Il parallelismo quantistico

La principale prerogativa del calcolo quantistico è certamente costituita dal suo parallelismo intrinseco: la possibilità di effettuare in sovrapposizione più computazioni ciascuna delle quali influisca sul risultato interferendo con le altre è ad esempio alla base della superiore efficienza dell'algoritmo di Shor rispetto ad ogni algoritmo classico per la fattorizzazione dei numeri interi. Nonostante questo non esiste, al momento, alcuna indicazione diretta riguardo al modo di mettere a frutto tale caratteristica nella soluzione di una classe abbastanza generale di problemi: l'utilizzo del parallelismo quantistico necessita, di volta in volta, di soluzioni *ad hoc*. Questa apparente mancanza di generalità costituisce probabilmente, al presente, uno dei principali ostacoli nello sviluppo di ulteriori algoritmi quantistici per la soluzione di problemi non banali.

In linea generale, comunque, il parallelismo quantistico sembra essere utilizzabile con profitto nella determinazione di proprietà globali di un insieme di dati che classicamente richiederebbero, per il loro calcolo, l'iterazione di una stessa computazione su valori successivi dell'ingresso. L'esempio principe è, a questo riguardo, ancora una volta l'algoritmo di Shor, nel quale tale proprietà globale è costituita dalla periodicità di una funzione. Per illustrare questa tesi riportiamo nel seguito un esempio classico con valore puramente euristico introdotto da D. Deutsch nel 1985 e successivamente ripreso da A. Ekert ([Deutsch], [Ekert]). Per permettere una più agevole discussione dell'esempio premettiamo, nel paragrafo seguente, una descrizione semplificata del calcolatore quantistico.

Il calcolatore quantistico a due registri

Una computazione è un processo algoritmico finito che produce simboli (l'"uscita") le cui proprietà astratte dipendono, fissata una certa interpretazione, da altri simboli (l'"ingresso"). Con una opportuna numerazione di tali simboli, essa può quindi essere descritta in tutta generalità come il calcolo di una funzione

$$f: \{0,1,...m-1\} \rightarrow \{0,1,... n-1\}$$

dove m ed n sono numeri naturali. Un computer classico evolve i simboli corrispondenti a ciascun ingresso, 0, 1, ..., m-1 in quelli corrispondenti a ciascuna uscita, f(0), f(1), ..., f(m-1). Nel caso generale di funzioni non bigettive un calcolatore quantistico, a causa del carattere unitario e quindi reversibile della sua evoluzione, deve effettuare questa computazione mantenendo la memoria degli ingressi (cfr. pag. 27).

Immaginiamo il calcolatore quantistico come una "scatola nera" fornita di due registri quantistici, uno per i dati in ingresso ed uno per i dati in uscita, supponendo che il primo mantenga inalterato il suo contenuto fino al termine della computazione[I]. Rappresentiamo ogni possibile ingresso x con uno stato $|x\rangle$ del primo registro ed analogamente ogni possibile uscita y con uno stato $|y\rangle$ del secondo registro. I vettori $|x\rangle$ appartengono allo spazio di Hilbert m-dimensionale $\mathcal{H}_1$, ed i vettori $|y\rangle$ appartengono allo spazio n-dimensionale $\mathcal{H}_2$. Stati corrispondenti ad ingressi ed uscite differenti sono ortogonali: $\langle x|x'\rangle = \delta_{xx'}$, $\langle y|y'\rangle = \delta_{yy'}$. Il calcolo di f corrisponde allora all'evoluzione dei due registri secondo una trasformazione unitaria $\mathcal{H}_1 \otimes \mathcal{H}_2 \xrightarrow{U_f} \mathcal{H}_1 \otimes \mathcal{H}_2$ la cui azione sugli stati $|x\rangle|0\rangle$ sia

[I] Dovremo inoltre aver cura di preparare il secondo registro in uno stato standard $|0\rangle$, in quanto la cancellazione di un dato casuale è un'operazione irreversibile che non può essere effettuata dal calcolatore quantistico.

$$|x\rangle|0\rangle \xrightarrow{U_f} |x\rangle|f(x)\rangle$$

(notiamo che questa condizione è soddisfatta da una famiglia infinita di trasformazioni U_f che sono da considerarsi equivalenti ai fini della computazione). La proprietà di universalità del calcolatore quantistico garantisce l'esistenza di una adeguata trasformazione U_f per ogni funzione computabile f.

