POLITECNICO DI TORINO
Repository ISTITUZIONALE

Quantum Computing tutorial

Original
Quantum Computing tutorial / Cirillo, G.A., Giusto, E., Gandino, F., Mondo, G.. - In: NOTIZIARIO TECNICO TELECOM ITALIA. - ISSN 2038-1921. - ELETTRONICO. - n.2 - 2020:(2020), pp. 58-77.

Availability:
This version is available at: 11583/2839257 since: 2020-07-31T12:38:58Z

Publisher:
TIM S.p.A.

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
GENERICO -- per es. Nature : semplice rinvio dal preprint/submitted, o postprint/AAM [ex default]

(Article begins on next page)

13 October 2026

QUANTUM COMPUTING TUTORIAL

Giovanni Amedeo Cirillo, Filippo Gandino,
Edoardo Giusto, Giovanni Mondo

Il Quantum Computing (QC) è rimasto a lungo un'idea nell'immaginario della comunità scientifica, ma grazie agli enormi progressi degli ultimi decenni sta acquistando una credibilità crescente al punto da ritenere realistica la sua applicazione su larga scala su un orizzonte temporale relativamente vicino.

I computer quantistici implementano una nuova modalità di processare le informazioni e, se la tecnologia riuscirà a rendere disponibile la capacità di calcolo che promette, potranno essere utilizzati per analizzare problemi non trattabili dai computer classici, aprendo nuove opportunità in termini di scoperte, innovazione e applicazione con impatti che potrebbero essere rivoluzionari in tutti i settori.

Anche se la tecnologia non ha raggiunto ancora la piena maturità, è già conveniente utilizzare il QC eventualmente con opportuni adattamenti, capendo le modalità e le logiche della programmazione quantistica, beneficiando dei vantaggi e delle opportunità di sviluppo di nuovi use cases e di apertura di nuovi scenari.

L'articolo si propone di fornire una panoramica sulle applicazioni, le tipologie di quantum computer, gli ambienti di sviluppo e la modellizzazione algoritmica per mostrare come il QC possa a tutti gli effetti essere preso in considerazione per sviluppare use case reali.

In fondo il quantum non è poi così "spooky" come potrebbe sembrare...

Applicazioni del Quantum Computing

Il QC è considerato la prossima futura grande rivoluzione dell'Information Technology e promette di avere grossi impatti in tutti i settori grazie alle enormi potenzialità di calcolo che renderà disponibili. I quantum computer si distinguono dai computer classici perché implementano una diversa modalità di elaborazione dei dati e sono pertanto adatti per processare classi di problemi differenti non risolvibili in tempi ragionevoli dai computer tradizionali.

La motivazione iniziale alla base dello sviluppo dei computer quantistici nasce dall'idea di superare i limiti mostrati dai computer classici, quando la ricerca avanzata nel campo della fisica e della chimica si è spinta a livello di particelle subatomiche. Su queste dimensioni intervengono i principi della meccanica quantistica e quindi, per progredire ulteriormente, si è pensato che fosse necessario disporre di nuovi computer che, funzionando secondo le stesse leggi, fossero in grado di modellare e simulare questi fenomeni in maniera più accurata.

Successivamente ci si è resi conto che un computer quantistico potesse essere utilizzato non solamente come simulatore, ma per trattare in maniera efficiente una più ampia gamma di problemi complessi, grazie alla sua modalità di elaborazione probabilistica intrinseca nei principi della meccanica quantistica [1],[2].
Essendo una tecnologia in gran parte ancora in uno stadio di ricerca, l'utilizzo del QC si diffonderà nel tempo, coerentemente con la crescita della sua capacità computazionale che, come verrà descritto nella sezione sulle Tipologie di Quantum Computer, si misura in qubit.
Il progressivo aumento della capacità di calcolo ne abiliterà, corrispondentemente, l'applicazione a casi d'uso di complessità crescente.

Partendo dall'idea originaria, l'impiego del QC ai settori della chimica e alla scienza dei materiali richiede una potenza di calcolo di almeno un centinaio di qubit. Considerando che nel 2017 il quantum computer di IBM gestiva 16 qubit e oggi 50, si può pensare che un centinaio di qubit non saranno disponibili prima del 2025.

L'applicazione del quantum a questi settori potrebbe avere conseguenze importanti come lo sviluppo di batterie per le automobili elettriche con maggior autonomia, nuovi processi industriali (il processo Haber - Bosch, per la sintesi industriale dell'ammoniaca, responsabile del consumo dell'1-2% dell'energia a livello globale e del 2-3% della produzione di CO_2, è impiegato da oltre un secolo per la produzione di fertilizzanti e prodotti per la pulizia perché finora non si è riusciti a sviluppare un processo alternativo meno dispendioso), così come nuovi materiali per realizzare Quantum Computer con prestazioni superiori alle attuali.

Un'altra area di possibile utilizzo è rappresentata dai problemi di ottimizzazione. Con metodi di ricerca di un minimo funzionale (annealing) si risolvono già da 40 anni problemi di ottimizzazione combinatoria, sfruttando il fatto che, il principio fisico della ricerca di un equilibrio molecolare corrisponde alla minimizzazione di una funzione.

Questo ha portato l'azienda canadese D-Wave, a partire dal 2007, allo sviluppo di un Quantum Annealer di migliaia di qubit (ma utilizzati in maniera diversa rispetto a IBM) e ispirato Fujitsu nello sviluppo di annealer basati su tecnologia CMOS tradizionale.

Un esempio di ottimizzazione combinatoria in cui questi metodi sono chiamati ad operare è quello della determinazione, nel minor tempo possibile, del miglior portafoglio di investimenti in un ventaglio di titoli in continua evoluzione.

Un settore interessato al QC è quindi quello della finanza, ma anche, generalizzando il concetto di ottimizzazione, la logistica per sfruttare al meglio le risorse, individuare i percorsi migliori per il trasporto, gestire l'approvvigionamento delle materie prime e dei prodotti. Sempre nel settore finanziario, l'interesse riguarda anche l'analisi dei rischi e le previsioni di mercato, per sfruttare l'applicabilità del QC a problemi complessi in cui intervengono molte variabili che sono caratterizzati da un certo grado di incertezza.
Promettenti sono anche le aspettative del QC per accelerare maggiormente l'evoluzione dell'Artificial Intelligence/Machine Learning (AI/ML), grazie alla capacità di processare grosse moli di dati, funzionale alla classificazione delle informazioni o per individuare pattern o aree di ottimizzazione (minimo o massimo).
Ne beneficerebbero di conseguenza quei settori come la medicina, che già utilizzano l'AI/ML, per migliorare le capacità di analisi

