^{\lambda} & \sin^2\frac{\theta}{2} \end{bmatrix}$$

ed allo stesso modo di quanto visto sopra possiamo scrivere

$$\xi(\rho) = \begin{bmatrix} \frac{1+\xi_z}{2} & \frac{\xi_x - i\xi_y}{2} \\ \frac{\xi_x + i\xi_y}{2} & \frac{1-\xi_z}{2} \end{bmatrix}$$

adesso siamo in grado di ricavare le componenti del vettore sulla sfera di Bloch dopo che ha agito il rumore. Avremo

$$\xi_z = 2\cos^{\frac{\theta}{2}} - 1 = \frac{1}{2}(e^{i\theta} + e^{-i\theta} + 2 - 1) = \cos\theta$$

$$\xi_x - i\xi_y = 2\cos\frac{\theta}{2}\sin\frac{\theta}{2}e^{-i\beta-\lambda} = e^{-i\beta-\lambda}(e^{i\frac{\theta}{2}} + e^{-i\frac{\theta}{2}})(\frac{e^{\frac{i\theta}{2}} - e^{-i\theta}}{2i}) = \frac{e^{\frac{i\theta}{2}} - e^{\frac{-i\theta}{2}}}{2i}e^{-i\beta-\lambda}$$

$$\xi_x - i\xi_y = \sin\frac{\theta}{2}e^{-i\beta}e^{-\lambda}$$

A questo punto ci accorgiamo che abbiamo trovato un ellissoide dato che rimangono

$$\xi_z = \cos\theta = r_z$$

$$\xi_x = e^{-\lambda} r_x$$
$$\xi_y = e^{-\lambda} r_y$$

ed inoltre la componente z del vettore sulla sfera di Bloch rimane invariata mentre le altre due decrescono con λ che cresce; in questo modo se λ cresce con il tempo durante il quale agisce il damping la sfera di Bloch si contrae nello stesso modo di quanto visto per il caso del phase damping.

Gli elementi per questa operazione quantistica valgono

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & u \end{bmatrix}$$

$$E_1 = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{1-u^2} \end{bmatrix}$$

e per $u = e^{-\lambda}$

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\lambda} \end{bmatrix}$$

$$E_1 = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{1-e^{-2\lambda}} \end{bmatrix}$$

Per parametrizzare la sfera in tre dimensioni sono state usate le coordinate sferiche usuali

$$\begin{cases} r_x = \sin\theta\cos\phi \\ r_y = \sin\theta\sin\phi \\ r_z = \cos\theta \end{cases}$$

Appendice B

Nel capitolo 3 si è mostrato come è possibile cercare soluzioni al problema di ricerca consultando l'oracolo $O(\sqrt{N})$ volte; in questa appendice ci preoccupiamo di mostrare come non sia possibile farlo richiamando l'oracolo meno di $\Omega(\sqrt{N})$ volte. In questo modo si dimostra che l'algoritmo è ottimale.

Per semplicità prenderemo in considerazione il caso in cui abbiamo un'unica soluzione x e supponiamo che l'algoritmo inizi nello stato $|\psi\rangle$.

Per determinare la soluzione x, possiamo applicare l'oracolo O_x che fornisce un phase-shift di -1 allo stato soluzione $|x\rangle$ e lascia invariati tutti gli altri stati ortogonali a $|\psi\rangle$; allora scriviamo O_x come $I - 2\,|x\rangle\,\langle x|$.

Applichiamo l'oracolo O_x esattamente k volte con rispettivi operatori unitari tra ogni applicazione dell'oracolo. Definiamo:

$$|\psi_k^x\rangle \equiv U_k O_x U_{k-1} O_x ... U_1 O_x\,|\psi\rangle$$

$$|\psi_k\rangle \equiv U_k U_{k-1} ... U_1\,|\psi\rangle$$

con la notazione $|\psi^k\rangle$ intendiamo lo stato che risulta dopo l'applicazione di U_k trasformazioni unitarie senza l'applicazione dell'oracolo O_x.

