

Performance dell' algoritmo Quantum Phase Estimation (QPE)

Descrizione e problemi della QPE, presentazione dell' algoritmo Iterative Phase Estimation per applicazioni near-term

Niccolò Tiezzi

Corso 'Quantum computing and technologies'
Università di Pisa

Quantum phase estimation (QPE)

Possibili soluzioni

Iterative Phase Estimation

Conclusioni

Quantum phase estimation (QPE)

Scopo: dato U operatore unitario, $|\psi\rangle$ un suo autostato
 $U|\psi\rangle = e^{2\pi i\varphi}|\psi\rangle$, stimare φ

Requisiti: t qubits per stimare $\varphi = 0.\varphi_1 \dots \varphi_t$, m qubits per implementare $|\psi\rangle$, porta che implementi CU , la versione controllata dell' operatore

Algoritmo:

1. Inizializzazione con $|0\rangle |\psi\rangle$
2. Sovrapposizione bilanciata del primo registro

$$H|0\rangle |\psi\rangle = \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle |\psi\rangle$$

3. Applicazione dell' operatore

$$\frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle U^j |\psi\rangle = \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{2\pi i j \varphi} |j\rangle |\psi\rangle$$

4. Stima di φ tramite Quantum Fourier Transform inversa

Applicazione dell' operatore

Per ottenere i bit dell' espansione di φ nelle fasi relative dei qubits del primo registro si usano delle operazioni controllate

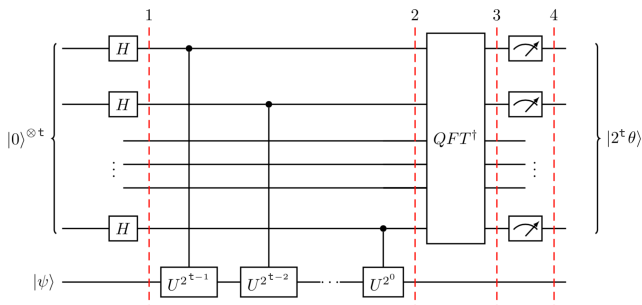


Figure 1: Circuito per QPE¹

¹<https://qiskit.org/textbook/ch-algorithms/quantum-phase-estimation.html>

Così facendo i primi t registri si trovano in $|\psi_2\rangle = \frac{1}{\sqrt{2^t}} \left(|0\rangle + e^{2\pi i 2^{t-1} \varphi} |1\rangle \right) \left(|0\rangle + e^{2\pi i 2^{t-2} \varphi} |1\rangle \right) \dots \left(|0\rangle + e^{2\pi i 2^0 \varphi} |1\rangle \right)$ i.e.

$$|\psi_2\rangle = \frac{1}{\sqrt{2^t}} \bigotimes_{k=1}^t \left(|0\rangle + e^{2\pi i \varphi 2^{t-k}} |1\rangle \right) = \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e^{2\pi i \varphi k} |k\rangle$$

che è (a meno del fattore 2^t all' esponentiale) proprio lo stato finale della Quantum Fourier Transform

$$|\varphi_1 \dots \varphi_t\rangle \xrightarrow{QFT} \frac{1}{\sqrt{2^t}} \sum_{k=0}^{2^t-1} e^{\frac{2\pi i \varphi k}{2^t}} |k\rangle$$

e quindi applicando la trasformata inversa e misurando il primo registro si trova proprio un' approssimazione a t qubits

$$\tilde{\varphi} = 0.\varphi_1 \dots \varphi_t$$

Implementazione

Usando un phase-gate come operatore $U = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2}{3}\pi i} \end{pmatrix}$ si ha

$U|1\rangle = e^{\frac{2}{3}\pi i}|1\rangle$, autostato implementabile con una porta X

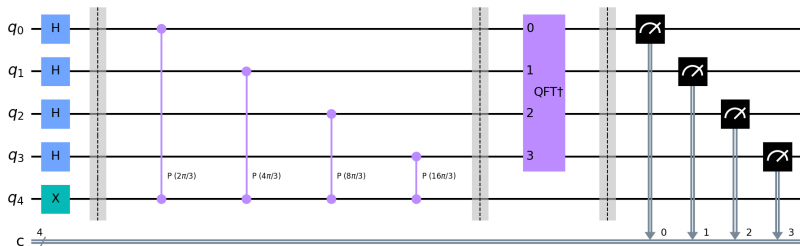


Figure 2: Implementazione in qiskit della QPE con una rotazione di $\frac{2}{3}\pi$ come operatore, i.e. $\varphi = 1/3$. Tenere a mente che in qiskit i registri sono numerati dal basso verso l'alto, al contrario rispetto alla figura 1

Esecuzione su simulatore ideale

$\varphi = 1/3 = 0.333 \dots$ non è esprimibile esattamente con t qubits, prenderemo quindi come stima la media dei 2 risultati con probabilità maggiore.