Attività \ Pianificazione	Short Term (2020 ÷2025)	Medium Term (> 2025)	Long Term (>2030)
Simulazione	**Manifattura** (Analisi molecolare)	**Chimica** (Ingegneria molecolare, Nuovi Processi industriali) **Manifattura** (Nuovi materiali) **Agricoltura** (Nuovi fertilizzanti) **Medicina** (Nuove medicine, nuove proteine) **Logistica** (Simulazione scenari) **Finanza** (Previsioni sui derivati, Analisi di rischio)	**Medicina** (Previsioni diffusione malattie)
Ottimizzazione	**Logistica** (Routing trasporto, ottimizzazione rete di distribuzione) **Telecomunicazioni** (Pianificazione, Gestione risorse non e near real-time) **Medicina** (Finanziamento nuove medicine, Analisi struttura proteine)	**Finanza** (ottimizzazione portfolio, gestione transazioni) **Logistica** (Gestione supply chain) **Medicina** (Gestione supply chain) **Manifattura** (Ottimizzazione processi produttivi, Gestione Supply chain, Programmazione produzione)	**Telecomunicazioni** (Gestione risorse real-time)
Artificial Intelligence/ Machine Learning	**Finanza** (stime patrimoniali)	**Logistica** (Gestione imprevisti, previsioni di approvvigionamento) **Finanza** (Gestione frodi) **Telecomunicazioni** (Analisi di mercato) **Medicina** (Diagnosi mediche, Analisi genetiche, Sperimentazioni di farmaci) **Manifattura** (Controllo qualità)	**Finanza** (Raccomandazioni di acquisto) **Manifattura** (Progettazione strutturale, Fluido dinamica) **Telecomunicazioni** (Automazione (SON-like) real-time su larga scala)
Calcolo		**Crittografia** (Nuovi protocolli di sicurezza quantum-proof per comunicazioni e dati)	**Crittografia** (Cracking RSA, nuovi sistemi di protezione quantum-proof per comunicazioni e dati)

Tabella 1
Previsione di applicazione del QC a casi d'uso specifici di vari settori

e diagnosi, ma anche altri settori, come le Telecomunicazioni, per attività analoghe ma su dati di altra natura [3].
Infine, nell'ampia gamma dei servizi legati alla sicurezza delle comunicazioni e alla protezione dei dati, il QC ha già avuto e continuerà ad avere grossi impatti. Si può dire che tutta l'attività di ricerca e sviluppo e gli enormi investimenti sulle tecnologie quantistiche sono stati innescati dal rischio legato alla capacità di un computer quantistico, con sufficiente potenza di calcolo (migliaia di qubit per un quantum computer tipo quello di IBM), di poter crackare facilmente gli attuali sistemi di cifratura (RSA) utilizzando un algoritmo creato da Peter Shor nel 1994.

Molti sistemi di cifratura a protezione delle comunicazioni, transazioni e dati sensibili nei settori finanziario, sanitario, militare... si basano sulla difficoltà matematica di scomporre un numero grande nei suoi fattori primi (fattorizzazione).

Il QC riesce ad analizzare facilmente i dati in frequenza (FFT) e questo permette di risolvere il problema della fattorizzazione in tempi esponenzialmente più veloci se confrontati anche con quelli del più potente, ad oggi, supercomputer classico.

Si presume che computer quantistici tipo quello di IBM, in grado di violare gli attuali sistemi di sicurezza, saranno disponibili non prima di 10 anni. Di fronte a questo rischio sono in corso molte iniziative volte a favorire lo sviluppo di nuovi algoritmi e sistemi di protezione delle comunicazioni e delle informazioni.

Sulla base del trend di sviluppo del QC la tabella seguente mostra gli ambiti di applicazione del QC e una previsione dei tempi in cui possono esserci ricadute pratiche.

Si tratta ovviamente di stime che derivano dallo stato dell'arte della tecnologia e che potrebbero subire rivisitazioni a seguito di accelerazioni o rallentamenti che potrebbero verificarsi nell'ambito dell'attività di ricerca e sviluppo [4].

Tipologie di Quantum Computer

Il computer quantistico è stato teorizzato negli anni Ottanta del secolo scorso, quando Richard Feynman, Jurij Manin e David Deutsch arrivarono autonomamente alla conclusione che un sistema quantistico avrebbe consentito di superare gli intrinseci limiti dei computer classici - in termini di accuratezza e tempistiche di esecuzione - nell'ambito della simulazione di sistemi fisici quantistici quali atomi, molecole o materiali. Deutsch per primo introdusse il concetto di Universal Quantum Computer [5], una macchina di Turing quantistica capace di simulare qualsiasi sistema fisico finito e realizzabile con sistemi idealmente isolati (temperatura richiesta pari a 0 K) con accuratezza arbitrariamente elevata.

Il modello di Deutsch si riferisce chiaramente ad un dispositivo di difficile realizzazione, in quanto richiederebbe un hardware assolutamente fault-tolerant ed interamente quantistico [6], ovvero costituito non solo da un'unità di esecuzione quantistica ma anche da una memoria quantistica, al momento non disponibile.

Tuttavia nel mondo della computer science si è mantenuto l'interesse per l'idea originaria di integrare i principi della meccanica quantistica nella logica di calcolo. Questo ha portato allo sviluppo di varie tipologie di computer quantistici, il cui principio di funzionamento comune consiste nell'applicazione di eccitazioni esterne al sistema quantistico codificante l'informazione associata al problema da risolvere, con l'intento di provocarne un'evoluzione temporale che consenta di raggiungere uno stato finale corrispondente alla soluzione del problema.

In particolare, i principi della fisica quantistica maggiormente adoperati per cercare di ottenere un vantaggio quantistico, ovvero una risoluzione computazionalmente più efficiente della migliore corrispettiva classica, sono la sovrapposizione e l'entanglement
Secondo il principio di sovrapposizione uno stato quantistico può essere rappresentato dalla combinazione di due o più stati quantistici.

Questo implica che l'unità di informazione quantistica, il qubit (quantum bit), può trovarsi nella sovrapposizione di due stati codificanti 0 e 1, ovvero può avere nello stesso tempo probabilità non nulle di valere tanto 0 quanto 1.

Questa proprietà risulta estremamente vantaggiosa nell'ambito della computazione in quanto la stessa operazione può essere simultaneamente valutata su più campioni "in sovrapposizione" di un dataset, ciascuno dei quali associato ad uno stato quantistico. La sovrapposizione è inoltre strettamente legata al fenomeno dell'interferenza, che nella computazione quantistica è sfruttata come un meccanismo di incremento della probabilità della soluzione del problema.

L'entanglement è una proprietà dei sistemi quantistici costituiti da più sottosistemi che possono essere soggetti ad un'intrinseca correlazione che rende impossibile la loro analisi individuale. Nell'ambito dell'informazione quantistica la correlazione intrinseca tra i qubit implica che se si esegue una misurazione/lettura su uno di loro, il risultato influenza istantaneamente i corrispettivi valori degli altri. Dal momento che il principio dell'entanglement vale anche se i qubit sono spazialmente distanti, questo trova spazio non solo nell'ambito della computazione quantistica ma anche in quello delle comunicazioni quantistiche.

I principi di sovrapposizione ed entanglement sono gli elementi che conferiscono ai computer quantistici quelle potenzialità che li rendono superiori rispetto ai computer classici nel trattare certe famiglie di problemi complessi.

Sono tuttavia stati instabili, che tendono ad esaurirsi a causa della decoerenza (disturbi dell'ambiente circostante) i cui effetti, aumentando nel tempo, determinano un progressivo incremento della probabilità di errore fino alla perdita totale delle proprietà quantistiche e quindi della possibilità di sfruttare le capacità di calcolo dei computer quantistici.

Quantum Gate Array

Il computer quantistico basato su modello Quantum Gate Array (QGA) è caratterizzato dall'esecuzione di operazioni sotto forma di porte quantistiche - una sorta di estensione al qubit della progettazione logica dell'elettronica digitale classica - con le quali è possibile approssimare tutte le possibili evoluzioni unitarie di un sistema quantistico (per approfondimenti sulle porte quantistiche si rimanda a [7]).

