Università degli Studi di Padova

DIPARTIMENTO DI FISICA E ASTRONOMIA

Corso di Laurea in Fisica

Tesi di laurea

Equivalenza fra modello circuitale ed adiabatico di computazione quantistica

Candidato:

Luca Mattiazzi

Matricola 1049190

Relatore:

Giuseppe Vallone

Anno Accademico 2014–2015

Indice

1 Introduzione

> *To me quantum computation is a new and deeper and better way to understand the laws of physics, and hence understanding physical reality as a whole*
> -David Deutsch

La meccanica quantistica è uno degli argomenti più contro-intuitivi della fisica tanto che dagli inizi del '900, quando si cominciarono a studiare i primi effetti non classici, trascorsero anni prima di raggiungere una padronanza sufficiente dell'argomento per poterlo sfruttare in strutture più complesse. Così nei primi anni '80, alcuni scienziati del livello di Richard Feyman, David Deutsch e Paul Benioff gettarono le basi per i tanto acclamati computer quantistici.

I motivi che hanno spinto a rivoluzionare i modelli computazionali già esistenti, nonostante fossero ben consolidati, sono essenzalmente tre.
Da un punto di vista pratico, la tendenza a rendere i componenti sempre più piccoli avrebbe portato, prima o poi, all'emergere di effetti quantistici. Questo spinse Paul Benioff, e Richard Feynman in minima parte, a scrivere alcuni dei loro lavori.
Nel frattempo la scienza computazionale cresceva a dismisura, dando vita ad algoritmi sempre più complessi ed originali, arrivando a concepirne alcuni che implicavano processi casuali, ben diversi dalle basi deterministiche dei computer (o *macchine di Turing*) già esistenti. La ricerca di un nuovo modello per algoritmi di questo tipo spinse Deutsch a chiedersi se fosse possibile utilizzare fenomeni quantistici, che possiedono una componente casuale, per poterli realizzare.
Infine Richard Feynman mostrò che un computer quantistico può simulare un computer classico senza problemi, mentre il contrario non è possibile se non con tempi di calcolo eccessivamente lunghi. Questo implicava che per simulare un sistema quantistico, un computer classico avrebbe dovuto affrontare grosse difficoltà, mettendo in evidenza la superiorità del primo.

Le differenze maggiori fra questi due, consistono nel fatto che il modello di computazione classico sfrutta processi deterministici e opera su bit, mentre il modello quantistico deve considerare anche alcuni fenomeni casuali e si applica a *qubit*.

1.1 Qubit

L'oggetto su cui operano gli algoritmi, nel caso classico, è chiamata bit. Si tratta di una variabile che può assumere solo due valori: 0 o 1. Il corrispettivo nell'ambito dell'informazione quantistica è detto invece *qubit* (quantum bit)e seppure abbia un nome molto simile alla sua controparte, esso è concettualmente diverso: si tratta di un vettore in uno spazio complesso bidimensionale. Supponiamo che una base ortonormale di questo spazio sia data

da $|0\rangle$ e $|1\rangle$. Dunque un vettore normalizzato può essere rappresentato da

$$|\psi\rangle = a|0\rangle + b|1\rangle \qquad a, b \in \mathbb{C} \quad t.c. \quad |a|^2 + |b|^2 = 1 \tag{1}$$

Un'altra qualità che distingue questi due oggetti è la loro natura, da una parte deterministica e dall'altra casuale. Ebbene se esaminiamo un bit a piacimento per determinare se esso valga 0 o 1, otterremo una risposta univoca in base alle trasformazioni a cui è stato sottoposto. Se invece proviamo a misurare un qubit per determinarne lo stato in cui si trova, ci verrà restituito $|0\rangle$ con una probabilità $|a|^2$ e $|1\rangle$ con probabilità $|b|^2$.
La natura casuale del qubit può apparirci però anche svantaggiosa: com'è possibile ottenere una soluzione univoca ad un problema, se questa è regolata da un fenomeno casuale? É presto detto: mediante vari processi, come ad esempio trasformazioni unitarie, possiamo rendere $|a|$ o $|b|$ infinitamente vicini a 1 così da poter estrarre con sicurezza la nostra soluzione. Questa caratteristica si rivela essere il vero punto di forza dell'oggetto appena introdotto e della computazione quantistica generale che non è più vincolata a dover operare o con 0 o con 1, ma può lavorare con superposizioni di questi stati.

Se ora volessimo rappresentare uno stato ad n qubits, possiamo esprimerlo come un vettore in uno spazio 2^n-dimensionale. Tenendo sempre come vettori di base $|0\rangle$ e $|1\rangle$ esso sarà rappresentato da

$$|0110010100 \cdots 1001\rangle \tag{2}$$

In generale, uno stato normalizzato in questo spazio può essere scritto in questo modo:

$$\sum_{x=0}^{2^n-1} a_x|x\rangle \tag{3}$$

Se ora dovessimo rappresentare un oggetto composto da n bits classici ci è sufficiente una stringa binaria di dimensione n, mentre per rappresentare un sistema di qubits della stessa dimensione dobbiamo calcolare tutti i 2^n numeri complessi che restituiscono la probabilità che l'insieme di qubits si trovi in quel dato stato, occupando decisamente più memoria. Come detto prima, un computer quantistico utilizza questi vettori, trasformandoli in vario modo, cambiando ad esempio le varie ampiezze a_x tramite delle trasformazioni unitarie (modello circuitale di computazione quantistica). Questo può essere fatto benissimo anche da un qualsiasi PC che utilizziamo tutti i giorni, però è anche da considerare la velocità con cui lo fa. Supponiamo di avere un computer che opera su n=100 qubits. Dunque per rappresentare un tipico stato avremo bisogno di scrivere i vari a_x, ovvero $2^n = 2^{100} \sim 10^{30}$ numeri complessi. Oltre a questo, dobbiamo riuscire ad eseguire una trasformazione su questo oggetto. Appare chiaro come un modello classico di computazione, che si basa su bit, impieghi ben più tempo a svolgere questo conto che non un modello quantistico che fa di questi vettori i suoi elementi costituenti.

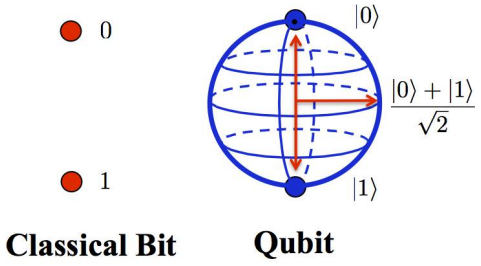

Figura 1: Confronto fra bit e qubit. Il secondo è rappresentato come spin di una particella lungo un asse

É però importante anche capire come possiamo rappresentare fisicamente gli stati così da poter realizzare questo nuovo modello, altrimenti tutto questo rimane un mero esercizio teorico. Un esempio di qubit, molto intuitivo, è lo spin di un elettrone. Per rappresentarlo, utilizzeremo la base di autostati dello spin lungo un asse specifico, ad esempio lo z.
Se questo ha autovalore positivo lo associamo allo stato $|0\rangle$, mentre nell'altro caso lo associamo a $|1\rangle$, come rappresentato in Figura 1. $|\psi\rangle$ in generale sarà una superposizione di questi due stati. Quando però andiamo a misurare il valore dello spin lungo l'asse, esso verrà proiettato in uno dei due autostati compatibilmente con le loro ampiezze, come mostrato prima.

Altri metodi di più facile realizzazione pratica sfruttano stati eccitati di un dato atomo, gli spin nucleari o gli stati quantistici di un fotone.

2 Modello Circuitale

Classicamente abbiamo uno o più bits che subiscono diverse trasformazioni attraverso le note porte logiche AND, OR, NOT, ecc.. per riprodurre un dato algoritmo il cui risultato sarà dato dall'insieme di bits alla fine del processo. Analogamente in questo modello di computazione quantistica abbiamo uno o più qubits soggetti a trasformazioni unitarie (concettualmente svolgono lo stesso ruolo delle porte logiche) per ottenere uno stato finale che decodifichi l'esito del nostro algoritmo.

2.1 Il problema di Deutsch

Uno dei primi esempi che concretizzarono la differenza in velocità di calcolo fra modello classico e quantistico, predetta da Feynman, lo dobbiamo a Deutsch.
Supponiamo di avere una scatola nera che trasforma un singolo bit x applicandovi una funzione f. Non conosciamo cosa accada all'interno di questa scatola, ma dev'essere qualcosa di complicato perché il calcolo per il risultato dura 24 ore. Ci sono 4 possibili funzioni f(x) (perché entrambe f(0) ed f(1)

possono avere valore, 0 o 1) e vorremo conoscere f. In totale occorrono 48 ore per ottenere entrambi i possibili risultati.

Ma non abbiamo tutto questo tempo: dobbiamo trovare la risposta in 24 ore, non 48. In realtà, supponiamo che ci basti sapere se f(x) sia costante (f(0)=f(1)) o meno ($f(0) \neq f(1)$). Comunque il tempo già predetto classicamente non cambia.

Ora supponiamo di avere una scatola nera *quantistica* che computi f(x) e modellizziamola come una trasformazione unitaria che agisca come segue:

$$\mathbf{U}_f \,:\, |x\rangle|y \;\longrightarrow |x\rangle|y \oplus f(x)\rangle \tag{4}$$

ove $\oplus$ è il simbolo di somma binaria. Questo processo cambia valore al secondo qubit se f(x)=1 e lo lascia invariato se f(x)=0. Possiamo risolvere il nostro problema se utilizziamo questa scatola nera 2 volte, ma se supponiamo di impiegare ancora un giorno per produrre l'output non abbiamo avuto alcun miglioramento. Possiamo ottenere il risultato interrogandola solo una volta?(Questo è conosciuto come il Problema di Deutsch)

Essendo la scatola nera un computer quantistico, possiamo decidere di dare in input una *superposizione* di $|0\rangle$ e $|1\rangle$. Se ad esempio il secondo qubit è preparato nello stato $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ avremo

$$\mathbf{U}_f \,:\, |x\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \longrightarrow |x\rangle\frac{1}{\sqrt{2}}(|f(x)\rangle - |1 \oplus f(x)\rangle\,) \tag{5}$$

$$= |x\rangle(-1)^{f(x)}\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{6}$$

così da isolare f in una fase. Ora supponiamo di preparare il primo qubit nello stato $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. In questo caso la scatola nera agirà come segue

$$\mathbf{U}_f \,:\, \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \longrightarrow \tag{7}$$

$$\frac{1}{\sqrt{2}}[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle] \tag{8}$$

Infine possiamo effettuare una misura sul primo qubit che lo proietti nella base

$$|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle) \tag{9}$$

Otterremo $|-\rangle$ se la funzione è costante e $|+\rangle$ altrimenti. Così risolviamo il problema interrogando un'unica volta la scatola nera ed impiegando solo 24 ore, a differenza del caso classico.
Notiamo infine che se per esempio avessi misurato $|x\rangle$ prima della computazione ottenendo il valore $|x_0\rangle$ avrei preparato lo stato in

$$|x_0\rangle|f(x_0)\rangle \tag{10}$$

distruggendo la superposizione creata prima e annullando completamente il vantaggio dell'algoritmo quantistico. Questo porta a grossi problemi quando uno stato interagisce con il mondo esterno, dando luogo alla decoerenza quantistica.