Il nostro obiettivo sarà quello di limitare la quantità

$$D_k \equiv \sum_x ||\psi_k^x - \psi_k||^2$$

dove per praticità abbiamo utilizzato la notazione ψ per indicare $|\psi\rangle$. Intuitivamente D_k è una misura della deviazione dopo k applicazioni dell'oracolo; se questa quantità è piccola allora tutti gli stati $|\psi_k^x\rangle$, indipendentemente da $|x\rangle$, sono praticamente uguali e non è possibile distinguere la soluzione x con alta probabilità.

La strategia della seguente dimostrazione consiste nel provare due cose: (1) un limite su D_k che mostra che esso non può crescere più velocemente di $O(k^2)$; (2) una dimostrazione che D_k deve essere $\Omega(N)$ se è possibile distinguere tra N alternative.

Prima di tutto diamo una dimostrazione per induzione che $D_k \leq 4k^2$ e questa risulta essere vera per $k = 0$ con $D_k = 0$.

Notiamo che

$$D_{k+1} = \sum_x ||O_x \psi_k^x - \psi_k||^2$$

$$= \sum_x ||O_x(\psi_k^x - \psi_k) + (O_x - I)\psi_k||^2$$

e se applichiamo $||b+c||^2 = ||b||^2 + 2||b||\;||c|| + ||c||^2$ con $b \equiv O_x(\psi_k^x - \psi_k)$ e $c \equiv (O_x - I)\psi_k = -2\,\langle x|\psi_k\rangle\,|x\rangle$, poiché $O_x^\dagger O = I$, otteniamo

$$D_{k+1} \leq \sum_x (||\psi_k^x - \psi_k||^2 + 4||\psi_k^x - \psi_k||\;|\,\langle x|\psi_k\rangle\,| + 4|\,\langle \psi_k|x\rangle\,|^2)$$

successivamente applichiamo la disuguaglianza di Cauchy-Schwarz al secondo termine della disuguaglianza scritta sopra tenendo conto anche del fatto che $\sum_x |\langle x|\psi_k\rangle|^2 = 1$ e otteniamo

$$D_{k+1} \leq D_k + 4\left(\sum_x ||\psi_k^x - \psi_k||^2\right)^{1/2}\left(\sum_{x'} |\langle \psi_k|x'\rangle|^2\right)^{1/2} + 4$$

$$\leq D_k + 4\sqrt{D_k} + 4$$

Dalla nostra ipotesi $D_k \leq 4k^2$ concludiamo che

$$D_{k+1} \leq 4k^2 + 8k + 4 = 4(k+1)^2$$

e terminiamo la nostra dimostrazione per induzione.

Per quanto riguarda la seconda parte dovremo dimostrare che la probablità di successo sarà alta solo se D_k è $\Omega(N)$.

Supponiamo che

$$|\langle x|\psi_k^x\rangle|^2 \geq 1/2$$

per ogni x in modo tale che una misura ci restituirà x con una probabilità di almeno un mezzo. Possiamo sostituire $e^{i\theta}|x\rangle$ a $|x\rangle$ in quanto non abbiamo una variazione di proabiità di successo, e senza perdita di generalità possiamo assumere che $\langle x|\psi_k^x\rangle = |\langle x|\psi_k^x\rangle|$, di conseguenza possiamo scrivere

$$||\psi_k^x - x||^2 = 2 - 2|\langle x|\psi_k^x\rangle| \leq 2 - \sqrt{2}$$

Definiamo adesso $E_k \equiv \sum_x ||\psi_k^x - x||^2$ ci accorgiamo che $E_k \leq (2-\sqrt{2})N$; inoltre definiamo $F_k \equiv \sum_x ||x - \psi_k||^2$. Riscriviamo D_k:

$$D_k = \sum_x ||(\psi_k^x - x) + (x - \psi_k)||^2$$

$$\geq \sum_x ||\psi_k^x - x||^2 - 2\sum_x ||\psi_k^x - x||\,||x - \psi_k|| + \sum_x ||x - \psi_k||^2$$