Questo modello assume dal punto di vista hardware che l'unità di esecuzione costituisca l'unica sezione prettamente quantistica del calcolatore.
A differenza delle porte logiche classiche, che possono essere progettate con un opportuno circuito a transistor, le porte quantistiche sono implementate da campi elettromagnetici oscillanti ad una frequenza di risonanza caratteristica di ciascun qubit costituente l'hardware e il cui valore assoluto dipende dalla tecnologia di fabbricazione (per esempio per i qubit superconduttivi nella banda delle microonde, per gli ioni intrappolati nella stessa banda o addirittura in banda ottica).

In Figura 1a è riportato un generico schema a blocchi – ciascuno dei quali è costituito da porte quantistiche, come riporta il dettaglio del modulo Valutazione – di un programma quantistico basato su modello QGA, anche detto circuito quantistico per analogia con i circuiti digitali costituiti da porte logiche.
Sinteticamente il flusso processivo prevede:

FORMALISMO DEL QUANTUM GATE ARRAY

Nel modello Quantum Gate Array lo stato quantistico di un sistema a N qubit è descritto da un vettore di stato

$|\psi> = [c_0\ c_1 \dots c_{(2^N)-1}]^T = \Sigma_i c_i|i>$

dove c_i è un numero complesso chiamato ampiezza di probabilità, il cui modulo quadro $|c_i|^2$ è pari alla probabilità che il sistema si trovi nell'autostato corrispondente al vettore $|i>$ (pertanto $\Sigma_i|c_i|^2 = 1$). Nel caso di un singolo qubit gli autostati sono 2 ($|0> = [1\ 0]^T$ e $|1> = [0\ 1]^T$), mentre per N qubit sono 2^N ($|00\dots0> = [1\ 0 \dots 0]^T \div |11\dots1> = [0 \dots 0\ 1]^T$). La sfera di Bloch, che costituisce la rappresentazione geometrica di un qubit $[c_0\ c_1]T$, è osservabile in Figura Aa. Tutti i vettori delimitati dall'origine degli assi cartesiani e da un punto sulla superficie della sfera sono associabili ad uno stato di un qubit; in particolare, i due vettori paralleli all'asse z sono associati agli autostati $|0>$ e $|1>$, mentre i rimanenti descrivono uno stato con sovrapposizione. I vettori che giacciono sullo stesso parallelo differiscono per fase ϕ e sono accomunati dall'angolo ϑ e soprattutto dalle probabilità $|c_0|^2$ e $|c_1|^2$.

Le porte quantistiche corrispondono a rotazioni del vettore di stato attraverso matrici complesse unitarie U

$U\ |\psi> = U\ \Sigma_i c_i|i> = \Sigma_i c_i U|i>$

L'espressione precedente mette in evidenza la linearità del modello, per cui è possibile valutare l'effetto di una porta su ciascun autostato individualmente ($U|i>$). Questo approccio è matematicamente più agevole del prodotto matrice-vettore e consente di progettare circuiti quantistici in maniera più intuitiva. Alcune porte quantistiche notevoli sono riportate in Figura Ab (per approfondimenti si rimanda a [15]). La reversibilità è una peculiarità delle evoluzioni unitarie, per cui per ciascuna porta quantistica descritta da matrice U è possibile definire una porta duale con matrice U' che ne annulla l'effetto, ovvero $U' = U^{-1}$, dove U^{-1} è la matrice inversa di U.

Le porte a sinistra coinvolgono un solo qubit e sono riconducibili a rotazioni degli assi cartesiani di un angolo ϑ: la porta in alto si chiama X ($R_x(\pi)$) ed è la corrispettiva quantistica della NOT classica (sostanzialmente scambia le ampiezze di probabilità di $|0>$ e $|1>$), la porta Z ($R_z(\pi)$) cambia il segno dell'ampiezza di probabilità di $|1>$, la porta $R_z(\vartheta)$ è associata ad una generica rotazione di un angolo ϑ dell'asse z, infine la porta H di Hadamard ($R_x(\pi)$ seguita da $R_y(-\pi/2)$) consente di generare una sovrapposizione uniforme di stati quando applicata ad un autostato.

Le porte a destra coinvolgono due o più qubit e sono operazioni di tipo controllato, per cui applicano un'evoluzione unitaria U diversa dall'identità ad uno o più qubit target in funzione del valore di uno o più qubit di controllo: la porta CNOT in alto inverte un qubit target se un altro di controllo vale 1, mentre la porta CCNOT o di Toffoli in basso inverte un qubit target se i due qubit di controllo valgono entrambi 1. In altre parole, le porte CNOT e CCNOT modificano il qubit target secondo l'operazione booleana XOR coinvolgente il qubit target stesso e il valore controllante (il valore del qubit di controllo nel caso della CNOT, il risultato dell'operazione AND coinvolgente i due qubit di controllo nel caso della CCNOT), pertanto sono adoperabili anche con bit classici e sono dette di tipo booleano-reversibile (la reversibilità può essere dimostrata applicando due CNOT o due CCNOT consecutivamente sugli stessi qubit). L'aspetto prettamente quantistico di queste porte è chiaramente legato al fatto che sono applicabili ad uno stato quantistico in sovrapposizione, le cui coppie di ampiezze di probabilità c_{10}-c_{11} e c_{110}-c_{111} risultano scambiate in seguito all'applicazione della CNOT e della CCNOT rispettivamente.

È possibile dimostrare che H, $R_z(\pi/2)$, $R_z(\pi/4)$ e CNOT costituiscono un set universale di porte quantistiche, con le quali è possibile approssimare qualsiasi altra porta quantistica.

Le porte quantistiche introdotte in precedenza sono adoperabili negli algoritmi di Grover e variazionali. Nel primo caso la soluzione è etichettata da un circuito che cambia il segno dell'ampiezza di probabilità della soluzione e che è implementato con porte quantistiche di tipo booleano-reversibile, per asserire un qubit ausiliario di flag secondo un meccanismo analogo a quello della progettazione delle reti logiche classiche basate su tavole di verità, con la porta esclusivamente quantistica Z, per codificare il valore del qubit di flag sul segno dell'ampiezza di probabilità dello stato. Per quanto concerne i circuiti di tipo variazionale la topologia delle porte è fissata mentre gli angoli di rotazione ϑ sono aggiornati dal minimizzatore classico ad ogni iterazione.