Abbiamo quindi verificato che effettivamente un computer quantistico può rendere i nostri calcoli più rapidi. Ma quanti altri problemi può risolvere più velocemente? E a *quanto* ammonta il suo guadagno rispetto al modello classico? Purtroppo non si ha un metodo sistematico per conoscerlo, è necessario ideare algoritmi nuovi di volta in volta. Di seguito riporteremo due dei risultati più importanti a riguardo, l'algoritmo di Grover e di Shor.

2.2 Algoritmo di Grover

Si tratta di uno dei punti di forza della computazione quantistica perché pur non avendo un guadagno in quanto a velocità esponenziale, risulta molto importante essendo un algoritmo dal largo impiego. Immaginiamo di voler svolgere una ricerca in un database contenente un gran numero di oggetti ($N \gg 1$) e noi ne vogliamo identificare uno in particolare. Come cercare un ago in un pagliaio.

Matematicamente possiamo immaginare il database come un insieme di elementi rappresentati da $x \in 0, 1, 2, \ldots, N-1$ per cui ne esiste esiste uno unico che restituisca $f(x_0) = 1$, mentre tutti gli altri danno come esito 0. Ora se i vari x sono in ordine casuale, è necessario controllarne almeno $N/2$ prima di avere una probabilità $P = 1/2$ di aver trovato x_0. Quello che ci dimostra Grover, è che in ambito quantistico è sufficiente interrogare il database un numero di volte dell'ordine di $\sqrt{N}$ volte.

Questo processo sfrutta un oracolo che sa qual è lo stato x_0 che ci interessa. Come lo formalizziamo?

2.2.1 L'oracolo

Questi sa che delle 2^n possibili stringhe di lunghezza n ne esiste una speciale, ω, l'elemento che stiamo cercando. Sottoponiamo quindi all'oracolo una certa x e otteniamo come risultato

$$f_\omega(x) = 0, \qquad x \neq \omega, \tag{11}$$

$$f_\omega(x) = 1, \qquad x = \omega \tag{12}$$

Si tratta però di un oracolo *quantistico*, perciò può accettare anche superposizioni di diversi stati. Vediamolo come la scatola nera del problema di Deutsch, che applica la seguente trasformazione unitaria:

$$\mathbf{U}_{f_\omega} : |x\rangle|y\rangle \longrightarrow |x\rangle|y \oplus f_\omega(x)\rangle \tag{13}$$

dove $|x\rangle$ è uno stato ad n-qubit mentre $|y\rangle$ è a singolo qubit.

Prendiamo ora $|y\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, così facendo abbiamo:

$$\mathbf{U}_{f_\omega} : |x\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \longrightarrow (-1)^{f_\omega}|x\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{14}$$

Se ora ignoriamo il secondo output, risulta

$$\mathbf{U}_\omega : |x\rangle \longrightarrow (-1)^{f_\omega}|x\rangle \tag{15}$$

che può anche essere visto come

$$\mathbf{U}_\omega = 1 - 2|\omega\rangle\langle\omega| \tag{16}$$

L'oracolo cambia il segno dello stato $|\omega\rangle$ ma agisce trivialmente su tutti gli stati ad esso ortogonali. Geometricamente ha un'interpretazione molto intuitiva: se applichiamo $\mathbf{U}_\omega$ ad un qualsiasi vettore nello spazio di Hilbert 2^n-dimensionale, questi viene riflesso attorno all'iperpiano ortogonale a $|\omega\rangle$.

2.2.2 L'iterazione di Grover

Anzitutto prepariamo il sistema nel seguente stato:

$$|s\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle \tag{17}$$

Ogni elemento del database viene equamente pesato in una superposizione, pertanto questo stato avrà al suo interno anche una componente $|\omega\rangle$, più precisamente

$$|\langle\omega|s\rangle| = \frac{1}{\sqrt{N}} \tag{18}$$

indipendentemente dal valore di ω. Se ora misurassimo $|s\rangle$ avremo una probabilità pari a $1/N$ di trovare lo stato che stiamo cercando. Seguendo l'iterazione di Grover ciò che facciamo è aumentare l'ampiezza di $|\omega\rangle$ e ridurre quella degli $|x \neq \omega\rangle$. Per fare questo, oltre alla riflessione $\mathbf{U}_\omega$ compiuta dall' oracolo implementiamo anche

$$\mathbf{U}_s = 2|s\rangle\langle s| - 1 \tag{19}$$

che conserva $|s\rangle$ ma riflette le componenti ortogonali a quest'ultimo.

Un'iterazione di Grover è data dalla trasformazione unitaria

$$\mathbf{R}_{grov} = \mathbf{U}_s\mathbf{U}_\omega \tag{20}$$

rappresentata graficamente in Figura 2 nella pagina successiva, dove

$$|\langle s|\omega\rangle| = \frac{1}{\sqrt{N}} \equiv \sin(\frac{\theta}{2}) \tag{21}$$

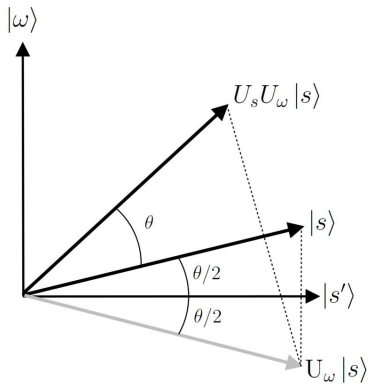

Figura 2: Rappresentazione grafica dell'algoritmo di grover. $|s'\rangle$ rappresenta lo stato ortogonale a $|\omega\rangle$

In tutto la riflessione ruota il vettore di un'angolo θ.
Dopo T iterazioni lo stato in cui ci troveremo formerà un angolo pari a $\frac{\theta}{2}(2T+1)$ rispetto all'asse $|s'\rangle$. Vogliamo ottenere $|\omega\rangle$ con la probabilità massima, misurando lo stato in cui si trova il sistema e ciò accade nel caso in cui l'angolo è vicino a $\pi/2$. Perciò

$$(2T+1)\frac{\theta}{2} \simeq \frac{\pi}{2} \Rightarrow 2T+1 \simeq \frac{\pi}{\theta} \tag{22}$$

Ricordando quindi che $\sin(\theta/2) = \frac{1}{\sqrt{N}}$, per $N \gg 1$ abbiamo

$$\frac{\theta}{2} \simeq \frac{1}{\sqrt{N}} \tag{23}$$

Se dunque scegliamo

$$T = \frac{\pi}{4}\sqrt{N}(1 + O(N^{-\frac{1}{2}})) \tag{24}$$

la probabilità di ottenere $|\omega\rangle$ come risultato di una misura sarà pari a

$$Prob(\omega) = \sin^2((2T+1)\frac{\theta}{2}) = 1 - O(\frac{1}{N}) \tag{25}$$

Concludiamo dunque che sono sufficienti solo $\frac{\pi}{4}\sqrt{N}$ interrogazioni del database (qui rappresentato come scatola di nera) per ottenere l'elemento cercato, con un miglioramento quadratico nel tempo di esecuzione.

2.3 Algoritmo di Shor

Uno fra gli algoritmi più efficienti pensati fin'ora è l'algoritmo di Shor per la fattorizzazione, che presenta uno speed-up esponenziale. Ha una grande importanza in ambito bancario: il protocollo di sicurezza RSA utilizzato ampiamente oggigiorno si basa sulla difficoltà di fattorizzare un numero grande,

compito arduo per un computer classico. Lo potreste aver usato voi stessi, se avete mai inviato il numero della vostra carta di credito attraverso internet.

La grande intuizione di Shor fu quella di tradurre il problema di fattorizzazione di un numero in quello di trovare il periodo di una funzione. Più precisamente, se N è il numero da fattorizzare ed a è un altro numero, scelto casualmente e tale per cui $a < N$, avremo che la seguente funzione

$$f(x) = a^x mod N \tag{26}$$

è periodica, di periodo r.

2.3.1 Trovare il periodo di una funzione

Per questo algoritmo, utilizzeremo la trasformata di Fourier quantistica, definita come:

$$|y\rangle \longrightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i x y/N} |x\rangle \tag{27}$$

analoga alla trasformata di Fourier discreta.
In input diamo 2 qubit, entrambi nello stato $|0\rangle$ e modifichiamo poi il primo qubit trasformandolo in una superposizioni di tutti i possibili stati di x. Ora ci troveremo nello stato

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|0\rangle \tag{28}$$

Al secondo qubit applichiamo U_f già vista per il problema di Deutsch, così da ottenere

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|f(x)\rangle \approx \frac{1}{\sqrt{r2^n}} \sum_{l=0}^{r-1} \sum_{x=0}^{2^n-1} e^{2\pi i x l/r} |x\rangle|\tilde{f}(l)\rangle \tag{29}$$

con $\tilde{f}(x)$ la trasformata di Fourier di f(x), o meglio

$$|\tilde{f}(l)\rangle = \frac{1}{\sqrt{r}} \sum_{x=0}^{r-1} e^{-2\pi i x l/r} |f(x)\rangle \tag{30}$$

A questo punto, applicando l'antitrasformata al primo bit ci resta

$$\frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} |\widetilde{l/r}\rangle|\tilde{f}(l)\rangle \tag{31}$$

e misurando il primo qubit ricavamo l/r, da cui possiamo ottenere il periodo r ripetendo l'algoritmo poche volte. Ora possiamo vedere la fattorizzazione vera e propria

2.3.2 Iterazione di Shor

Scegliamo casualmente un numero x tale per cui si abbia $0 < x < N$. Controlliamo che N e x non abbiano divisori comuni. Se ciò non accade, proseguiamo. Utilizziamo l'algoritmo di *period finding* appena visto per determinare r, intero più piccolo tale per cui

$$x^r \, mod \, N = 1 \tag{32}$$

Se r è pari e $x^r/2 \, mod \, N \neq -1$ allora i fattori saranno dati da $\gcd(x^r/2 - 1, N)$ e $\gcd(x^r/2 + 1, N)$ (con gcd *greatest common divisor*, massimo comun divisore), problema risolvibile con un algoritmo classico in maniera efficiente. Questo accade perché

$$a^r \, mod \, N = 1 \Rightarrow ((a^{r/2})^2 - 1) \, mod \, N = 0 \tag{33}$$

$$(a^{r/2} + 1)(a^{r/2} - 1) \, mod \, N = 0 \tag{34}$$

Ora sicuramente $a^{r/2} - 1$ non divide N, se così fosse il periodo dovrebbe essere r/2 o più piccolo. Imponendo poi che

$$a^{r/2} \, mod \, N \neq -1 \tag{35}$$

abbiamo che N non divide nemmeno $a^{r/2}+1$, pertanto dovrà avere un divisore comune con entrambi i fattori.