$$E_k + F_k - 2\sum_x ||\psi_k^x - x||\,||x - \psi_k||$$

Se a questo punto applichiamo la disuguaglianza di Cauchy-Schwarz otteniamo

$$\sum_x ||\psi_k^x - x||\,||x - \psi_k|| \leq \sqrt{E_k F_k}$$

e in questo modo abbiamo

$$D_k \geq E_k + F_k - 2\sqrt{E_k F_k} = (\sqrt{F_k} - \sqrt{E_k})^2.$$

Per concludere la dimostrazione dobbiamo notare che utilizzando di nuovo la disuguaglianza di Cauchy-Schwarz si può far vedere che $F_k \geq 2N - 2\sqrt{N}$ con $|\psi\rangle$ vettore di stato normalizzato e $|x\rangle$ una base di N vettori ortonormali.

Combinando quindi la condizione per F_k e quella per $E_k \leq (2-\sqrt{2})N$ vediamo che $D_k \geq cN$ per N sufficientemente grandi, con c che è una costante sempre minore di $(\sqrt{2} - \sqrt{2-\sqrt{2}})^2 \approx 0.42$.

Dal momento che $D_k \leq 4k^2$ possiamo concludere che

$$k \geq \sqrt{cN/4}$$

e possiamo sintetizzare che, come supposto in precedenza, per avere probabilità di successo di almeno un mezzo nel cercare una soluzione al problema di ricerca abbiamo bisogno di utilizzare l'oracolo $\Omega(\sqrt{N})$ volte in maniera indipendente dalle scelte per U_k.

Appendice C

Alleghiamo l'immagine del circuito utilizzato nella sezione 4.2 anche in questo caso realizzata con QISKit.