La notazione vettoriale può essere anche adoperata per definire algebricamente l'entanglement. In un sistema a N-qubit privo di entanglement il vettore di stato può essere scomposto in N vettori, ciascuno associato ad un qubit. Per esempio, il vettore di stato coinvolgente due qubit

$|\psi> = 1/\sqrt{2}*[1\ 1\ 0\ 0]^T = 1/\sqrt{2}|00> + 1/\sqrt{2}|01>$

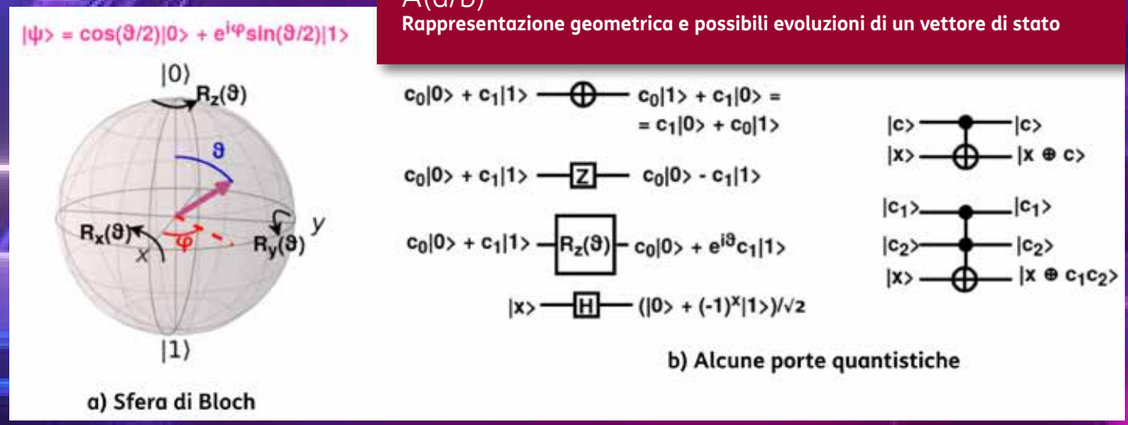
può essere scomposto in due vettori $|0>$ e $(|0> + |1>)/\sqrt{2}$. Se al contrario la scomposizione non è algebricamente possibile, allora il vettore descrive uno stato entangled. L'esempio più noto di stato entangled è il cosiddetto $|\Phi^+>$ di Bell per due qubit, il cui vettore di stato

$|\psi> = 1/\sqrt{2}*[1\ 0\ 0\ 1]^T = 1/\sqrt{2}|00> + 1/\sqrt{2}|11>$

non può essere scomposto in due vettori disgiunti. In tal caso se si misura $|0>$ sul qubit di sinistra si è certi che il qubit di destra varrà $|0>$, viceversa se si misura $|1>$ sul qubit di sinistra quello di destra varrà sicuramente $|1>$.

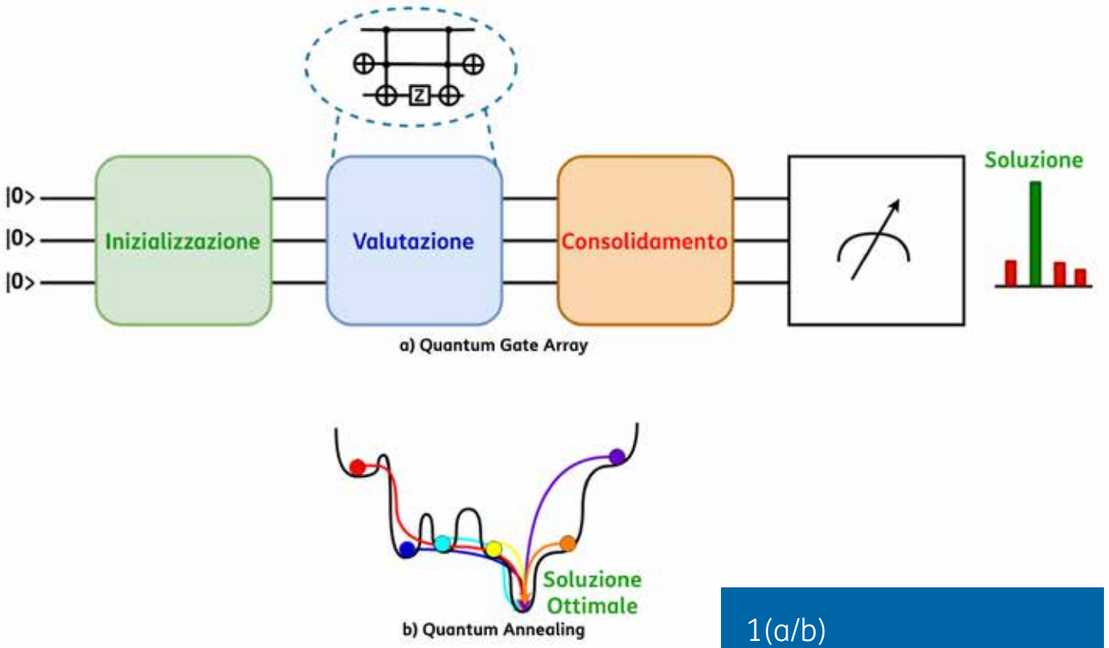
giovanni_cirillo@polito.it, edoardo.giusto@polito.it

A(a/b)
Rappresentazione geometrica e possibili evoluzioni di un vettore di stato



a) Sfera di Bloch

b) Alcune porte quantistiche



1(a/b)
Meccanismi concettuali di funzionamento di un Quantum Gate Array e di un Quantum Annealer

1. predisposizione del sistema allo stato ground |00...0>;
2. configurazione del sistema in uno stato corrispondente alla mappatura dei dati del problema (inizializzazione);
3. sfruttando il meccanismo della sovrapposizione, elaborazione simultanea su tutti gli autostati associati ai dati al fine di individuare od etichettare le soluzioni del problema (valutazione);
4. applicazione di ulteriori porte quantistiche per variare lo stato del sistema, così che la soluzione del problema presenti una probabilità di misura significativamente maggiore degli altri risultati (consolidamento).

Il modello QGA si è affermato negli anni Novanta del secolo scorso con l'individuazione dei primi problemi caratterizzati da un vantaggio quantistico.

I casi più emblematici sono rappresentati dal problema di fattorizzazione dei numeri interi, risolvibile con l'algoritmo di Peter Shor (1994) [8] con complessità polinomiale anziché esponenziale come nel caso classico e la ricerca in un insieme non ordinato, risolvibile con l'algoritmo di Grover (1996) [9] con complessità proporzionale alla radice quadrata del numero degli elementi costituenti il dataset, inferiore rispetto alla complessità lineare dell'elaborazione classica.

Pur essendo ancora basso il grado di maturità del QGA e distante dal modello computazionale del computer quantistico universale di Deutsch, questa tecnologia è la più investigata per l'analogia di funzionamento con i computer classici e quindi la versatilità di utilizzo con algoritmi in grado di risolvere problemi strettamente applicativi (per esempio di ottimizzazione o di simulazione di molecole) con evidente vantaggio quantistico.

L'hardware QGA può essere quantistico a tutti gli effetti, per esempio superconduttivo-o a ioni intrappolati o spin molecolari, oppure può essere costituito da simulatori classici in cui ci si limita a riprodurre il funzionamento di un calcolatore quantistico, calcolando la distribuzione di probabilità a fine esecuzione di un circuito e tenendo eventualmente conto delle non-idealità dell'hardware quantistico sotto forma di modelli di rumore semplificati.

Le soluzioni basate su tecnologia classica sono interessanti anche per il loro possibile impiego on-premises in particolare a livello di edge computing nel 5G per gestire applicazioni time-critical, dal momento che non necessitano di essere installate in ambienti isolati e mantenute a temperature prossime allo 0 assoluto.

Per un approfondimento sul Quantum Gate Array si rimanda al box "Formalismo del Quantum Gate Array" e per un esempio di algoritmo quantistico al box "Un esempio di algoritmo per Quantum Gate Array - Algoritmo di Grover".

Quantum Annealer vero e inspired

Il simulated annealing è un algoritmo che si è diffuso a partire dagli anni Ottanta per risolvere problemi di ottimizzazione, per individuare il minimo globale di una funzione di costo che presenta più minimi locali.