Viste le condizioni che abbiamo dovuto imporre per poter svolgere quest'algoritmo, potremo chiederci se queste non siano troppo restrittive, rendendolo utilizzabile un numero di volte irrisorio. Fortunatamente, la probabilità che le condizioni vengano rispettate è

$$P = 1 - \frac{1}{2^m} \tag{36}$$

con m numero di fattori che compongono N, maggiore o uguale ad 1/2.

Abbiamo parlato fin'ora di guadagno esponenziale per questo algoritmo.. ma effettivamente a quanto ammonta? Shor permette di fattorizzare un numero in un tempo $T \sim (\log(N))^3$, quando uno fra i migliori algoritmi classici impiega $T \simeq exp[c(\ln n)^{1/3}(\ln \ln n)^{2/3}]$ (chiamato in inglese number field sieve).
Nella pratica, applicando l'algoritmo classico a numeri molto grandi riusciamo a dare una stima di c (~ 1.9). Ad esempio, un fattore a 65 digit di un numero di 300 digit può essere trovato in un mese impiegando centinaia di computer. Se ora utilizziamo questa legge per capire quanto impiegheremo a fattorizzare un numero di 400 digit, scoprimao che occorrono 10^{10} anni, un tempo totalmente fuori dalla nostra portata. Ora assumiamo di avere un computer quantistico che possa fattorizzare un numero di 130 digit in un mese, come quello classico. Per svolgere l'algoritmo su un numero di 400 digit impiegherebbe solo 3 anni, valore ben più ragionevole della sua controparte classica.

2.4 Debolezze del modello circuitale

Per quanto possa sembrare semplice a livello logico, presenta molte debolezze se si comincia a progettarne una realizzazione pratica.
Per esempio, al classico errore di *bit flip* va ad aggiungersi un potenziale errore di fase. Rilevare e correggere entrambi, contemporaneamente, è un compito tutt'altro che semplice. Tuttavia recentemente si stanno facendo grossi progressi a riguardo.

Un problema ben più grave è dovuto alla *fragilità* degli stati con cui opera un computer quantistico. Questi infatti sono vettori generici nello spazio di Hilbert che se interagiscono anche minimamente con il mondo esterno divengono *entangled* con esso. Ciò causa la perdita dell'informazione riguardo le fasi reciproche fra gli stati che compongono questa superposizione nell'ambiente. Da un vettore nello spazio di Hilbert si passa ad una *miscela statistica* a cui non possiamo più applicare in nostri algoritmi ben congeniati come Grover o Shor.

Questo fenomeno, chiamato decoerenza quantistica, è il più grosso scoglio che il modello circuitale deve affrontare e che resta tutt'ora irrisolto. Potremo limitarlo con un tipo di computazione diversa, ma che porti agli stessi vantaggi?

3 Modello adiabatico

Presenteremo ora un modello atto a risolvere problemi di *soddisfacibilità*, ovvero a verificare che una o un'insieme di condizioni siano verificate. Queste clausole possono essere vere o false a seconda dell'insieme di bits in esame.
Più recente del circuitale, l'*adiabatic quantum computation* si basa sull'evoluzione adiabatica di autostati di un'Hamiltoniana variabile nel tempo.

Per svolgere un algoritmo, si prepara il sistema nello stato fondamentale dell'Hamiltoniana iniziale (di facile realizzazione, che chiameremo H_B). Questa viene fatta variare lentamente, affinché si rimanga nello stato ad energia minore durante tutta la trasformazione (fatto garantito dal Teorema Adiabatico), finché il sistema non è descritto da un'Hamiltoniana finale (H_P) che decodificherà la soluzione al dato problema nel suo stato fondamentale, che andremo infine a misurare.
L'Hamiltoniana totale è data da un'interpolazione lineare fra H_B e H_P come segue

$$H(t/T) = (1 - t/T)H_B + (t/T)H_P \qquad t \in [0, T] \tag{37}$$

Osserviamo facilmente che $H(0) = H_B$ ed $H(T) = H_P$.

Per semplicità poniamo $t/T = s$ ottenendo

$$\widetilde{H}(s) = (1 - s)H_B + s\,H_P \qquad s \in [0, 1] \tag{38}$$

Come possiamo vedere il tempo impiegato a svolgere un dato algoritmo non è più rappresentato dal numero di trasformazioni unitarie a cui il nostro

stato è sottoposto, ma da T, il tempo di durata della trasformazione affinché il teorema Adiabatico sia valido.

3.1 L'Hamiltoniana finale H_P

Se passiamo alla computazione quantistica, i bit z_i vengono sostituiti da qubit $|z_i\rangle$ di spin 1/2, autostati della componente z dell'i-esimo spin. Questi si rappresenteranno come

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \text{e} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{39}$$

Quindi avremo

$$\frac{1}{2}(1 - \sigma_z^{(i)}) \qquad \text{dove} \qquad \sigma_z^{(i)} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{40}$$

Se prendiamo questa come H_P, essa rappresenterà la condizione soddisfatta solo nel caso in cui lo spin ha autovalore positivo, dato che il suo stato fondamentale sarà $|0\rangle$. Possiamo quindi associare ad ogni H_P una clausola, che sarà soddisfatta solo per lo stato fondamentale della stessa. Vedremo più avanti alcuni esempi.

Nel caso ci siano più condizioni, l'hamiltoniana totale sarà data dalla somma di quelle relative alle singole richieste:

$$H_P = \sum_C H_{P,C} \tag{41}$$

3.2 L'Hamiltoniana iniziale H_B

Può sembrare poco rilevante ma in realtà anch'essa influenza pesantemente il tempo necessario ad ottenere una soluzione.
Supponiamo che $H_B^{(i)}$ sia un'Hamiltoniana che agisce su un unico bit, l'i-esimo. Una tipica scelta per essa è

$$H_B^{(i)} = \frac{1}{2}(1 - \sigma_x^{(i)}) \qquad \text{con} \qquad \sigma_x^{(i)} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{42}$$

e quindi

$$H_B^{(i)}|x_i = x\rangle = x|x_i = x\rangle \tag{43}$$

dove

$$|x_i = 0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{e} \qquad |x_i = 1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \tag{44}$$

Vogliamo ora determinare che stato soddisfi una data condizione C. Affinché la computazione avvenga efficientemente sceglieremo

$$H_B = \sum_{i=1}^{n} d_i H_B^{(i)} \tag{45}$$

ove d_i è il numero di clausole in cui è coinvolto il bit i-esimo.
Lo stato fondamentale di H_B è $|x_1 = 0\rangle|x_2 = 0\rangle \cdots |x_n = 0\rangle$ che, riscritto nella base z risulta

$$|x_1 = 0\rangle|x_2 = 0\rangle \cdots |x_n = 0\rangle = \frac{1}{2^{n/2}} \sum_{z_1} \sum_{z_1} \cdots \sum_{z_n} |z_1\rangle|z_2\rangle \cdots |z_n\rangle \tag{46}$$

3.3 Teorema adiabatico

Vediamo ora come possiamo determinare il tempo computazionale T, basandoci sul Teorema in questione.
Un sistema quantistico evolve secondo l'equazione di Schrödinger

$$i\frac{d}{dt}|\psi(t)\rangle = \widetilde{H}(s)|\psi(t)\rangle \tag{47}$$

Usando la base di autostati istantanea definita da $H(t)|n(t)\rangle = E_n(t)|n(t)\rangle$, il generico stato del sistema $|\psi(t)\rangle$ può essere scritto come

$$|\psi(t)\rangle = \sum_n a_n(t) \exp\left(-i\int_0^t E_n(t')dt'\right)|n(t)\rangle \tag{48}$$

Sostituendolo nell'equazione (47), dopo qualche passaggio otteniamo l'espressione per i vari coefficienti a_m:

$$\frac{d}{dt}(a_m e^{-i\gamma_m}) = -\sum_{n\neq m} a_n \frac{\langle m|\dot{H}|n\rangle}{\Delta E_{nm}} e^{-i\gamma_m} \exp\left(-i\int_0^t \Delta E_{nm}(t')dt'\right) \tag{49}$$

con il gap energetico $\Delta E_{nm}(t) = E_n(t) - E_m(t)$ e la fase di Berry

$$\gamma(t) = i\int_0^t dt'\langle n(t')|\dot{n}(t')\rangle \tag{50}$$

Se l'evoluzione è sufficientemente lenta possiamo ottenere le soluzioni alla (49) perturbativamente. Dopo un'integrazione per parti, il contributo al prim'ordine restituisce

$$a_m(t) \approx a_m^0 e^{i\gamma_m(t)} - i\left[\sum_{n\neq m} a_n^0 \frac{\langle m|\dot{H}|n\rangle}{\Delta E_{nm}^2} e^{i\phi_{nm}}\right] \tag{51}$$

ove $\phi_{nm} \in \mathbb{R}$ denota una fase pura.

Osserviamo che se

$$\frac{\langle m|\dot{H}|n\rangle}{\Delta E_{nm}^2} \ll 1 \tag{52}$$

è verificato per tutti tempi e per i diversi $|n\rangle$, il coefficiente del rispettivo stato non cambierà durante tutta la trasformazione. Ciò implica che se si

parte da $|m\rangle$ (ovvero $a_m(0) = 1$), vi si rimane per tutta la durata della computazione.
In particolare vogliamo che il sistema rimanga nello stato fondamentale dell'Hamiltoniana, pertanto l' equazione (52) può essere riscritta come

$$1 \gg \frac{\langle 1|\dot{H}|0\rangle}{\Delta E_{10}^2} \tag{53}$$

e con qualche altro passaggio otteniamo

$$dt \gg \frac{\langle 1|\frac{d\widetilde{H}}{ds}|0\rangle}{\Delta E_{10}^2} ds \tag{54}$$

$$T \gg \int_0^1 \frac{\langle 1|\frac{d\widetilde{H}}{ds}|0\rangle}{\Delta E_{10}^2} ds \tag{55}$$

$$\tag{56}$$

che porta infine a

$$T \gg \frac{D_{max}}{g_{min}^2} \tag{57}$$

ove

$$D_{max} = \max_{0\leq s\leq 1} |\langle 1|\frac{d\widetilde{H}}{ds}|0\rangle| \tag{58}$$

$$g_{min} = \min_{0\leq s\leq 1} (E_1(s) - E_0(s)) \tag{59}$$

Per i problemi che affronteremo D_{max} non assume valori troppo grandi, pertanto in generale $T \sim g_{min}^{-2}$.