Nel seguito concentreremo la nostra attenzione sullo stato complessivo $|\Phi\rangle$ dei due registri al termine della computazione. Indicato con

$$\varphi_{11}|\alpha_1\rangle|\beta_1\rangle + \varphi_{12}|\alpha_1\rangle|\beta_2\rangle + \ldots + \varphi_{mn}|\alpha_m\rangle|\beta_n\rangle$$

lo sviluppo di $|\Phi\rangle$ su una base $\{|\alpha_i\rangle|\beta_j\rangle\}$ di $\mathcal{H}_1 \otimes \mathcal{H}_2$ supporremo che sia sempre possibile compiere congiuntamente sui due registri un'unica misura che fornisca il risultato $|\alpha_1\rangle|\beta_1\rangle$ con probabilità $|\varphi_{11}|^2$, $|\alpha_1\rangle|\beta_2\rangle$ con probabilità $|\varphi_{12}|^2$ e così via. L'assunzione di poter misurare un'osservabile completa sullo spazio prodotto di due o più sistemi (centinaia di bit quantistici nel caso dell'algoritmo di Shor) è onnipresente nelle attuali trattazioni del calcolo quantistico. Per quanto poco realistica essa possa sembrare è forse il caso di ricordare che, per computazioni di una certa complessità, il problema centrale risiede nel calcolo coerente della funzione f; una volta in grado di compiere computazioni arbitrarie lo stesso calcolatore quantistico potrebbe forse essere programmato in modo da rendere più agevole, mediante una opportuna trasformazione unitaria, la misura dei risultati. Lo studio delle limitazioni imposte agli algoritmi dalla scarsa realizzabilità pratica di misure su sistemi troppo complessi rappresenta un problema aperto la cui analisi non rientra, però, nello scopo di questa trattazione.

Un semplice esempio

Come sopra visto la computazione della funzione f(x) da parte di un calcolatore quantistico corrisponde, per ogni x fissato, alla seguente evoluzione dei registri di ingresso e di uscita del calcolatore:

$$|x\rangle|0\rangle \xrightarrow{U_f} |x\rangle|f(x)\rangle$$

Il parallelismo intrinseco garantito dal principio di sovrapposizione ci permette ora di preparare il primo registro in una sovrapposizione di tutti i possibili ingressi e di calcolare *tutti* i valori di f(x) applicando *una sola volta* la trasformazione U_f, ovvero al costo computazionale di un singolo calcolo:

$$\frac{1}{\sqrt{m}}\left[\sum_{k=0}^{m-1}|k\rangle\right]|0\rangle \xrightarrow{U_f} \frac{1}{\sqrt{m}}\left(|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle + \ldots + |m-1\rangle|f(m-1)\rangle\right).$$

Questo è troppo bello per essere vero. Ci sono, in effetti, alcune limitazioni. Notiamo anzitutto che lo stato in cui il calcolatore si arresta non corrisponde in genere ad uno stato puro dei registri di ingresso e di uscita presi separatamente, ma è una generalizzazione di uno stato correlato di tipo EPR. Non esiste sfortunatamente alcun processo di misura in grado di estrarre da tale stato tutti i valori f(0), f(1), ..., f(m-1), né si può pensare di crearne m copie in modo da misurare i diversi valori separatamente (non-clonabilità). Purtuttavia esistono misurazioni che, sfruttando l'interferenza tra computazioni parallele, permettono di estrarre dallo stato alcune *proprietà globali* di tutti i valori calcolati di f(x), mettendo così a frutto il parallelismo quantistico; una illustrazione tipica di questo fatto è costituita, in letteratura, dal seguente esempio [Deutsch].