La sua variante quantistica si chiama quantum annealing [10] ed il suo corrispettivo calcolatore è chiamato Quantum Annealer (QA).
La funzione di costo di un problema risolvibile con QA viene mappata sul profilo dell'energia - quindi nell'orientazione - di un insieme di spin/qubit che interagiscono lungo un asse con altri spin e/o con un campo magnetico esterno.

Il profilo di potenziale del sistema (vedasi Figura 1b) presenta dei minimi locali ed uno globale, il cui autostato (minimo assoluto) corrisponde alla soluzione ottimale del problema.

È possibile constatare che il profilo dell'energia di un insieme di spin/qubit è assimilabile alla funzione di costo di un problema classico di ottimizzazione combinatoria Quadratic Unconstrained Binary Optimization (QUBO) [11], una categoria di problemi risolvibili classicamente con algoritmi di complessità computazionale non-polinomiale.

Elaborare un algoritmo QUBO su un QA, si traduce nell'applicazione di un campo nel piano trasverso a quello di interazione dei qubit che, come osservabile nella Figura 1b, genera una sovrapposizione di stati - in questo contesto corrispondente alla simultanea presenza del sistema in tutte le buche di potenziale del profilo energetico – e facilita il raggiungimento del minimo assoluto della funzione di costo attraverso l'effetto tunnel, secondo cui un sistema quantistico può attraversare una barriera arbitrariamente alta di energia potenziale; questa modalità di funzionamento determina il vantaggio quantistico.
Analogamente alla famiglia QGA, anche per il QA l'hardware può essere effettivamente quantistico, vedasi i chip superconduttivi di D-Wave Systems, oppure classico quantum-inspired, ovvero costituito da emulatori come quello fabbricato da Fujitsu.

Il termine quantum-inspired trae origine dal fatto che l'emulatore classico cerca di imitare il funzionamento a run-time di un ambiente quantistico (solitamente ideale) attraverso delle routine ottimizzate per la riproduzione delle evoluzioni unitarie quantistiche.
Un approccio quantum-inspired, pur mostrando dei limiti legati all'overhead di risorse classiche richieste, risulta attualmente vantaggioso in termini di numero di qubit utilizzabili (ottomila emulati

UN ESEMPIO DI ALGORITMO PER QUANTUM GATE ARRAY

Algoritmo di Grover

Pubblicato nel 1996 , fornisce un approccio nuovo al problema della ricerca di un elemento in una lista non ordinata, di tipo NP (nondeterministic polynomial time), che non è risolvibile classicamente se non con una ricerca esaustiva, che consiste banalmente nel leggere uno dopo l'altro tutti gli elementi fino a trovare quello desiderato.

Un'analogia visiva del problema può essere questa: consideriamo una fila di N cassetti, dei quali uno soltanto contiene una pallina. Per sapere dov'è, con la ricerca esaustiva dovremo aprire uno dopo l'altro al più N-1 cassetti. Ripetendo molte volte l'esperimento avremo un valore atteso per il numero di tentativi pari a N/2.

L' algoritmo di Grover ci permette invece di trovare la soluzione in un numero di passaggi pari a √N, decisamente inferiore al crescere di N.

Per restare nell' analogia della pallina nei cassetti, è come se, invece di aprire cassetti, potessimo assestare dei colpetti alla cassettiera, per individuare dall'eco dove si trova quello pieno.

Naturalmente si tratta di un'analogia che però, seppur grossolanamente, sottolinea quanto l'incremento di prestazioni dato dal Quantum Computing rispetto a quello classico non consista in un semplice aumento di velocità o di parallelismo, ma nella possibilità di mettere in atto strategie di calcolo precedentemente impossibili da realizzare.

Descrizione del funzionamento

Codifichiamo l'insieme delle N possibili soluzioni in un registro di n qubit, dove $n = \log_2(N)$.

Il primo step consiste nel preparare i qubit del registro in una sovrapposizione di stati equiprobabile, usando l'operatore $H^{\otimes n}$, chiamato porta universale di Hadamard di ordine n.

Successivamente vengono applicati al registro, in sequenza, l'operatore Oracolo (Uω)e l'operatore diffusione di Grover ($H^{\otimes n}.(2|0^n\rangle\langle 0^n| - I_n).H^{\otimes n}$).

Questa operazione altera lo stato del registro, amplificando la probabilità che in fase di lettura venga osservata la configurazione x_ω.

Per massimizzare questa probabilità la sequenza Oracolo – Grover va eseguita per un numero di iterazioni pari a √N.

Oracolo U_ω

L'operatore oracolo serve per marcare un particolare stato x_ω del registro, rappresentante la soluzione. Detta x la configurazione di qubit in ingresso, l'operatore restituisce lo stesso stato x se $x \neq x_\omega$, lo stato negato $-x_\omega$ per $x = x_\omega$.

Operatore diffusione

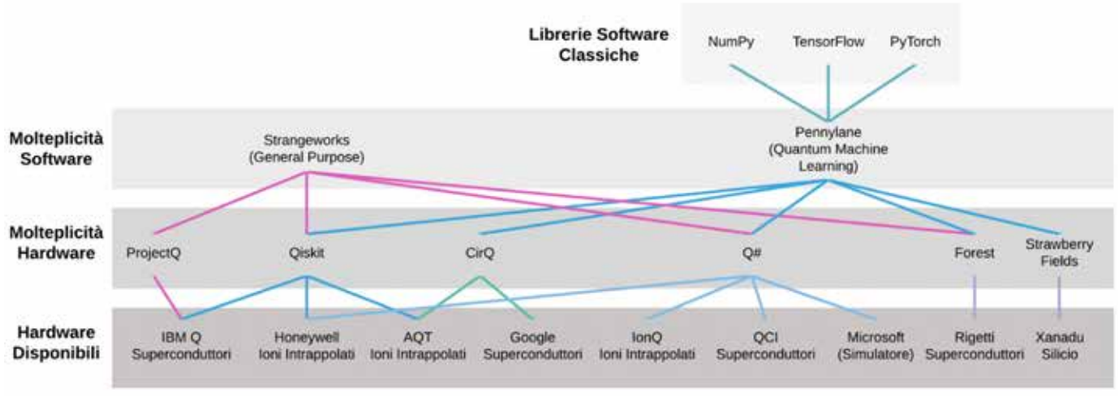
L'operatore diffusione di Grover è così definito:

$H^{\otimes n}.(2|0^n\rangle\langle 0^n| - I_n).H^{\otimes n}$

A valle dell'oracolo concorre a sbilanciare la configurazione di qubit, indirizzandola verso quella marcata.

L' espressione tra parentesi altro non è, se non una regola simile a quella dell'oracolo, che però restituisce 1 per la configurazione $|0^n\rangle$ (tutti i qubit a 0) e -1 per tutte le altre.

vincenzo.cuciti@telecomitalia.it



2
Hardware/Software stack per QGA

per Fujitsu anziché duemila fisici di D-Wave), connettività tra gli stessi per stabilire le relazioni tra qubit e di conseguenza la loro fruizione (l'architettura emulata Fujitsu è fully-connected mentre in quella D-Wave la connettività tra qubit è limitata) ed assenza di effetti legati alla decoerenza durante l'esecuzione.