3.4 Esempi

3.4.1 Problemi a singolo qubit

Consideriamo un problema in cui la singola condizione sia soddisfatta se e solo se $z_1 = 1$. In questo caso, possiamo prendere

$$H_P = \frac{1}{2}(1 + \sigma_x^{(1)}) \tag{60}$$

Come H_B scegliamo invece

$$H_B^{(1)} = \frac{1}{2}(1 - \sigma_x^{(1)}) \tag{61}$$

che corrisponde ad n=1 e $d_1 = 1$ nella formula (45).

L'Hamiltoniana interpolante $\widetilde{H}(s)$ data dall'equazione (38) ha come autovalori

$$E_{1,0} = \frac{1}{2}(1 \pm \sqrt{1 - 2s + 2s^2}) \tag{62}$$

riportati in Figura 3

Osserviamo che g_{min} non è piccolo, perciò possiamo svolgere la nostra computazione in poco tempo.

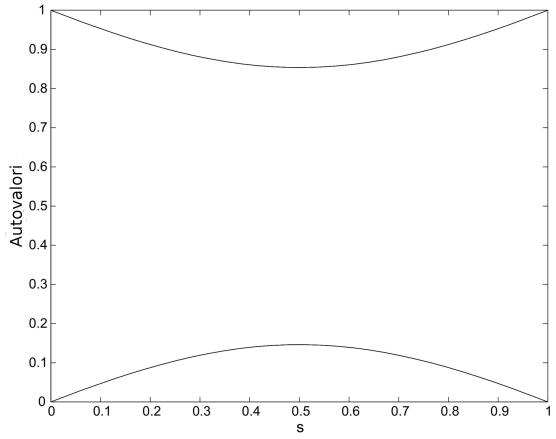

Figura 3: Andamento degli autovalori nel problema a singolo qubit

3.4.2 Problemi a 2 qubit

Un esempio semplice di condizione che coinvolge 2 qubit è che essi siano discordi, soddisfatta dagli stati 01, 10 ma non da 11 e 00. Avendo due possibili soluzioni lo stato fondamentale di H_P sarà

$$\frac{1}{\sqrt{2}}(|z_1 = 0\rangle|z_2 = 1\rangle + |z_1 = 1\rangle|z_2 = 0\rangle) \tag{63}$$

ovvero una superposizione degli stati che soddisfano la condizione. H_B è sempre data dall'equazione (45) con n=2 e $d_1 = d_2 = 1$ mentre H_P è

$$H_P = \frac{1}{2}(1 + \sigma_z^{(1)}\sigma_z^{(2)}) \tag{64}$$

Gli autovalori istantanei di $\widetilde{H}(s)$ sono mostrati in Figura 4

Dal grafico può sembrare che il gap fra i due autovalori più piccoli sia nullo verso la fine, rendendo impossibile l'applicazione del teorema adiabatico. Va però osservato che l'operazione $|z_1\rangle|z_2\rangle \longrightarrow |z_2\rangle|z_1\rangle$ commuta con $\widetilde{H}(s)$. Oltretutto gli stati fondamentali iniziale e finale sono invarianti sotto lo scambio di bit, mentre lo stato che inizialmente si trova ad energia E_1

$$\frac{1}{\sqrt{2}}(|x_1 = 0\rangle|x_2 = 1\rangle - |x_1 = 1\rangle|x_2 = 0\rangle) \tag{65}$$

termina nello stato antisimmetrico

$$\frac{1}{\sqrt{2}}(|z_1 = 0\rangle|z_2 = 1\rangle - |z_1 = 1\rangle|z_2 = 0\rangle) \tag{66}$$

Essendo quindi $\widetilde{H}(s)$ invariante sotto trasformazioni di questo genere non è possibile il passaggio fra due stati a simmetrie diverse, pertanto l'unico gap significativo è $E_2(s) - E_0(s)$

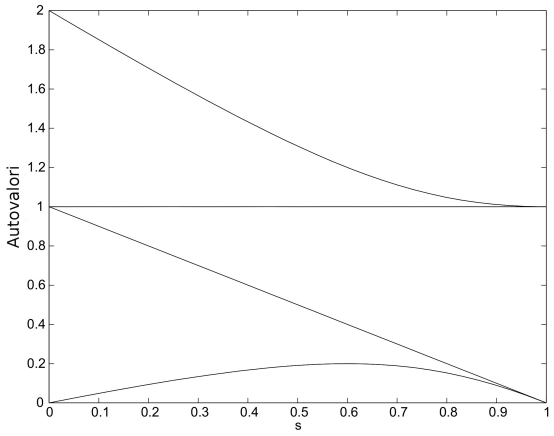

Figura 4: I 4 autovalori di $\widetilde{H}(s)$ associati al problema dei 2 bit discordi.

3.5 Algoritmo di Grover

Andiamo ora ad indagare se il modello adiabatico di computazione quantistica riesce effettivamente a riprodurre gli ottimi risultati ottenuti da quello circuitale.

In questo algoritmo la condizione è soddisfatta da un unico stato $|w\rangle = |w_1\rangle|w_2\rangle....|w_n\rangle$ sconosciuto. Questi dovrà essere lo stato fondamentale della nostra Hamiltoniana finale, pertanto avremo che

$$H_P|z\rangle = \begin{cases} |z\rangle & \text{se } z \neq w \\ 0 & \text{se } z = w \end{cases} \tag{67}$$

$$H_P = 1 - |w\rangle\langle w| \tag{68}$$

dove per brevità si è posto $|z\rangle = |z_1\rangle|z_2\rangle....|z_n\rangle$
Come al solito H_B sarà tale per cui lo stato a minore energia corrisponda a $|x\rangle = |0\rangle|0\rangle....|0\rangle$.
Combinando queste due Hamiltoniane risulta

$$\widetilde{H}(s) = (1-s)\sum_{j=1}^{n}\frac{1}{2}(1-\sigma_x^{(j)}) + s(1-|w\rangle\langle w|) \tag{69}$$

Osserviamo però che $\widetilde{H}(s)$ è collegata a

$$\widetilde{H}(s) = (1-s)\sum_{j=1}^{n}\frac{1}{2}(1-\sigma_x^{(j)}) + s(1-|0\rangle\langle 0|) \tag{70}$$

tramite una semplice trasformazione unitaria. Ciò rende gli spettri delle due Hamiltoniane equivalenti, pertanto sarà sufficiente studiare la seconda per ottenere una stima su g_{min} valida anche per la prima.

Entrambi gli stati fondamentali sono simmetrici per scambio di qubit (sono tutti $|0\rangle$), pertanto possiamo restringere la nostra indagine al sottospazio (n+1) dimensionale degli stati simmetrici anziché lavorare nel classico spazio di dimensione 2^n
Conviene anche definire questi stati in termini di spin totale.
Definito $\overrightarrow{S} = (S_x, S_y, S_z)$ come

$$S_a = \frac{1}{2}\sum_{j=1}^{n} \sigma_a^{(j)} \tag{71}$$

per a = x,y,z, lo stato simmetrico ha $\overrightarrow{S}^2$ pari a $\frac{n}{2}(\frac{n}{2}+1)$, con $\overrightarrow{S}^2 = \overrightarrow{S}_x^2 + \overrightarrow{S}_y^2 + \overrightarrow{S}_z^2$.

Possiamo riscrivere i vari stati come autostati di S_x o di S_z:

$$S_a|m_a = m\rangle = m|m_a = m\rangle \qquad m = -\frac{n}{2}, -\frac{n}{2}+1, ..., \frac{n}{2} \tag{72}$$

ove non abbiamo indicato l'indice di spin totale in quanto non cambia mai. Riscriviamo ora $|z\rangle$ in termini di questi nuovi stati. Avremo che

$$|m_z = \frac{n}{2} - k\rangle = \left(\begin{array}{c} n \\ k \end{array} \right)^{-\frac{1}{2}} \sum_{z_1+z_2+\cdots+z_n=k} |z_1\rangle|z_2\rangle \cdots |z_n\rangle \tag{73}$$

Per k = 0,1,...,n. In particolare

$$|m_z = \frac{n}{2}\rangle = |z = 0\rangle \tag{74}$$

Ora possiamo riscrivere

$$\widetilde{H}(s) = (1-s)\left[\frac{n}{2} - S_x\right] + s\left[1 - |m_z = \frac{n}{2}\rangle\langle m_z = \frac{n}{2}|\right] \tag{75}$$

Abbiamo cosi ridotto $\widetilde{H}(s)$ ad una matrice (n+1) dimensionale, più semplice da analizzare.
Vorremo risolvere

$$\widetilde{H}(s)|\psi\rangle = E|\psi\rangle \tag{76}$$

per i due autovalori più piccoli, così da poter dare una stima del tempo T necessario per svolgere la computazione. Moltiplicando l'equazione precedente per $\langle m_x = \frac{n}{2} - r|$ otteniamo

$$\begin{aligned}[s + (1-s)r]\langle m_x = \frac{n}{2} - r|\psi\rangle - s\langle m_x = \frac{n}{2} - r|m_z = \frac{n}{2}\rangle\langle m_z = \frac{n}{2}|\psi\rangle \\ = \frac{n}{2}|\psi\rangle = E\langle m_x = \frac{n}{2} - r|\psi\rangle\end{aligned}$$

Per comodità sostituiamo E con una variabile λ per cui $E = s + (1-s)\lambda$ e otteniamo

$$\frac{1-s}{s}\langle m_x = \frac{n}{2} - r|\psi\rangle = \frac{1}{r\lambda}\langle m_x = \frac{n}{2} - r|m_z = \frac{n}{2}\rangle\langle m_z = \frac{n}{2}|\psi\rangle \tag{77}$$

Moltiplichiamo per $\langle m_x = \frac{n}{2} - r|m_z = \frac{n}{2}\rangle$ e sommiamo su r, così da avere

$$\frac{1-s}{s} = \sum_{r=0}^{n} \frac{1}{1-\lambda} P_r \tag{78}$$

con

$$P_r = |\langle m_x = \frac{n}{2} - r|m_z = \frac{n}{2}\rangle|^2 \tag{79}$$

Utilizzando l'equazione (73) con $k = 0$ e l'analoga con x al posto di z si ottiene

$$P_r = \frac{1}{2^n} \begin{pmatrix} n \\ k \end{pmatrix} \tag{80}$$

che ha (n+1) soluzioni.