Supponiamo che sia assegnata una funzione

$$f: \mathbf{A} \rightarrow \mathbf{A}$$

dove **A** è un sottoinsieme finito dell'insieme **N** dei numeri naturali, e che venga richiesto stabilire se f è costante o meno sul suo dominio. Se disponiamo di un

calcolatore classico programmato per accettare in ingresso un punto x del dominio e restituire f(x), possiamo risolvere il problema calcolando f(x) per ogni x appartenente ad **A** e confrontando i risultati ottenuti. Nel caso particolarmente semplice in cui **A**={0, 1} sarà quindi sufficiente calcolare f(0) ed f(1) e confrontarli; così facendo avremo risolto il problema utilizzando due volte il nostro calcolatore classico. E' evidente che una sola computazione non fornisce alcuna indicazione sul risultato del confronto, in quanto dalla conoscenza di f(0) non si può inferire nulla su f(1) e viceversa. Sorprendentemente, però, utilizzando *una sola volta* un calcolatore quantistico programmato per calcolare f e misurando opportunamente i risultati si ha una probabilità ½ di stabilire con certezza se f(0)=f(1) o f(0)≠f(1).

Per renderci conto di questo, adattiamo il nostro modello "a scatola nera" del calcolatore quantistico al calcolo della funzione assegnata

$$f: \{0,1\} \rightarrow \{0,1\}.$$

Si noti che sono ora possibili soltanto quattro scelte per la funzione f:

$$\begin{cases} f_1(0)=0 \\ f_1(1)=0 \end{cases} \qquad \begin{cases} f_2(0)=1 \\ f_2(1)=1 \end{cases} \qquad \begin{cases} f_3(0)=0 \\ f_3(1)=1 \end{cases} \qquad \begin{cases} f_4(0)=1 \\ f_4(1)=0 \end{cases}$$

Può apparire naturale rappresentare queste quattro funzioni in termini di matrici 2x2; rinviamo però l'uso di tale rappresentazione al prossimo paragrafo facendo qui direttamente riferimento allo stato dei due registri del calcolatore, anche al fine di restare vicini all'esposizione di questo esempio normalmente riportata in letteratura.

Essendo il dominio ed il codominio costituiti dall'insieme {0,1}, i registri di ingresso e di uscita saranno rappresentati ciascuno da un sistema a due stati o bit quantistico; indicheremo la "base computazionale" di un bit con $\{|0\rangle,|1\rangle\}$. L'evoluzione dei due registri per effetto della computazione sarà descritta da una trasformazione unitaria U_f tale che si abbia

$$|0\rangle|0\rangle \xrightarrow{U_f} |0\rangle|f(0)\rangle \quad \text{e} \quad |1\rangle|0\rangle \xrightarrow{U_f} |1\rangle|f(1)\rangle.$$

Volendo utilizzare una sola volta il calcolatore quantistico, è a questo punto conveniente preparare il registro di ingresso in una sovrapposizione dei due punti del dominio ed effettuare la computazione

$$\left[\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\right]|0\rangle \xrightarrow{U_f} \frac{1}{\sqrt{2}}(|0\rangle|f(0)\rangle+|1\rangle|f(1)\rangle).$$

Esibiremo ora una osservabile completa sullo spazio prodotto dei due registri la cui misura permette con probabilità ½ di stabilire con sicurezza se f(0)=f(1) o f(0)≠f(1) e che nel rimanente 50% dei casi fornisce un risultato non significativo che segnala che l'algoritmo ha fallito ed è necessario ripetere la computazione.