Simulando il circuito sono stati ottenuti i seguenti risultati

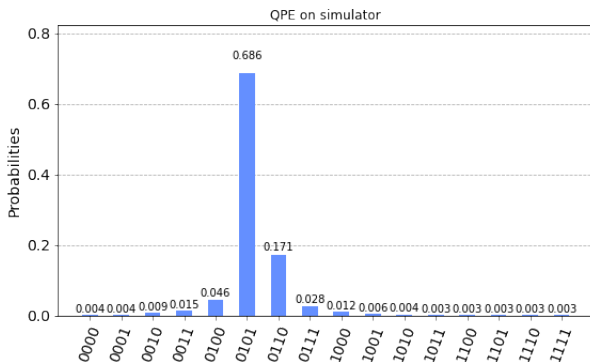


Figure 3: QPE su simulatore ideale

I due risultati migliori sono $0101 (2) = 5 (10)$ e $0110 (2) = 6 (10)$, ricordando che la QPE restituisce $2^t \varphi$ con in questo caso $t=4$ le stime sono

$$\varphi_1 = \frac{5}{2^4} = 0.3125$$

$$\varphi_2 = \frac{6}{2^4} = 0.375$$

che differiscono da $1/3$ rispettivamente di $\approx 6\%$ e $\approx 12\%$

Facendo una media pesata con le probabilità si ottiene la stima $\tilde{\varphi} = 0.324$, con una differenza di circa $\approx 3\%$, risultato migliorabile aumentando il numero dei counting qubits

Esecuzione su device reale

L' esecuzione del medesimo circuito su un device reale ha prodotto:

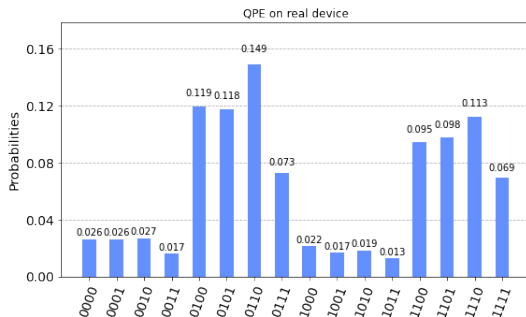


Figure 4: QPE su device 'ibmq-bogota'

I risultati presentano un discreto rumore e le probabilità sono molto più uniformi, non migliorabile aumentando il numero di counting qubits (interazioni fra i qubits fisici)

Possibili soluzioni

1. Correggere gli errori di misura tramite calibrazione del circuito
2. Sfruttare il mapping casuale durante il transpiling scegliendo un circuito con una profondità più bassa possibile per evitare rumore fra i qubits fisici
3. Cambiare algoritmo con uno meno complesso (Iterative Phase Estimation)

Soluzione 1

Prima della misura è possibile che il processo venga falsato (e.g. per un bitflip) per interazioni spurie fra i qubits

La libreria Qiskit ² consente di poter calibrare il device in uso facendoci girare dei circuiti di calibrazione che preparano il sistema nei vari stati di base computazionale e lo misurano subito dopo. Viene costruito un filtro specifico per quel device da applicare ai risultati delle misure effettuate sul nostro circuito.