Le piattaforme classiche su cui eseguire gli emulatori non si limitano a dei calcolatori; infatti sono disponibili risolutori QUBO sotto forma di acceleratori hardware su Field Programmable Gate Array [12] pensati per poter ottenere vantaggi computazionali immediati on-premises.

Oltre a Fujitsu anche Toshiba e Microsoft sviluppano soluzioni hardware o software classiche (quantum-inspired) per la risoluzione di problemi di ottimizzazione.

Anche per la famiglia degli annealer valgono le stesse considerazioni evidenziate nella sezione relativa ai QGA, sull'impiego dei quantum-inspired in modalità on-premises e in particolare a livello di edge.

Annealing con Quantum Gate Array ed evoluzioni

Anche i computer QGA possono risolvere problemi di ottimizzazione attraverso tecniche ibride quanto-classiche di tipo variazionale.

In questo modello lo stesso circuito quantistico – inizializzato anche in questo caso nello stato di ground - è eseguito più volte, con i contributi delle porte quantistiche iterativamente aggiornati da un minimizzatore classico di una funzione di costo, finché non si converge ad uno stato in cui la probabilità corrispondente alla soluzione del problema è predominante rispetto a quelle di tutti gli altri stati possibili.

Le due procedure variazionali più adoperate sono il Quantum Approximate Optimization Algorithm (QAOA) [13], che emula l'evoluzione temporale di un QA, ed il Variational Quantum Eigensolver (VQE) [14], un algoritmo concepito per determinare l'energia minima di un sistema quantistico.

Ambienti di sviluppo per Quantum Computing

Sviluppo software ed accessibilità hardware

Oggigiorno il software per QC consiste principalmente in librerie Python interfacciabili ad hardware quantistico via-cloud o ad hardware classico adoperato come simulatore od emulatore a cui si accede o localmente o via-cloud.

Queste librerie sono sviluppate o da produttori di hardware quantistico – vedasi Qiskit di IBM Q, Cirq di Google, Forest di Rigetti, Strawberry Fields di Xanadu e Leap di D-Wave - o da startup che si dedicano allo sviluppo di soluzioni software - per esempio Orquestra di Zapata Computing - o da realtà accademiche come ProjectQ di ETH.

Microsoft si distingue per aver sviluppato il linguaggio di programmazione Q# (derivato da C#) attualmente utilizzabile nel Microsoft Quantum Development Kit.

La disponibilità piuttosto limitata di hardware programmabili rende necessario la definizione di ambienti multipiattaforma, che consentano quindi l'utilizzabilità del software sviluppato su quante più piattaforme possibili, tanto hardware quanto software.

Nel caso dell'hardware, la stessa libreria è interfacciabile ad hardware differenti non necessariamente dello stesso costruttore, previa autorizzazione all'accesso all'hardware.

Per esempio, Qiskit consente l'esecuzione di algoritmi quantistici sui quantum computer di IBM Q a superconduttori e di Alpine Quantum Technologies e Honeywell a ioni intrappolati, mentre il codice sviluppato in ambiente Microsoft può essere eseguito tramite i servizi cloud di Azure Quantum su hardware superconduttivo di Quantum Circuits, Inc. (spin-off dell'università di Yale) e su hardware a ioni intrappolati Honeywell e IonQ.

Per quanto concerne il software, la stessa libreria special-purpose o lo stesso ambiente general-purpose possono essere adoperate in sinergia con software sviluppato da terzi. Il caso più emblematico è costituito da Pennylane di Xanadu, una libreria Python per il Quantum Machine Learning ed interfacciabile con framework per computazione quantistica quali Qiskit, Cirq, Forest e Strawberry Fields attraverso librerie classiche e diffuse per il Machine Learning quali NumPy, Tensorflow e PyTorch. A questa categoria appartiene anche il software general-purpose sviluppato da Strangeworks che può interagire con software Qiskit, Cirq, ProjectQ e Forest.

Si potrebbero definire delle linee guida per la scelta della piattaforma di esecuzione di un algoritmo quantistico:

- se il numero di operazioni richieste per risolvere un problema è tale per cui la decoerenza è trascurabile, è in generale preferibile adoperare hardware quantistico, tenendo anche conto che l'esecuzione non risentirebbe dell'overhead computazionale intrinseco della simulazione o dell'emulazione classiche di un sistema quantistico;
- se al contrario il problema risulta irrisolvibile da un quantum computer reale, la simulazione in assenza di rumore con hardware classico risulta la scelta più ragionevole.

Dal momento che l'accesso via-cloud all'hardware reale è condiviso da migliaia di utenti ogni giorno, ogni costruttore stabilisce un limite massimo di esecuzioni giornaliere o mensili per ciascun utente.
La simulazione classica in presenza di rumore potrebbe risultare pertanto preferibile per la prototipizzazione o l'ingegnerizzazione del software, dal momento che consentirebbe di stimare i risultati attesi da un dispositivo reale e di ottimizzare l'esecuzione

senza dover “consumare” accessi all’hardware quantistico.

Modellizzazione algoritmica

Sebbene non si sia ancora consolidata una metodologia di definizione di algoritmi per QA o QGA, sia per la complessità del formalismo sia per i limiti intrinseci dell’hardware, in entrambi i casi è riscontrabile un chiaro orientamento verso un approccio ibrido iterativo in cui un computer classico non solo pilota un processore quantistico, ma ne elabora anche le soluzioni per ripresentargli un sotto-problema.

Il QA ha sostanzialmente ridotto i tempi ed aumentato l’affidabilità della risoluzione di problemi QUBO, per i quali erano già disponibili dalla fine del secolo scorso delle metodologie risolutive classiche.

Da un punto di vista metodologico due possibili approcci possono essere adoperati per la risoluzione di problemi di ottimizzazione: uno che porta allo sviluppo di una funzione di costo (Hamiltoniana) da minimizzare, l’altro basato su metodi che legano la rappresentazione del problema all'architettura del QA.
Il problema dell'assegnazione dei PCI nelle reti LTE e 5G, descritto in un articolo del notiziario tecnico di aprile [16], è stato affrontato con entrambi gli approcci.

L’approccio legato all’Hamiltoniana riconduce il problema ad un modello che consiste nel trovare la combinazione ottimale (Optimization) di un set di variabili binarie (Binary), che possono cioè assumere due soli valori mutuamente esclusivi (0/1, si/no, on/off...), minimizzando un polinomio quadratico (Quadratic) che, oltre a modellare il problema, include anche i vincoli a cui le variabili devono sottostare, arrivando così ad una formulazione matematica compatta, formalmente senza vincoli, perché inglobati e quindi Unconstrained.

Mettendo assieme le varie parole chiavi in inglese si ottiene l’acronimo QUBO. La definizione del polinomio segue regole codificate [11] da cui si può costruire una matrice QUBO che, ad esempio, associa un peso tra ogni variabile e ad ogni risorsa ed è fornita come input al QA. La costruzione di questa matrice a partire dall'Hamiltoniana può essere semplificata e automatizzata attraverso librerie quali PyQUBO [17] accessibili via API, anche se rispetto ad uno sviluppo ad-hoc potrebbe essere meno performante.

Il secondo approccio (di prossima pubblicazione) risolve l'assegnazione dei PCI tramite una serie di bisezioni. Il set iniziale di siti viene ripartito in due sottogruppi distinti, compilando la matrice QUBO in modo che siano minime le relazioni tra i due sottogruppi (minima l’interferenza). Il procedimento è iterato su ogni sottogruppo che si è generato nella suddivisione precedente, finché le dimensioni si riducono al punto da trovare una soluzione che minimizza il numero di PCI assegnati.