Facendo variare s tra 0 ed 1 otteniamo una radice per $\lambda < 0$, una per $0 < \lambda < 1$, un'altra $1 < \lambda < 2$ e così via, di cui noi interessano le prime due. Mostriamo che esiste un certo valore di s per cui in entrambi i casi λ è molto vicine a 0, coincidendo con il punto a gap minimo.
Il termine di sinistra nella formula (78) spazia fra tutti i valori positivi al variare di s fra 0 a 1, quindi possiamo prendere $s = s^*$ per cui

$$\frac{1-s^*}{s^*} = \sum_{r=1}^{n} \frac{P_r}{r} \tag{81}$$

Per s=s* l'equazione agli autovalori (sempre la (78)) diventa:

$$\frac{P_0}{\lambda} = \sum_{r=1}^{n} P_r \frac{\lambda}{r(r-\lambda)} \tag{82}$$

Sappiamo che $P_0 = 2^{-\frac{n}{2}}$. Effettuiamo un cambio di variabile, ponendo $\lambda = 2^{-\frac{n}{2}}u$. Otteniamo dunque

$$\frac{1}{u} = \sum_{r=1}^{n} P_r \frac{u}{r(r - 2^{-\frac{n}{2}}u)} \tag{83}$$

Essendo $2^{-\frac{n}{2}} \ll 1$, possiamo trascurare il secondo termine a denominatore così da avere infine

$$\frac{1}{u} \approx \sum_{r=1}^{n} \frac{P_r}{r^2} \tag{84}$$

che restituisce

$$\lambda \approx \pm \left(\sum_{r=1}^{n} \frac{P_r}{r^2} \right)^{-\frac{1}{2}} 2^{-n/2} \tag{85}$$

Sapendo che

$$g_{min} \approx 2(1 - s^*) \left(\sum_{r=1}^{n} \frac{P_r}{r^2} \right)^{-\frac{1}{2}} \cdot 2^{-n/2} \tag{86}$$

e

$$\sum_{r=1}^{n} \frac{P_r}{r} = \frac{2}{n} + O\left(\frac{1}{n^2}\right) \tag{87}$$

$$\sum_{r=1}^{n} \frac{P_r}{r^2} = \frac{4}{n^2} + O\left(\frac{1}{n^3}\right) \tag{88}$$

otteniamo infine

$$g_{min} \simeq 2 \cdot 2^{-\frac{n}{2}} \tag{89}$$

Date le condizioni imposte dal teorema adiabatico, avremo che $T \sim 2^n$, inaspettatamente, non è presente alcuna accelerazione rispetto al caso classico. Cambiamo scelta di H_B, sperando di ottenere un risultato migliore e poniamo

$$H_B = 1 - |\psi_0\rangle\langle\psi_0| \tag{90}$$

Ove $|\psi_0\rangle$ è la superposizione di tutti gli stati di base

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^{N-1} |i\rangle \tag{91}$$

Con questo accorgimento, il gap diventa

$$g(s) = \sqrt{1 - 4(1 - \frac{1}{N})s(1 - s)} \tag{92}$$

Con un $g_{min} = \frac{1}{\sqrt{N}}$ a $\frac{s}{2}$ come mostrato dal grafico in Figura 5.

Dunque non si rileva alcuna accelerazione significativa. Possibile non ci sia modo di rendere il modello adiabatico efficiente tanto quanto quello circuitale?

3.5.1 Condizioni di trasformazione localmente

Fino ad ora abbiamo sempre mantenuto la velocità della variazione ($\frac{ds}{dt}$) costante costringendo il sistema a non abbandonasse mai lo stato fondamentale di $\widetilde{H}(s)$. Ma la probabilità di avere un passaggio ad un altro autostato è costante lungo tutto il percorso? No, possiamo assumere che quest'ultima sia

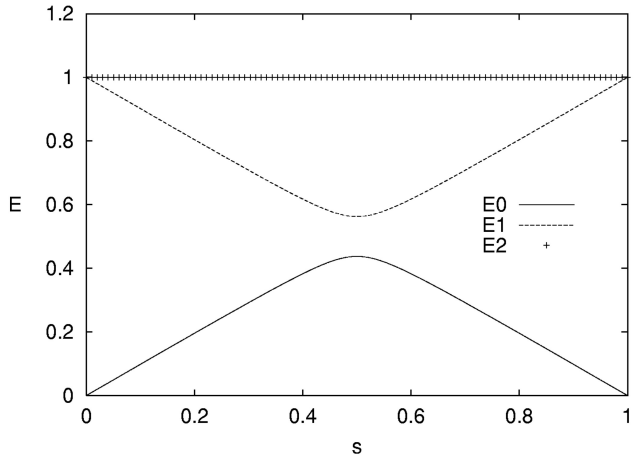

Figura 5: Autovalori dell'Hamiltoniana dipendente dal tempo $\widetilde{H}(s)$, per l'algoritmo di Grover nel caso di N=64.

proporzionle al gap, che è variabile e l'unico punto veramente critico è s_{min} come si può vedere nel grafico. Proviamo quindi a riscrivere le condizioni imposte su T con $\frac{ds}{dt}$ variabile nel tempo.
Essendo

$$|\langle 0;T|\psi(T)|^2 \geq 1-\varepsilon^2 \tag{93}$$

Con

$$\frac{D_{max}}{g_{min}^2} \leq \varepsilon \tag{94}$$

Riscrivendo D_{max}

$$D_{max} = \max_{0\leq t\leq T} |\langle 1|\frac{dH}{dt}|0\rangle| \leq \max_{0\leq t\leq T} |\langle 1|\frac{d\widetilde{H}}{ds}|0\rangle||\frac{ds}{dt}| \tag{95}$$

Otteniamo la seguente condizione:

$$|\frac{ds}{dt}| \leq \varepsilon \frac{g^2}{|\langle 1|\frac{d\tilde{H}}{ds}|0\rangle|} \tag{96}$$

Ora l'elemento $|\langle 1|\frac{d\widetilde{H}}{ds}|0\rangle|$ è bloccato da

$$|\langle 1|\frac{d\widetilde{H}}{ds}|0\rangle| \leq 1 \tag{97}$$

Prendiamo quindi

$$\frac{ds}{dt} = \varepsilon g^2(s) = \varepsilon[1-4\frac{N-1}{N}s(1-s)] \tag{98}$$

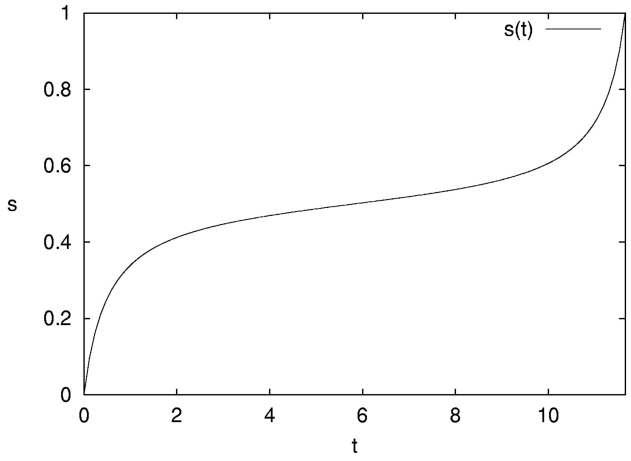

Figura 6: Andamento della funzione di evoluzione temporale s(t) nel caso di N=64. L'evoluzione adiabatica globale apparrirebbe come una retta che congiunge i punti s(0)=0 ed s(T)=1.

con $\varepsilon \ll 1$. Dopo l'integrazione risulta

$$t = \frac{1}{2\varepsilon}\frac{N}{\sqrt{N-1}}[\arctan[\sqrt{N-1}(2s-1)] + \arctan[\sqrt{N-1} \tag{99}$$

Invertendo la funzione otteniamo s(t) come in Figura 6.

Come si può osservare, H(t) cambia rapidamente se il gap è grande, e rallenta in prossimità di g_{min}.
Ricaviamo ora il tempo totale di calcolo, ad s=1

$$T = \frac{1}{\varepsilon}\frac{N}{\sqrt{N-1}}\arctan\sqrt{N-1} \tag{100}$$

Che per $N \gg 1$ restituirà

$$T \simeq \frac{\pi}{2\varepsilon}\sqrt{N} \tag{101}$$

Otteniamo nuovamente lo speed up quadratico presente anche nel modello circuitale migliorando il nostro precedente risultato. Osserviamo che $T \sim g_{min}^{-1}$

Non si tratta di una casualità. Quando infatti abbiamo imposto la condizione data dall'equazione (57) abbiamo richiesto che non si abbandonasse mai lo stato fondamentale, mentre a noi basta che il sistema vi si trovi al termine della computazione.
Possiamo rilassare la restrizione suggerita dal teorema adiabatico prendendo

$$\frac{ds}{dt} = \Delta E_{10}(s)h(s) \tag{102}$$

$\Delta E(s) =$	$\sqrt{(s-1/2)^2+g_{min}^2}$	$\sqrt{(s-1/2)^4+g_{min}^2}$
d=-1	g_{min}^{-2}	$g_{min}^{-3/2}$
d=0	$g_{min}^{-1}\ln(g_{min}^{-2}$	g_{min}^{-1}
$d \geq 1$	g_{min}^{-1}	g_{min}^{-1}

Tabella 1: Andamento di T per gap di diversa struttura, al variare della velocità d'interpolazione(prima colonna. Nella migliore delle ipotesi il tempo di computazione è dell'ordine di g_{min}^{-1}

con $h(t) \geq 0$. In questo caso infatti avremo che

$$a_1(1)e^{-i\gamma_1(1)} = -\int_0^1 ds\, a_0(s)e^{-i\gamma_1(s)} \frac{\langle 1|\frac{d\tilde{H}}{ds}|0\rangle}{\Delta E_{10}} \exp\left(-i\int_0^s \frac{ds'}{h(s'}\right) \quad (103)$$

come mostrato in [7]
Se scegliamo adeguatamente h(s) possiamo rendere l'esponente piccolo a piacere e dunque annullare l'ampiezza a_1 indesiderata.
Ad esempio per un generico gap della forma

$$\Delta E_{10}(s) = [(s-s_{min})^{2a} + g_{min}^b]^{1/b} \quad (104)$$

$h(s) = \alpha_d \Delta E^d$ (d coefficiente) rappresenta una buona scelta. Per $2a(d+1)/b > 1$, si può facilmente mostrare che $\alpha_d = O(T\, g_{min}^{d+1-b/2a})$ soddisfa queste nuove condizioni con $T = O(g_{min}^{-1})$. Nella Tabella 1 si può vedere l'influenza della scelta di d in due diversi modelli di gap.

In Figura 7 osserviamo questo modello applicato all'algoritmo di Grover. Il grafico in alto rappresenta appieno la filosofia di questa nuova condizione: lo stato del sistema abbondana per per gran parte lo stato fondamentale a metà computazione, per poi farvi ritorno a fine trasformazione.