²<https://qiskit.org/textbook/ch-quantum-hardware/measurement-error-mitigation.html>

Esempio: circuito a 2 qubits, base $\{|ij\rangle\}$, $i, j \in \{0, 1\}$, effettuando misure sul circuito preparato nei 4 stati di base un tipico risultato può essere:

Misurazioni				
	00	01	10	11
00	9808	95	96	1
01	107	9788	2	103
10	95	1	9814	90
11	1	107	87	9805

Table 1: Possibile esito di 10000 misurazioni, la colonna a sinistra rappresenta come sono stati preparati gli stati, idealmente sono presenti solo termini diagonali

Dividendo per il numero di misure si ottiene la matrice

$$M = \begin{pmatrix} 0.9808 & 0.0107 & 0.0095 & 0.0001 \\ 0.0095 & 0.9788 & 0.0001 & 0.0107 \\ 0.0096 & 0.0002 & 0.9814 & 0.0087 \\ 0.0001 & 0.0103 & 0.0090 & 0.9805 \end{pmatrix}$$

Sapendo lo stato del circuito prima della misura possiamo usare la matrice per mappare i conteggi ideali in dei conteggi che si otterrebbero tipicamente dato quello specifico device (essendo M specifica di un dato device su cui è stata fatta la calibrazione)

e.g. se alla fine del circuito si ha $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, l' esito ideale è

$$C_{ideal} = \begin{pmatrix} 5000 \\ 0 \\ 0 \\ 5000 \end{pmatrix}$$

Un possibile esito affetto da rumore si ottiene allora come

$$C_{noisy} = MC_{ideal}$$

$$C_{noisy} = \begin{pmatrix} 4905 \\ 101 \\ 91 \\ 4903 \end{pmatrix}$$

Quello che ci interessa è in realtà il processo opposto in cui si ha un vettore di risultati affetti da rumore e li si vuole filtrare per mitigare gli errori di misura

La correzione del rumore di qiskit applica la matrice inversa

$$C_{mitigated} = M^{-1}C_{noisy}$$

Soluzione 2

Durante il transpiling³ il circuito viene mappato per essere ottimizzato e per meglio adattarsi alla effettiva topologia del device fisico (i.e. come sono implementati i collegamenti fra i qubits)

Il circuito che meglio si adatta al device è definito dal concetto di **profondità circuitale**, che sostanzialmente indica il numero di porte consecutive sui registri, più porte implicano più interazioni. Per adattare il circuito al device fisico è necessario scomporre le porte in operazioni effettivamente implementate sul device (tipicamente solo cnot e rotazioni), eseguire swap sui registri per adattarli a quelli del device e fare tutto questo complicando il circuito il meno possibile.

³<https://qiskit.org/documentation/apidoc/transpiler.html> 

Tipicamente la scomposizione delle porte avviene in modo più o meno univoco ma gli swap dei registri devono essere fatti nel modo più ottimizzato possibile

Dedurre quale sia la miglior mappatura con gli swap è invece molto difficile (problema NP)

Per arginare il problema qiskit esegue una mappatura in modo stocastico, quindi non per forza sempre ottimale

Per migliorare la situazione si può quindi passare al transpiler il circuito varie volte e selezionare quello con la profondità minore

Analisi delle soluzioni 1 e 2

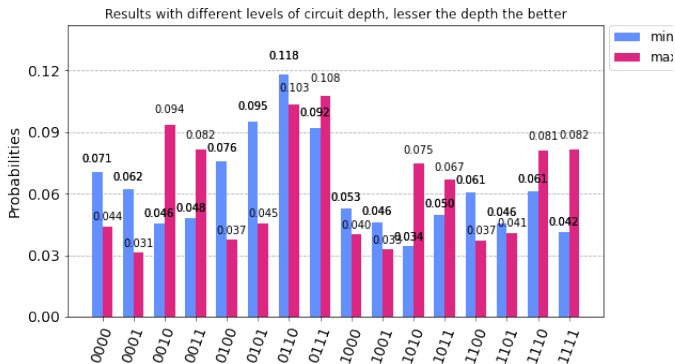
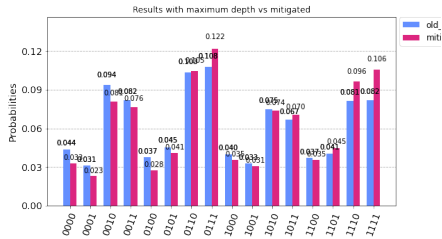
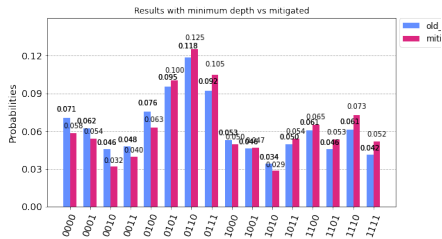


Figure 5: Profondità minima (blu) vs profondità massima (rosa)