La costruzione di tale osservabile è assolutamente *ad hoc* ed è basata sul fatto che esistono unicamente quattro istanze diverse di questo problema, corrispondenti alle quattro funzioni f sopra riportate. A seconda dell'istanza del problema che stiamo considerando, ovvero della funzione f assegnata, lo stato dei due registri del calcolatore in seguito alla computazione sarà perciò uno dei seguenti:

$$|f_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|0\rangle) \qquad |f_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle|1\rangle+|1\rangle|1\rangle)$$

$$|f_3\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle+|1\rangle|1\rangle) \qquad |f_4\rangle = \frac{1}{\sqrt{2}}(|0\rangle|1\rangle+|1\rangle|0\rangle)$$

Per risolvere il problema è quindi necessario discriminare gli stati $|f_1\rangle$ ed $|f_2\rangle$, corrispondenti alla funzione costante a zero e costante ad uno, dai rimanenti. Questo può essere fatto misurando sullo spazio prodotto dei due registri un'osservabile **G** che abbia i seguenti autostati non degeneri:

$$|\text{UGUALI}\rangle = \frac{1}{2}(|0\rangle+|1\rangle)(|0\rangle-|1\rangle) \qquad |\text{DIVERSI}\rangle = \frac{1}{2}(|0\rangle-|1\rangle)(|0\rangle-|1\rangle)$$

$$|\text{FALLIMENTO}\rangle = \frac{1}{2}(|0\rangle+|1\rangle)(|0\rangle+|1\rangle) \qquad |\text{ERRORE}\rangle = \frac{1}{2}(|0\rangle-|1\rangle)(|0\rangle+|1\rangle)$$

Per rendersi conto di ciò è sufficiente osservare lo sviluppo degli stati finali $|f_1\rangle$, $|f_2\rangle$, $|f_3\rangle$, $|f_4\rangle$ sulla base di tali autostati:

$$|f_1\rangle = \frac{1}{\sqrt{2}}(|\text{UGUALI}\rangle + |\text{FALLIMENTO}\rangle),$$

$$|f_2\rangle = -\frac{1}{\sqrt{2}}(|\text{UGUALI}\rangle - |\text{FALLIMENTO}\rangle),$$

$$|f_3\rangle = \frac{1}{\sqrt{2}}(|\text{DIVERSI}\rangle + |\text{FALLIMENTO}\rangle),$$

$$|f_4\rangle = -\frac{1}{\sqrt{2}}(|\text{DIVERSI}\rangle - |\text{FALLIMENTO}\rangle).$$

L'osservabile **G** gode pertanto delle seguenti proprietà:

- Un risultato "**UGUALI**" può essere ottenuto solo quando i registri si trovano in uno degli stati $|f_1\rangle$ e $|f_2\rangle$, corrispondenti alle due funzioni per le quali f(0) ed f(1) sono effettivamente uguali.
- Analogamente un risultato "**DIVERSI**" può essere ottenuto solo a partire da uno degli stati $|f_3\rangle$ e $|f_4\rangle$, corrispondenti alle due funzioni per le quali f(0) è diverso da f(1).
- Un risultato "**FALLIMENTO**" può essere ottenuto con probabilità ½ in qualunque dei quattro stati $|f_1\rangle$, $|f_2\rangle$, $|f_3\rangle$, $|f_4\rangle$ si trovino i registri del calcolatore; esso non fornisce alcuna indicazione utile e costringe a ripetere la computazione.
- Un risultato "**ERRORE**" può verificarsi soltanto nel caso che un errore si sia introdotto nella computazione di f, in quanto $|\text{ERRORE}\rangle$ è ortogonale agli stati $|f_1\rangle$, $|f_2\rangle$, $|f_3\rangle$ e $|f_4\rangle$ ottenibili computando le quattro funzioni.