Vediamo quindi che le condizioni di evoluzione adiabatica *locale* portano ad un miglioramento significativo nella stima del tempo di computazione, in quanto $T \sim g_{min}^{-1}$.

4 Equivalenza fra modello adiabatico e circuitale di computazione quantistica

Il modello Adiabatico, almeno nel caso dell'algoritmo di Grover, sembra equivalente a quello circuitale e non c'è apparente motivo per sospettare una possibile inferiorità del primo rispetto al secondo. Ma dunque quanto velocemente l'AQC è in grado di simulare il modello standard?
Per rispondere a questa domanda utilizzeremo alcuni strumenti fornitici

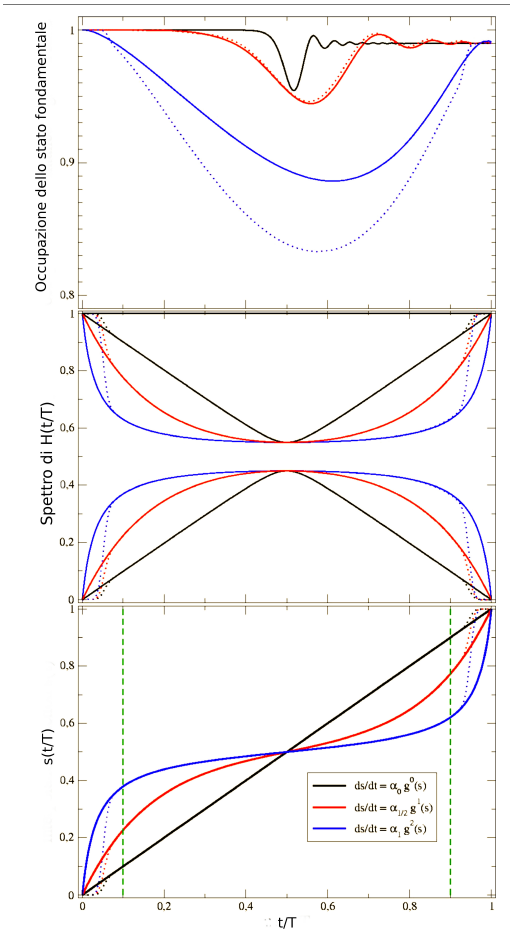

Figura 7: Evoluzione della funzione d'interpolazione (grafico in basso), dello spettro (centrale) e dell'occupazione istantanea dello stato fondamentale (grafico in alto) per il problema di Grover con N=100 stati. Le linee tratteggiate rappresentano la curva interpolante di classe C^∞ unita con una funzione di test.

da un'altro tipo di computazione quantistica, la Ground State Quantum Computation (GSQC).

4.1 Ground state quantum computation

Anziché descrivere un qubit come un sistema quantistico con due stati di base a cui sono applicate N trasformazioni unitarie, lo si rappresenta come un sistema quantistico indipendente dal tempo che si sviluppa in uno spazio di Hilbert 2(N+1) dimensionale. Questo modello fu ideato proprio per sopperire alle debolezze del modello circuitale. Basandosi su uno stato costante nel tempo risulta infatti molto più robusto nei confronti della decoerenza quantistica.
Per meglio comprendere questo nuovo approccio, immaginiamo di avere un elettrone che possa stare solo in due cellette, una a destra ed una a sinistra. Pur essendo una soluzione poco pratica da realizzare, da un'idea semplice ed intuitiva di cosa sia un qubit.
Durante la computazione lo stato del sistema attraverserà N evoluzioni temporali unitarie U_j (in questo caso matrici 2 X 2) che modificheranno lo stato dell'elettrone in accordo con

$$|\psi(t_j)\rangle = U_j|\psi(t_{j-1})\rangle \tag{105}$$

Supponiamo che l'elettrone si trovi nella cella di destra a t=0, avremo allora

$$|\psi(t_0)\rangle = |0\rangle \tag{106}$$

Viene ora applicata U_1 che porta il qubit in

$$|\psi(t_1) = a_1|0\rangle + b_1|1\rangle \tag{107}$$

Applicandovi altre N trasformazioni U_j, a e b assumeranno in tutto N+1 valori diversi.

Per riprodurre questa evoluzione temporale in un approccio indipendente però dal tempo dovremo conservare tutte le 2(N + 1) ampiezze. Possiamo immaginare di avere, anziché un elettrone con due possibili posizioni che attraversa N+1 computazioni, una particella condivisa fra 2(N+1) stati quantistici come mostrato in Figura 8. L'ampiezza data in ogni posizione sostituirà quella di ogni passo temporale fatto durante l'algoritmo.
Supponiamo di scrivere i diversi stati di base come $|0_0\rangle, |1_0\rangle, .., |0_i\rangle|1_i\rangle, .., |0_N\rangle, |1_N\rangle$.
Il sistema può essere descritto da

$$\begin{bmatrix} \langle 0_0|\Psi\rangle \\ \langle 1_0|\Psi\rangle \\ \langle 0_1|\Psi\rangle \\ \langle 1_1|\Psi\rangle \\ \vdots \\ \langle 0_N|\Psi\rangle \\ \langle 1_N|\Psi\rangle \end{bmatrix} = \frac{1}{\sqrt{N-1}} \begin{bmatrix} \langle 0_0|\psi(t_0)\rangle \\ \langle 1_0|\psi(t_0)\rangle \\ \langle 0_1|\psi(t_1)\rangle \\ \langle 1_1|\psi(t_1)\rangle \\ \vdots \\ \langle 0_N|\psi(t_N)\rangle \\ \langle 1_N|\psi(t_N)\rangle \end{bmatrix} = \frac{1}{\sqrt{N-1}} \begin{bmatrix} \begin{bmatrix} \langle 0_0|\psi(t_0)\rangle \\ \langle 1_0|\psi(t_0)\rangle \end{bmatrix} \\ U_1 \begin{bmatrix} \langle 0_1|\psi(t_0)\rangle \\ \langle 1_1|\psi(t_0)\rangle \end{bmatrix} \\ \vdots \\ U_N \dots U_1 \begin{bmatrix} \langle 0_N|\psi(t_0)\rangle \\ \langle 1_N|\psi(t_0)\rangle \end{bmatrix} \end{bmatrix}$$

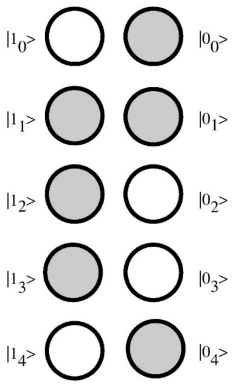

Figura 8: Esempio di stato fondamentale in gsqc, con N04, dove a puro scopo illustrativo lo stato è preso come $[1\ 0\ \sqrt{1/2}\ \sqrt{1/2}\ 0\ 1\ 0\ 1\ 1\ 0]^\dagger/\sqrt{5}$ e i pallini più scuri indicano un ampiezza non nulla

Rappresentiamolo ora formalmente.

Siano $c^\dagger_{j,x}$ e $c_{j,x}$ operatori di creazione e distruzione fermionici, con $j \in \{0, 1, \ldots, N+1\}$ indice che descrive il corrispondente step temporale ed $x \in \{0, 1\}$ che indica la posizione dell'elettrone di volta in volta.
Il sistema avrà come stati di base i vari $c^\dagger_{i,x}|vac\rangle$.
Raggruppiamo gli operatori in covettori $C^\dagger_i = [c^\dagger_{i,0} c^\dagger_{i,1}]$. Con questo accorgimento risulta

$$|\psi(t_0)(s)\rangle = C^\dagger_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} |vac\rangle = c^\dagger_0 |vac\rangle = |0_0\rangle \tag{108}$$

Ne segue che lo stato complessivo sarà dato da:

$$|\Psi^N\rangle = (C^\dagger_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + C^\dagger_1 U_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \cdots + C^\dagger_N U_N \cdots U_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix}) |vac\rangle \tag{109}$$

Se ora prendiamo l'Hamiltoniana:

$$H = \varepsilon \begin{bmatrix} I & -U^\dagger_1 & & & \\ -U_1 & 2I & -U^\dagger_2 & & \\ & -U_2 & 2I & -U^\dagger_3 & \\ & & & \ddots & -U^\dagger_N \\ & & & -U_N & I \end{bmatrix} \tag{110}$$

ove ε è una costante, è facile mostrare che $H|\Psi\rangle = 0$. Formalmente si scriverà come $H = \sum_{i=1}^N h^j(U_j)$ con

$$h^j(U_j) = \varepsilon [C^\dagger_j - C^\dagger_{j-1} U^\dagger_j][C_j - U_j C_{j-1}] \tag{111}$$

Abbiamo così prodotto uno stato $|\Psi\rangle$ che rappresenta un qualsiasi algoritmo standard e che risulta essere lo stato fondamentale di un'Hamiltoniana conosciuta.

Non ci resta che prendere questa come H_P del nostro problema.

4.2 Algoritmi a singolo qubit

Una volta costruita l'Hamiltoniana totale e calcolato g_{min} possiamo risalire al tempo computazionale necessario per il modello adiabatico a simulare quello circuitale, mediante la relazione precedentemente ricavata $T \sim g_{min}^{-1}$. Scegliamo dunque

$$\widetilde{H}(s) = \sum_{i=j}^{N} h^j(\lambda(s)U_j) \tag{112}$$

con $\lambda : s \in [0,1] \to [0,1]$ tale per cui $\lambda(0) = 0$ e $\lambda(1) = 1$.
Se $\lambda = 0$, allora

$$h^i(0) = \varepsilon C_j^\dagger C_j \tag{113}$$

mentre per $\lambda = 1$ avremo

$$\widetilde{H}(1) = \sum_{j=1}^{N} h^j(U_j) \tag{114}$$

Lo stato ad energia minore dell'Hamiltoniana sarà:

$$|\Psi^j(s)\rangle = [1 + C_j^\dagger(\lambda(s)U_j)C_{j-1}]|\Psi^{j-1}(s)\rangle \tag{115}$$

Intuitivamente il termine j-esimo è costruito distruggendo quello alla riga j-1 e creandone uno nuovo nella riga successiva.
Inizialmente ci troviamo nello stato $|\Psi^0\rangle$, poi, facendo aumentare gradualmente $\lambda(s)$, gli elementi di tunneling della matrice diventano sempre più rilevanti e via via tutti gli altri stati $c_{j,x}^\dagger|vac\rangle$ si occupano.
Per poter calcolare gli autovalori dell'Hamiltoniana è necessario risolvere $D_{n+1}^2 = 0$ ove

$$D_{N+1}^2 \equiv det\left(\frac{H(s)}{\varepsilon} - \frac{E}{\varepsilon}\right) \tag{116}$$