Una volta completata la serie di bisezioni successive si esegue una fase di retroazione, che consiste nel raggruppare un numero di sottogruppi pari ad una potenza di due, smantellando quindi parte del lavoro di ripartizione e creando così un nuovo sottogruppo maggiore sul quale viene applicato nuovamente il processo di suddivisione; questa retroazione permette di uscire da eventuali minimi locali. Il procedimento termina assegnando i PCI alle singole antenne.
Le dimensioni dei sottogruppi si riducono progressivamente ad ogni iterazione fino ad arrivare a dimensioni gestibili con le attuali potenze di calcolo dei quantum computer QA e quindi il processo scala efficacemente anche per migliaia di celle, senza la necessità di dover applicare algoritmi di partizionamento sui set di celle.

I computer QGA, pur essendo teoricamente più completi e versatili dal punto di vista computazionale rispetto ai QA, risultano meno maturi per applicazioni pratiche tanto in termini di qubit equipaggiati (decine anziché migliaia) quanto in termini di metodologie di sviluppo di algoritmi.

Tuttavia l’esperienza acquisita negli ultimi due decenni consente di definire delle procedure progettuali generalmente valide. Innanzitutto si potrebbe osservare che, pur essendo i fenomeni alla base della computazione quantistica in molti casi controintuitivi, il formalismo matematico che li descrive è preciso.

L'approccio generalmente più affidabile è costituito dall’algebra lineare complessa, secondo cui tutte le operazioni sono matrici di evoluzioni unitarie che modificano lo stato del sistema quantistico.

Cercando una declinazione più prettamente circuitale, si può osservare che entrambi gli algoritmi di Grover e variazionali – i quali sono oggigiorno quelli di maggiore utilizzo per la loro versatilità - presentano delle ripetizioni delle fasi di valutazione e di consolidamento (si veda la Figura 1) finalizzate ad una massimizzazione della probabilità di ottenere la migliore soluzione del problema, con la sostanziale differenza che nel caso di Grover l’iterazione è eseguita interamente sul Quantum Computer mentre nel caso variazionale questa coinvolge anche un computer classico.

Si può dunque concludere che nella formulazione di un algoritmo per architettura QGA è solitamente richiesto di ripetere delle operazioni per consolidare il risultato finale.
La scelta dell’uno o dell’altro algoritmo dipende notevolmente dalla complessità del circuito da progettare e dal tipo di soluzione da individuare.
Se la soluzione del problema è assoluta (una e una sola, eventualmente comune a più dati) è possibile adoperare l’algoritmo di Grover, la cui valutazione corrisponde all’etichettatura della soluzione con un apposito circuito chiamato oracolo, che valuta simultaneamente la condizione di etichettatura su tutti i possibili stati in sovrapposizione/dati del dataset. L’oracolo potrebbe tuttavia risultare troppo complesso - in termini di numero totale di porte quantistiche e numero di qubit ausiliari richiesti - per un’esecuzione su hardware reale, pertanto i risolutori Grover per casi d’uso concreti sono attualmente eseguiti su simulatori classici di qubit ideali.

Se la soluzione del problema è invece relativa (ottimale tra una serie di soluzioni possibili), l’approccio variazionale è preferibile.

Si potrebbe partire dal QAOA, riproponendo quindi le metodologie di modellizzazione del QA su un QGA, con l’evidente limite legato al minor numero di qubit adoperabili.

Questo algoritmo si basa su un circuito quantistico fissato (simulatore del QA) e una funzione di costo non specifica e scelta dal progettista.

Un interessante vantaggio dei circuiti variazionali rispetto a quelli fissati come quello di Grover è che l'ottimizzatore classico cerca di compensare gli effetti della decoerenza durante l’esecuzione; tuttavia se il numero di operazioni richieste è tale per cui la decoerenza risulta in ogni caso significativa, potrebbe essere preferibile il VQE.

Questo si distingue per l’utilizzo di una specifica funzione di costo - il valore atteso di un sistema quantistico, che risulta sempre maggiore o uguale all’energia minima del sistema - ed adopera come circuito un ansatz, ossia un circuito parametrico non strettamente legato al problema e concepito per ispezionare lo spazio delle soluzioni in funzione del valore atteso, garantendo la possibilità di generare entanglement [18].

Pur essendo un ansatz potenzialmente più semplice del circuito del QAOA, il calcolo del valore atteso potrebbe richiedere un numero maggiore di operazioni di quello della funzione di costo del QAOA, aumentando così i tempi di esecuzione dell’intera procedura iterativa.

Conclusioni

L’utilizzo su larga scala del QC è ancora lontano, ma può essere utilizzato fin da subito traendo i benefici che derivano dal cosiddetto vantaggio quantistico.

Proprio perché non si dispone ancora del “Universal Quantum Computing”, gli use cases trattabili devono essere selezionati in funzione delle potenzialità e delle modalità di impiego attuali della tecnologia, secondo un processo di verifica che

tenga conto indicativamente di alcune linee guida:

1. l'area annealing è più matura e pronta all'uso rispetto ai modelli che si ispirano al universal quantum computer (gate array)
2. l'annealing è più adatto per problemi di ottimizzazione (combinatoria), categoria tra le più importanti e diffuse in vari campi in particolare anche nel settore delle telecomunicazioni
3. la complessità computazionale degli use cases deve essere commisurata alla potenza di calcolo delle soluzioni attuali di QC e può essere gestita dimensionando i dati con tecniche di partizionamento di cui esiste una consolidata esperienza di algoritmi classici
4. l'accesso alle soluzioni di QC avviene via cloud e quindi questa modalità può essere utilizzata per use case non-real o near-real time
5. sono disponibili soluzioni quantum-inspired basate su tecnologia tradizionale, che possono essere installate on-premises, con una maggiore maturità nell'area annealing e che possono essere impiegate per applicazioni real-time, eventualmente dimensionando opportunamente il problema (vedi punto 3 precedente)

Prendendo come riferimento questo quadro ci si rende conto come il numero delle applicazioni candidabili per essere sviluppate già adesso in ottica QC è elevato. Il loro numero è destinato col tempo ad ampliarsi, coerentemente con il miglioramento della tecnologia e di conseguenza delle "linee guida" che progressivamente imporranno sempre meno limiti.