La QPE risulta molto imprecisa anche dopo aver selezionato il circuito più ottimizzato fra 300 passati al transpiler, le probabilità maggiori (0110) e minori (1010) differiscono solamente di $\approx 8\%$



(a) Profondità massima vs mitigazione



(b) Profondità minima vs mitigazione

Figure 6: Effetti della mitigazione (rosa) degli errori di misura

Dai grafici 6a e 6b si deduce che il problema della QPE non risiede negli errori di misura, l' algoritmo continua ad essere inutilizzabile. La figura 5 mostra come l' ottimizzare il circuito per ridurre il rumore sembri la strada giusta per risolvere il problema, anche se pure in questo caso l' esecuzione su device reale non ha prodotto risultati soddisfacenti.

Iterative Phase Estimation (IPE)

Scopo: dato U operatore unitario, $|\psi\rangle$ un suo autostato
 $U|\psi\rangle = e^{2\pi i\varphi}|\psi\rangle$, stimare φ

Requisiti: un singolo counting qubit, n qubits per implementare $|\psi\rangle$, porta che implementi CU , un registro a m bit classici $c_0 \dots c_{m-1}$ per salvare $\varphi = 0.1 \dots \varphi_m$ e la possibilità di implementare condizionali classici e re inizializzazioni dei qubit ⁴

In un algoritmo quantistico fare operazioni condizionali classiche significa implementare un controllo sul registro classico anzichè fra registri quantistici

⁴[https:](https://qiskit.org/documentation/tutorials/textbook/01_IQPE.html)

Algoritmo:

1. Preparazione in $|+\rangle |\psi\rangle$
2. Applicando 2^{m-1} volte U al secondo registro, la fase relativa del primo qubit è $(\varphi = 0.\varphi_1 \dots \varphi_m)$

$$\frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i 2^{m-1} 0.\varphi_1 \dots \varphi_m} |1\rangle \right) = \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i 0.\varphi_m} |1\rangle \right)$$

che se misurato *in base* X produce 0 o 1 a seconda che $\varphi_m = 0, 1$ e si salva il risultato nel registro classico c_0


3. Si *reinizializza* il primo qubit a $|+\rangle$ e si applica U 2^{m-2} volte, la fase relativa adesso sarà $e^{2\pi i 0.\varphi_{m-1}\varphi_m} = e^{2\pi i 0.\varphi_{m-1}} e^{2\pi i \varphi_m/4}$
4. Per estrarre φ_{m-1} è necessario correggere la fase relativa applicando una rotazione di $-\pi\varphi_m/2$ controllata da c_0 , quindi un condizionale classico
5. Il passo k -esimo consiste nell' inizializzazione, applicazione di U 2^{m-k} volte e correzione di angolo $-2\pi 0.0\varphi_{m-k+2} \dots \varphi_m$

e.g. per $m = 4$ al passo 3 la correzione dovrà essere di angolo

$$-2\pi i 0.0\varphi_3\varphi_4 = -2\pi i \left(\frac{\varphi_3}{4} + \frac{\varphi_4}{8} \right)$$

A seconda che $\varphi_3\varphi_4 = 01, 10, 11$ si avrà una rotazione di $-\frac{\pi}{4}, -\frac{\pi}{2}, -\frac{3\pi}{4}$, che sarà controllata dai registri classici che saranno rispettivamente in $0001 (2) = 1 (10), 0010 (2) = 2 (10), 0011 (2) = 3 (10)$

Il controllo sul registro classico viene fatto leggendo il valore *decimale* (e non binario) in cui il registro si trova ⁵

⁵https://qiskit.org/documentation/tutorials/circuits/3_summary_of_quantum_operations.html#Non-unitary-operations 