Per semplificare i calcoli, applichiamo una trasformazione unitaria agli operatori C e $C^\dagger$, così da avere

$$\widetilde{C}_j = (U_j^\dagger \cdots U_1^\dagger)C_j \tag{117}$$

Ciò trasforma la nostra matrice in

$$\widetilde{H}(s) = \sum_{i=1}^{N} h^i(\lambda I) \tag{118}$$

Scrivendo $\widetilde{H}(s)$ troviamo la relazione iterativa

$$D_{N+1} = (1 + \lambda^2 - E/\varepsilon)D_N - \lambda^2 D_{N-1} \tag{119}$$

La soluzione di $D_{N+1} = 0$ identifica il gap energetico per il processo a singolo qubit. Otteniamo $E_{0,s} = 0$ e

$$E_{n,s} = \varepsilon \left[(1 - \lambda(s))^2 + 2\lambda(s) \left(1 - \cos \frac{\pi n}{N+1}) \right) \right] \tag{120}$$

per $n = 1, \ldots N$. Minimizzando quest'espressione rispetto a λ nel caso in cui n=1, otteniamo $\lambda_{min} = \cos \frac{\pi}{N+1}$. L'energia del primo stato eccitato è

$$E_{1,s} \geq \varepsilon \sin^2 \frac{\pi}{N+1} \tag{121}$$

che per $N \gg 1$ ci restituisce $E_{1,s} \sim O(\frac{1}{N^2})$. Essendo $E_{0,s} = 0$ avremo $g_{min} \sim 1/N^2$ e dunque T sarà dell'ordine di N^2. Risulta dunque che per algoritmi a singolo qubit il modello adiabatico simuli efficientemente il modello circuitale.

4.3 Algoritmi a più qubits

Vogliamo però studiare l'equivalenza fra questi due tipi di computazione per modelli un po' più complessi.
La relazione ricorsiva data dall'equazion (115) si generalizza facilmente per M qubits non interagenti:

$$|\Psi^j(s)\rangle = \Pi_{A=1}^{M}(1 + C_{A,j}^{\dagger}(\lambda U_{A,j})C_{A,j-1})|\Psi^{j-1}(s)\rangle \tag{122}$$

ove

$$|\Psi^0(s)\rangle = \Pi_{A=1}^{M} C_{A,0}^{\dagger} \begin{bmatrix} 1 \\ 0 \end{bmatrix} |vac\rangle \tag{123}$$

L'Hamiltoniana del caso a più qubits è semplicemente la somma delle diverse Hamiltoniane a singolo qubit e l'evoluzione adiabatica, nel caso in cui non ci siano interazioni, è identica al caso con un singolo qubit molto poco interessante.
Ora permettiamo a due qubits di interagire tramite una porta logica qualsiasi, ad esempio un Controlled NOT (ciò che fa questa porta logica è trasformare il secondo qubit nel suo opposto, se il primo è nello stato $|1\rangle$).
Supponiamo che i qubit A e B abbiano un'interazione allo step j. Anziché applicare le Hamiltoniane non interagenti a $|\Psi^{j-1}(s)\rangle$ scriveremo

$$\begin{aligned} |\Psi^j(s)\rangle = (I + c_{A,j,0}^{\dagger}\lambda c_{A,j-1,0} C_{B,j}^{\dagger}(\lambda I)C_{B,j-1} + \\ c_{A,j,1}^{\dagger}\lambda c_{A,j-1,1} C_{B,j}^{\dagger}(\lambda \sigma_x)C_{B,j-1})|\Psi^{j-1}(s)\rangle \end{aligned}$$

in cui al posto dell'identità viene applicata la porta NOT , rappresentata da σ_x, solo quando il primo qubit si trova nello stato $|1\rangle$.
$\widetilde{H}(s)$ diventerà

$$h^j_{A,B}(\lambda, CNOT) = h^j_{A,B}(ID) + h^j_{A,B}(N) + h^j_{A,B}(P) \tag{124}$$

ove

$$\begin{aligned} h^j_{A,B}(ID) = \varepsilon(C_{B,j}c_{A,j,0} - \lambda^2 C_{B,j-1}c_{A,j-1,0})^\dagger \\ (C_{B,j}c_{A,j,0} - \lambda^2 C_{B,j-1}c_{A,j-1,0}) \\ h^j_{A,B}(N) = \varepsilon(C_{B,j}c_{A,j,1} - \lambda^2 \sigma_x C_{B,j-1}c_{A,j-1,1})^\dagger \\ (C_{B,j}c_{A,j,1} - \lambda^2 \sigma_x C_{B,j-1}c_{A,j-1,1}) \end{aligned}$$

sono il corrispettivo a due particelle dell'Hamiltoniana non interagente $h^j_A(\lambda I)$ e del NOT gate $h^j_A(\lambda\sigma_x)$.
La terza componente

$$h^j_{A,B}(P) = \varepsilon \sum_{i<j,k\geq j} C^\dagger_{A,i}C_{A,i}C^\dagger_{B,k}C_{B,k} + C^\dagger_{A,k}C_{A,k}C^\dagger_{B,i}C_{B,i} \tag{125}$$

crea una barriera di energia che impedisce cambi di fase indesiderati nelle posizioni diverse dalla j-esima.

Per comprendere gli effetti della porta logica sul gap, consideriamo prima il caso per M = 2 qubits con un singolo CNOT allo step j. Per facilitarne l'analisi, scomponiamo $\widetilde{H}$ in $\widetilde{H}_0$ e $\widetilde{H}_1$ che rappresentano rispettivamente l'Hamiltoniana del singolo qubit e quella dell'interazione mediante la porta CNOT. Conosciamo tutti gli autostati di H_0 dall'analisi precedentemente fatta, in cui risulta che $g_{min} \sim N^{-2}$.

Valutiamo prima gli stati dei qubit senza la porta logica CNOT. In questo caso abbiamo due regioni disgiunte: dallo step 0 al j-1 e dal j-esimo all'N. Dato che il nostro elettrone occuperà una di queste regioni il cui spettro è quello di un qubit non interagente, l'energia del primo stato eccitato sarà dell'ordine di $1/(N+1)^2$ che è relativamente elevata. Se trascuriamo questi stati ad alte energie, solo i (doppiamente degeneri) stati fondamentali delle regioni in considerazione potranno contribuire significativamente allo stato di ogni qubit, che può effettivamente occuparme quattro diversi, portando in totale ad un sistema che vive in uno spazio di Hilbert 16 dimensionale.

Questa base ha il vantaggio di rendere più facile la diagonalizzazione dell'Hamiltoniana della porta CNOT e dunque il calcolo dei suoi autovalori. Ne risulta uno stato fondamentale (di degenerazione 4) ad energia 0, un primo stato eccitato ad energia $\frac{\varepsilon}{j(N-j+1)}$ (di degenerazione 8) ed uno stato doppiamente eccitato di energia $\frac{\varepsilon}{(N-j+1)^2} + \frac{\varepsilon}{j(N-j+1)} + \frac{\varepsilon}{j^2}$ con degenerazione 4. Dunque $g_{min} \sim 1/N^2$.
Che relazione esiste fra questi autostati nel sottospazio 16 dimensionale con

gli effettivi valori di energia nello spazio generale? Gli stati degeneri ad energia minima corrispondono a quelli del sistema totale, mentre i primi stati eccitati comportano un limite superiore al gap nello spazio complessivo. Da questi abbiamo che $E_{upper} \sim 1/N^2$.

É però importante stabilire un limite inferiore del gap per poter determinare T nel peggior caso possibile.

Proviamo dunque per assurdo che

$$\langle \Psi | \widetilde{H} | \Psi \rangle < \frac{\alpha}{(N+1)^4} \tag{126}$$

Supponiamo che $|\psi\rangle$ sia qualche stato a due particelle ortogonale allo stato fondamentale. Assumiamo che

$$\langle \psi | \widetilde{H} | \psi \rangle < \frac{\alpha}{(N+1)^4} \equiv E_{lower} \tag{127}$$

Consideriamo $|\psi\rangle$ nella base di $\widetilde{H}_0$

$$|\psi\rangle = \sum_{n,i} c_{n,i} |\phi_{n,i}\rangle \tag{128}$$

dove i è l'indice di degenerazione. Affermando che $|\psi\rangle$ è ortogonale allo stato fondamentale di $\widetilde{H}$ si intende che non ha contributi dagli stati per cui

$$(\widetilde{H}_0 + \widetilde{H}_1) |\phi_{n,i}\rangle = 0 \tag{129}$$

dunque $|\psi\rangle$ può essere composto da uno degli autostati di $\widetilde{H}_0$ con autoenergia maggiore di 0 oppure dai 12 autostati ad energia minima per $\widetilde{H}_0$, ma ortogonali allo stato fondamentale dell'Hamiltoniana totale. Una riflessione più approfondita mostra che non possono esservi contributi unicamente da questi ultimi, altrimenti la disuguaglianza sarebbe banalmente vera.
$|\psi\rangle$ deve dunque contenere qualche contributo dagli stati eccitati di $\widetilde{H}_0$ e dunque avere almeno un'energia pari a $\frac{\varepsilon\pi^2}{[2(N+1)]^2}$ come abbiamo visto nell'analisi a singolo qubit. La condizione data dall'equazione (126) limita il contributo di questi stati a

$$\sum_{n>0,i} |c_{n,i}|^2 < \frac{E_{lower}\, 4(N+1)^2}{\varepsilon\pi^2} \tag{130}$$

ciò vale nonostante la presenza di $\widetilde{H}_1$, in quanto essa è semidefinita positiva. Dunque otteniamo

$$\langle\psi|H|\psi\rangle = \sum_{i,j} c^*_{n=0,i} c_{n=0,j} \langle\phi_{n=0,i}|\widetilde{H}_0 + \widetilde{H}_1|\phi_{n=0,j}\rangle \tag{131}$$

$$+ \sum_{n>0,m>0,i,j} c^*_{n,i} c_{m,j} \langle\phi_{n,i}|\widetilde{H}_0 + \widetilde{H}_1|\phi_{m,j}\rangle \tag{132}$$

$$+ \sum_{m>0,i,j} (c^*_{m,i} c_{n=0} \langle\phi_{m,i}|\widetilde{H}_1|\phi_{n=0,j}\rangle + c^*_{n=0} c_{m,j} \langle\phi_{n=0,i}|\widetilde{H}_1|\phi_{m,j}\rangle) \tag{133}$$

$$> \frac{\varepsilon}{(N+1)^2}\left(1 - \sum_{n>0,i} |c_{n,i}|^2\right) + \frac{\varepsilon\pi^2}{4(N+1)^2} \sum_{n>0,i} |c_{n,i}|^2 \tag{134}$$

$$-2|\sum_i c_{n=0,i}||\sum_{m>0,i} c_{m,i}|\frac{\mu}{(N+1)^2} \tag{135}$$

$$> \frac{\varepsilon}{(N+1)^2} - 2\sqrt{12}|\sqrt{4(N+1)^2-16}\left(\frac{E_{lower}4(N+1)^2}{\varepsilon\pi^2}\right)^{\frac{1}{2}}\frac{\mu}{(N+1)^2} \tag{136}$$

$$= \frac{1}{(N+1)^2}\left[\varepsilon - 2\sqrt{12}\left(\frac{4(N+1)^2-16}{(N+1)^2}\right)^{\frac{1}{2}}\left(\frac{4\alpha}{\varepsilon\pi^2}\right)^{\frac{1}{2}}\mu\right] > E_{lower} \tag{137}$$

con $-\mu/(N+1)^2$ minore del valore più negativo di $\langle\phi_{m>0,i}|\widetilde{H}_1|\phi_{n=0,j}\rangle$.
Assurdo, quindi $\alpha/(N+1)^4$ è il bound inferiore al gap e di conseguenza un limite massimo al tempo di computazione.
Una simile argomentazione funziona anche per più CNOT gate, arrivando allo stesso identico risultato.