Siamo ormai nella fase in cui possiamo sfruttare la potenza computazionale del QC ■

Riferimenti

1. KATWALA, A. (2020, 03 18). Inside big tech's high-stakes race for quantum supremacy. Tratto da Wired: https://www.wired.co.uk/article/quantum-supremacy-google-microsoft-ibm
2. KATWALA, A. (2020, 03 05). Quantum computers will change the world (if they work). Tratto da Wired: https://www.wired.co.uk/article/quantum-computing-explained
3. Moltzau, A. (2019, 10 13). Quantum Information and AI. Tratto da Medium: https://towardsdatascience.com/quantum-computing-and-ai-789fc9c28c5b
4. A Quantum Computing Use Case Roadmap from IBM. (s.d.). Tratto da Quantum Computing Report: https://quantumcomputingreport.com/a-quantum-computing-application-roadmap-from-ibm/
5. David Deutsch and Roger Penrose - 1997 - Quantum theory, the Church–Turing principle and the universal quantum computer, Proc. R. Soc. Lond. A40097-117, http://doi.org/10.1098/rspa.1985.0070
6. Jack Krupansky - 2019- What Is a Universal Quantum Computer?, https://medium.com/@jackkrupansky/what-is-a-universal-quantum-computer-db183fd1f15a
7. Travis S. Humble, Himanshu Thapliyal, Edgard Munoz-Coreas, Fahd A. Mohiyaddin, Ryan S. Bennink - 2018- Quantum Computing Circuits and Devices, https://arxiv.org/pdf/1804.10648.pdf
8. Peter W. Shor - 1994 - Algorithms for quantum computation: discrete logarithms and factoring, Proceedings 35th Annual Symposium on Foundations of Computer Science. IEEE Comput. Soc. Press: 124–134, doi:10.1109/sfcs.1994.365700
9. Lov Grover - 1996 - A fast quantum mechanical algorithm for database search, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, (May 1996), https://arxiv.org/abs/quant-ph/9605043
10. 10. Tadashi Kadowaki, Hidetoshi Nishimori - 1998 - Quantum annealing in the transverse Ising model, Physical Review E 58.5, https://journals.aps.org/pre/abstract/10.1103/PhysRevE.58.5355
11. Fred Glover, Gary Kochenberger, Yu Du - 2019 - A Tutorial on Formula ting and Using QUBO Models, https://arxiv.org/abs/1811.11538
12. 3. Yu Zou, Mingjie Lin - 2020 - Massively Simulating Adiabatic Bifurcations with FPGA to Solve Combinatorial Optimization, https://dl.acm.org/doi/pdf/10.1145/3373087.3375298
13. Edward Fahri, Jeffrey Goldstone - 2014 - A Quantum Approximate Optimization Algorithm, https://arxiv.org/pdf/1411.4028.pdf
14. Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, Jeremy L. O'Brien - 2014 - A variational eigenvalue solver on a photonic quantum processor, https://www.nature.com/articles/ncomms5213.pdf
15. Travis S. Humble, Himanshu Thapliyal, Edgard Munoz-Coreas, Fahd A. Mohiyaddin, Ryan S. Bennink - 2018- Quantum Computing Circuits and Devices, https://arxiv.org/pdf/1804.10648.pdf
16. Andrea Boella, Michele Federico, Giuseppe Minerva, Mauro Alberto Rossotto - 2020 - Quantum computing per l'ottimizzazione delle reti mobili (4.5G e 5G), https://www.telecomitalia.com/content/tiportal/it/notiziariotecnico/edizioni-2020/n-1-2020/Quantum-Computing-ottimizzazione-delle-reti-mobili.html
17. Kotaro Tanahashi, Shinichi Takayanagi, Tomomitsu Motohashi, Shu Tanaka – 2019 - Application of Ising Machines and Software Development for Ising Machines, https://journals.jps.jp/doi/full/10.7566/JPSJ.88.061010
18. Samuel Yen-Chi Chen, Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Xiaoli Ma, Hsi-Sheng Goan - 2019 - Variational Quantum Circuits for Deep Reinforcement Learning, https://arxiv.org/pdf/1907.00397.pdf

Librerie software per Quantum Computing e di supporto

1. NumPy - https://numpy.org/
2. TensorFlow - https://www.tensorflow.org/
3. PyTorch - https://pytorch.org/Pennylane - https://pennylane.ai/
4. Qiskit - https://qiskit.org/
5. Cirq - https://cirq.readthedocs.io/en/stable/
6. Strawberry Fields - https://strawberryfields.readthedocs.io/en/stable/#
7. Forest - http://docs.rigetti.com/en/stable/
8. Q# - https://www.microsoft.com/en-us/quantum/development-kit
9. ProjectQ - https://projectq.ch/

Costruttori di hardware per Quantum Computing

1. IBMQ - https://www.ibm.com/quantum-computing/
2. Honeywell - https://www.honeywell.com/en-us/company/quantum
3. IonQ - https://ionq.com/
4. AQT - https://www.aqt.eu/
5. Google - https://research.google/teams/applied-science/quantum/
6. Microsoft - https://www.microsoft.com/en-us/quantum
7. Rigetti - https://rigetti.com/
8. Xanadu - https://www.xanadu.ai/
9. Quantum CIrcuits, Inc. - https://quantumcircuits.com/
10. D-Wave - https://www.dwavesys.com/

Servizi cloud per Quantum Computing

1. IBM Quantum Experience - https://quantum-computing.ibm.com/
2. Microsoft Azure Quantum - https://azure.microsoft.com/en-us/services/quantum/#features
3. Strangeworks - https://strangeworks.com/
4. DWave - https://cloud.dwavesys.com/

Giovanni Amedeo Cirillo *giovanni_cirillo@polito.it*

Ha conseguito la Laurea e la Laurea Magistrale in Ingegneria Elettronica al Politecnico di Torino nel 2016 e nel 2018 rispettivamente. È attualmente dottorando in Ingegneria Elettronica presso il laboratorio VLSI del Dipartimento di Elettronica e Telecomunicazioni del Politecnico di Torino, sotto la supervisione del Prof. Maurizio Zamboni, della Prof.ssa Mariagrazia Graziano e della Dott.ssa Giovanna Turvani. Le sue attività di ricerca sono principalmente dedicate allo sviluppo di un framework multilivello per la simulazione e l'ingegnerizzazione di tecnologie per Quantum Computing, con attuale particolare interesse per quelle molecolari (per ulteriori informazioni https://www.vlsilab.polito.it/quantumcomputing/) ■

Filippo Gandino *filippo.gandino@polito.it*

Filippo Gandino (Socio IEEE) ha conseguito i diplomi M.S. e Ph.D. in Ingegneria Informatica presso il Politecnico di Torino, rispettivamente nel 2005 e nel 2010. Attualmente è Professore Associato presso il Dipartimento di Automatica e Informatica del Politecnico di Torino. I suoi interessi di ricerca includono ubiquitous computing, RFID, WSN, sicurezza e privacy, modellazione di rete e quantum computing ■

Edoardo Giusto *edoardo.giusto@polito.it*

Ha conseguito la Laurea e Laurea Magistrale in Ingegneria Informatica al Politecnico di Torino nel 2015 e nel 2017 rispettivamente. È attualmente dottorando in Ingegneria Informatica presso il Dipartimento di Automatica e Informatica (DAUIN) del Politecnico di Torino, sotto la supervision del Prof. Maurizio Rebaudengo, del Prof. Bartolomeo Montrucchio e del Prof. Filippo Gandino. I suoi interessi di ricerca comprendono le Wireless Sensor Networks, l'IoT e il Quantum Computing ■

Giovanni Mondo *giovanni.mondo@telecomitalia.it*

Laurea in Ingegneria Elettronica e Dottorato in Robotica presso l'Università di Genova, Laurea triennale in Economia presso UniNettuno. Inizia a collaborare nella ricerca di TIM (all'epoca CSELT) nel 1998 e viene assunto nel 2001. Ha collaborato a diversi progetti legati ai servizi per le reti mobili cellulari, principalmente come sviluppatore lato server. Da fine 2018 alle attività di amministrazione server e Information Visualization affianca quella di analisi del Quantum Computing in generale e dello sviluppo di modelli ispirati al Quantum Annealing in particolare ■