Implementazione

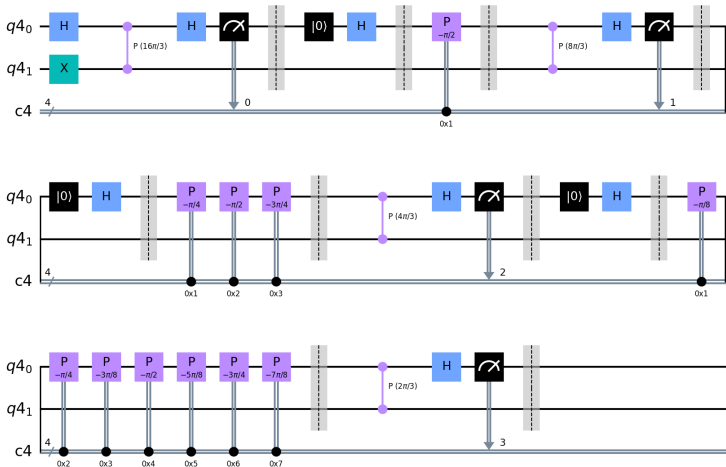


Figure 7: Circuito per IPE implementato in qiskit con $U = R_z(2/3\pi)$ e $m=4$

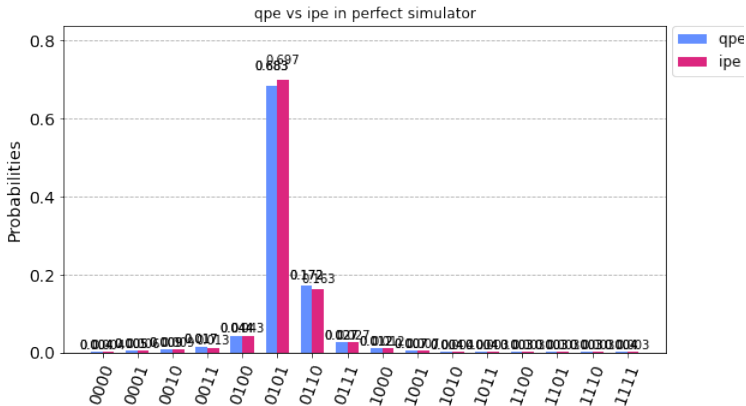


Figure 8: Confronto IPE (rosso) e QPE (blu) su simulatore ideale con phase-gate di $2\pi/3$

I risultati su simulatore sono pressochè identici con la Quantum Phase Estimation tradizionale, la IPE è quindi un valido sostituto della QPE

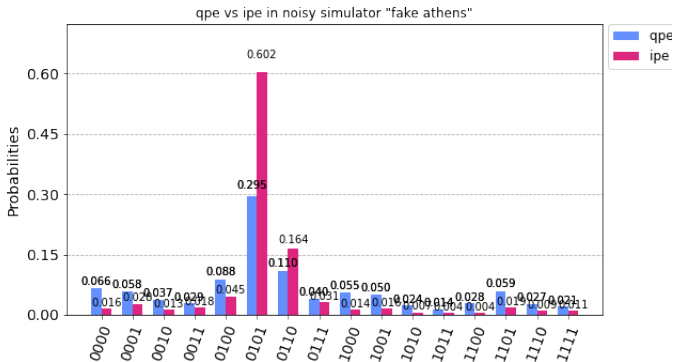


Figure 9: Confronto IPE (rosso) e QPE (blu) su simulatore con modello di rumore realistico applicato

Il grafico mostra i risultati di una QPE con entrambi gli accorgimenti visti in precedenza (ottimizzazione durante il transpiling e mitigazione degli errori di misura) confrontati con l' algoritmo IPE usato senza correzioni

I migliori 2 risultati (0101,0110) sono chiaramente distinguibili dal resto (il terzo con la probabilità più alta, 0100, differisce dal secondo di $\approx 70\%$)
Abbiamo quindi le stime

$$\varphi_1 = \frac{5}{2^4} = 0.3125$$

$$\varphi_2 = \frac{6}{2^4} = 0.375$$

che differiscono da $1/3$ rispettivamente di $\approx 6\%$ e $\approx 12\%$, risultati analoghi a quelli ottenuti col simulatore privo di rumore

Conclusioni

- ▶ L' algoritmo IPE è una valida alternativa alla phase estimation tradizionale
- ▶ Minor complessità (profondità) del circuito, i.e. circuito meno affetto da rumore
- ▶ Non è necessario aumentare il numero di qubits per aumentare la precisione
- ▶ Ad oggi (Marzo 2022) i condizionali classici non sono stati ancora implementati sui device fisici IBM anche se sono in lavorazione ⁶

⁶<https://research.ibm.com/blog/quantum-development-roadmap>

Bibliografia

- ▶ M.A. Nielsen, I.L. Chuang - Quantum computation and quantum information
- ▶ IBM Qiskit textbook <https://qiskit.org/textbook-beta>