Otteniamo dunque che

$$E_{upper} \sim \frac{1}{(N+1)^2} \tag{138}$$

ed

$$E_{lower} \sim \frac{1}{(N+1)^4} \tag{139}$$

Questo non è però il risultato finale. Infatti ad ora ogni step temporale della nostra computazione ha la stessa ampiezza, all'interno dello stato fondamentale. Pertanto se si volesse solo l'esito del nostro algoritmo rappresentato dalla combinazione di tutte le N trasformazioni unitarie, l'otterrei

con una probabilità dell'ordine di $\frac{1}{(N+1)^M}$ se effettuassi una misura. Toppo bassa.

Per aumentarla, modifichiamo l'Hamiltoniana come segue:

$$h^N(U) = \varepsilon(\frac{1}{\delta}C_N^\dagger - C_{N-1}^\dagger U^\dagger)(\frac{1}{\delta}C_N - UC_{N-1}) \tag{140}$$

Ciò comporta che anziché avere $E_{upper} =\sim \frac{1}{(N+1)^2}$, sarà

$$E_{upper} = \frac{1}{(N+1)(N+\delta^2)} \tag{141}$$

ed un

$$E_{lower} = \frac{\alpha}{(N+1)^2(N+\delta^2)^2} \tag{142}$$

Se ora $\frac{1}{\delta} = \frac{1}{\sqrt{MN}}$ la probabilità che la nostra misura fallisca diminuisce vistosamente e l'upper bound diventa

$$E_{upper} = \frac{1}{(NM+N)(N+1)} \sim \frac{1}{N^2M} \tag{143}$$

con un lower bound pari

$$E_{lower} = \frac{\alpha}{(N+1)^2(NM+N)^2} \sim \frac{1}{N^4M^2} \tag{144}$$

Abbiamo raggiunto un buon livello di equivalenza fra le due tipologie di computazione quantistica. Possiamo però migliorarlo ancora: esiste un procedimento che ci permette di passare da un algoritmo ad N step temporali ed M qubits, ad uno equivalente ma con 7 step tempoorali ed M(N+1) Qubit, che si basa sulla seguente identità

$$U_1|0\rangle\frac{|0\rangle U_2\rangle + |1\rangle U_2|1\rangle}{\sqrt{2}} = \frac{1}{2}\sum_{i=0,..,3}|\Phi_i\rangle U_2\sigma_i U_1|0\rangle \tag{145}$$

con $|\Phi_i\rangle\frac{1}{\sqrt{2}}(|0\rangle\sigma_i|0\rangle + |1\rangle\sigma_i|1\rangle)$ con $\sigma_0 = I$. L'uguaglianza può essere facilmente dimostrata scrivendo esplicitamente $U_1|0\rangle = a|a\rangle + b|1\rangle$. Nel caso in cui U_2 sia l'operatore identità, quest'uguaglianza è sfruttata per permettere il trasporto di informazione quantistica, chiamata in modo ambiguo *quantum teleporting* trattato in [5] . Generalizziamo quest'affermazione ad N gate unitari, non solo due. essa diventa

$$U_1|0\rangle\frac{|0\rangle U_2\rangle + |1\rangle U_2|1\rangle}{\sqrt{2}}\frac{|0\rangle U_3\rangle + |1\rangle U_3|1\rangle}{\sqrt{2}}\cdots\frac{|0\rangle U_N\rangle + |1\rangle U_N|1\rangle}{\sqrt{2}} \tag{146}$$

$$= \frac{1}{2^{N-1}}\sum_{i_1,...,i_{N-1}}|\Phi_{i_1}\rangle\cdots|\Phi_{i_{N-1}}\rangle U_N\sigma_{i_{N-1}}U_2\sigma_{i-1}U_1|0\rangle \tag{147}$$

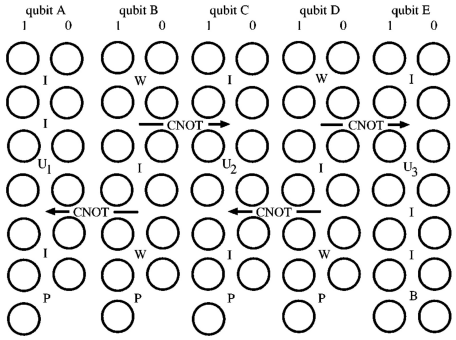

Figura 9: Ground state computer che applicano porte logiche in parallelo per produrre $U_N \cdots U_1|0\rangle$In questo caso abbiamo N=3 ed M=5. Le sette righe di qubit corrispondono alla sequenza di operazioni da svolgere.

Se $i_1 = i_2 = \cdots = i_{N-1} = 0$ otteniamo uno stato in cui i primi N qubits formano coppie EPR di stati entangled, mentre l'ultimo qubit contiene l'esito della nostra computazione. Quest'operazione dimostra che non è necessario applicare tutte le varie trasformazioni unitarie in serie per arrivare al risultato, ma possiamo ottenerlo applicandole in parallelo.
Pwer farlo, anzitutto si inizzializzano i 2N - 1 qubits nello stato $|0\rangle$. Gli si applica poi una porta logica di Walsh Hadamard:

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

su ogni qubit dispari.
Gli altri qubits sono sottoposti ad un CNOT gate così da ottenere N coppie EPR nello stato

$$|0\rangle \frac{|0\rangle\rangle + |1\rangle|1\rangle}{\sqrt{2}} \frac{|0\rangle\rangle + |1\rangle|1\rangle}{\sqrt{2}} \cdots \frac{|0\rangle\rangle + |1\rangle|1\rangle}{\sqrt{2}} \tag{148}$$

ed uno $|0\rangle$
Applichiamo poi le rispettive U_i ad ogni altro qubit in parallelo, riconducendoci così allo stato desiderato. Eseguiamo un altro CNOT gate fra bit adiacenti e si applica nuovamente la porta logica di Walsh-Hadamard, così da ottenere che lo stato $|\Phi_0\rangle$ andrà in $|0\rangle|0\rangle$, lo stato $|\Phi_1\rangle$ in $|0\rangle|1\rangle$, $|\Phi_2\rangle$ in $-i|1\rangle|1\rangle$ ed in fine $|\Phi_3\rangle$ in $|1\rangle|0\rangle$.
Se ora noi misurassimo i primi 2N qubit ed ottenessimo da tutti il valore $|0\rangle$, costringeremo l'ultimo qubit ad essere nello stato a noi cercato. Quest'occasione però è molto improbabile. Va dunque svolto un ultimo passaggio: applichiamo un'Hamiltoniana di boost che aumenti la probabilità di ottenere come stato finale dei primi 2N qubits $|0\rangle$. Sarà simile a quella vista per estrarre la misura dalla nostra ground state computation.

L'algoritmo è riportato in Figura 9.
Dopo queso accorgimento, le condizioni sul gap diventano

$$E_{upper} = \frac{1}{N^2 M} = \frac{1}{49M(2N+1)} \sim \frac{1}{NM} \tag{149}$$

ed

$$E_{lower} = \frac{1}{N^4 M^2} = \frac{1}{7^4[M(2N+1)]^2} \sim \frac{1}{(MN)^2} \tag{150}$$

trasformando gli N step temporali e gli M qubits in 7 step temporali e (2N+1)M qubits. Abbiamo dunque provato che il modello Adiabatico simula in tempo polinomiale il modello circuitale, con $T = (MN)^2$ nel peggiore dei casi.

4.4 Conclusioni

Dopo una veloce introduzione ed alcuni esempi riguardanti il modello ciruitale, si è constatato che questi è di difficile realizzazione. Abbiamo dunque analizzato un nuovo tipo di computazione che, sfruttando gli stati fondamentali di un'Hamiltoniana variabile nel tempo, risulta più resistente a fenomeni di decoerenza o di dispersione di energia (trovandosi già allo stato meno energetico). Infine abbiamo dimostrato che questi due modelli sono equivalenti. Ciò comporta che gli ottimi risultati del modello standard possano essere riprodotti dall' AQC con al più un rallentamento polinomiale, senza però risultare così sensibili ai fenomeni di disturbo dovuti all'ambiente esterno e rendendo apparentemente più vicina la realizzazione di un computer quantistico.

Riferimenti bibliografici

[1] A. Mizel, M. W. Mitchell, and M. L. Cohen, Phys. Rev. A 63, 040302 (2001).

[2] A. Mizel, M. W. Mitchell, and M. L. Cohen, Phys. Rev. A 65, 022315 (2002).

[3] A. Mizel, Phys. Rev. A 70, 012304 (2004).

[4] A. Mizel, D. A. Lidar and M. Mitchell Phys. Rev. Lett. 99, 070502 (2007)

[5] C. H. Bennett et al., Phys. Rev. Lett. 70, 1895 (1993).

[6] E. Farhi et al., *Quantum Computation by Adiabatic Evolution* MIT CTP 2936, arXiv:quant-ph/0001106.

[7] G. Schaller, S. Mostame, and R. Schutzhold, Phys. Rev. A 73, 062307 (2006).

[8] J. Preskill Lecture notes for Course on Quantum Computation, http://www.theory.caltech.edu/people/preskill/ph229/ (consultato nel mese 04/2015)

[9] J. Roland and N. J. Cerf, Phys. Rev. A 65, 042308 (2002).

[10] L. I. Schiff, *Quantum Mechanics* (McGraw-Hill, New York, 1968).

[11] M.A. Nielsen and I. L. Chuang *Quantum information and Quantum computation* (Cambridge University press, New York, 2000)