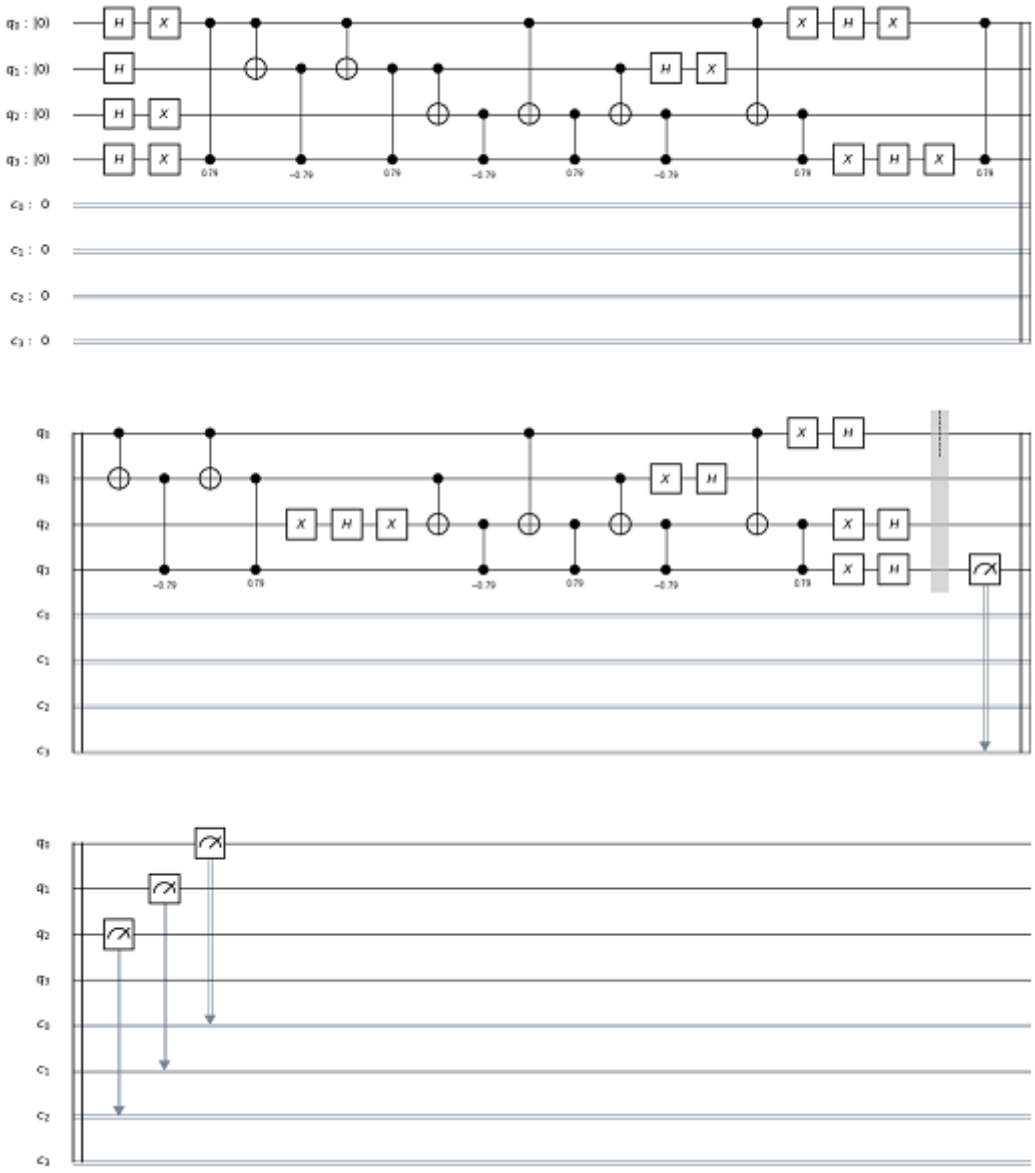

Figura 5.1: Implementazione circuitale utilizzata per l'algoritmo di Grover.

Appendice D

In questa appendice alleghiamo la tabella con alcuni dei valori che si possono ottenere con QISKit riguardanti le caratteristiche tecniche dei qubit presenti nel dispositivo fisico reale a 14 qubits.

	Freq(GHz)	$T_1(\mu s)$	$T_2(\mu s)$
q_0	5.10008	57.72163	19.37949
q_1	5.23865	59.83234	78.72645
q_2	5.033	83.44097	158.16485
q_3	4.89618	47.97239	26.27267
q_4	5.02722	47.06564	36.94458
q_5	5.06715	26.9517	52.43944
q_6	4.92381	62.66735	44.15911
q_7	4.97452	46.23334	76.61967
q_8	4.73978	54.1285	65.00887
q_9	4.96337	23.85105	25.21871
q_{10}	4.94509	58.10349	66.00896
q_{11}	5.00526	63.90961	95.50202
q_{12}	4.76014	59.32794	62.77022
q_{13}	4.96847	22.55154	35.86655

Bibliografia

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2010

[2] QISKit - Open Source Quantum Information Software Kit. https://qiskit.org

[3] IBM Q Frequently Asked Questions. https://quantumexperience.ng.bluemix.net/qx/tutorial?sectionId=full-user-guidepage=000-FAQ 2F000-Frequently$_A sked_Q uestions IBM. Quantum computer composer. https://quantumexperience.ng.bluemix.net/qx/editor$

[4] V. Blomkvist Karlsson, P. Strömberg, *4-qubit Grover's algorithm implemented for the ibmqx5 architecture*, Degree project in Computer Science, KTH Royal Institute of Technology in Stockholm, 2018

[5] C. Figgatt, D. Maslov, K. A. Landsman, N. M. Linke, S. Debnath, C. Monroe, *Complete 3-Qubit Grover Search on a Programmable Quantum Computer*, Nature Communications **8** 1918, (2017)

[6] L.K. Grover, *A fast quantum mechanical algorithm for database search*, Proceedings, 28th Annual ACM Symposium on the Theory of Computing (STOC), 1996, 212-219