Notiamo infine che gli stati $|f_1\rangle$, $|f_2\rangle$, $|f_3\rangle$ ed $|f_4\rangle$ generano, come è evidente dal loro sviluppo, un sottospazio a dimensione tre dello spazio quadridimensionale degli stati finali dei due registri. Questo fatto preclude la

possibilità di ideare una misura che, distinguendoli con sicurezza, permetta di risalire dallo stato finale del calcolatore ad una delle quattro funzioni f_1, f_2, f_3 ed f_4. In altri termini, il parallelismo quantistico ci permette di utilizzare una sola computazione per stabilire con probabilità ½ la proprietà globale f(0)=f(1) o f(0)≠f(1), ma impedisce di estrarre da tale computazione entrambi i valori di f(0) ed f(1).

Ci si può ragionevolmente domandare che vantaggio presenti una soluzione quantistica che fornisce la risposta utilizzando una computazione anziché due ma con probabilità di successo ½, e quindi in un tempo medio uguale al caso classico. Tuttavia, supponiamo che dal risultato della nostra computazione dipenda una decisione cruciale da prendere nell'arco di 24 ore, come ad esempio acquistare o vendere dei titoli sul mercato azionario, e che calcolare un singolo valore di f(k) richieda quasi un giorno[I]. Chiaramente, in una situazione di questo tipo una macchina classica non è di alcuna utilità, perché non riesce in alcun caso a fornire una risposta nel tempo desiderato. La macchina quantistica garantisce invece la soluzione del problema in media un giorno su due.

[I] Il fatto che esistano soltanto quattro funzioni f:{0,1}→{0,1} rispecchia una semplicità puramente descrittiva che non deve trarre in inganno. Un problema molto generale esprimibile in questo formalismo è l'analisi di una scelta tra due alternative, rappresentate dai punti del dominio; f(x) descrive in tal caso la maggiore o minore utilità di ciascuna opzione rispetto all'altra. Il calcolo di f(x) può quindi richiedere la simulazione di un sistema di complessità anche notevole.

L'esempio di Deutsch è generalizzabile?

Esponiamo ora alcune considerazioni sulla generalizzabilità dell'algoritmo quantistico sopra descritto a funzioni con un dominio più esteso, che permetterebbe una interessante analisi dell'andamento dell'efficienza della soluzione in funzione delle dimensioni del problema.

Con riferimento al caso trattato nel paragrafo precedente, il calcolo quantistico di ciascuna delle funzioni

$$\begin{cases} f_1(0)=0 \\ f_1(1)=0 \end{cases} \qquad \begin{cases} f_2(0)=1 \\ f_2(1)=1 \end{cases} \qquad \begin{cases} f_3(0)=0 \\ f_3(1)=1 \end{cases} \qquad \begin{cases} f_4(0)=1 \\ f_4(1)=0 \end{cases}$$

corrisponde ad una trasformazione unitaria sullo spazio quadridimensionale $\mathcal{H}$ dei registri di ingresso e d'uscita del calcolatore. Indicata con $\{|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle\}$ una base per $\mathcal{H}$, il calcolo di f_1 richiederà ad esempio una trasformazione unitaria della forma

$$\mathbf{U}_{f_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha & 0 & \gamma \\ 0 & 0 & 1 & 0 \\ 0 & \beta & 0 & \delta \end{pmatrix}$$

con le condizioni di normalizzazione $|\alpha|^2 + |\beta|^2 = 1$ e $|\gamma|^2 + |\delta|^2 = 1$ e la condizione di ortogonalità $\alpha^*\gamma + \beta^*\delta = 0$. Ad ogni funzione f corrisponde pertanto una famiglia di matrici **U**.

Esiste tuttavia una rappresentazione più compatta e più intuitiva delle funzoni f. Consideriamo il sottoinsieme $\mathcal{H}'$ di $\mathcal{H}$ formato dalle combiazioni lineari a coefficienti reali della base $\{|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle\}$. Pur non essendo uno sottospazio di $\mathcal{H}$ (se $\mathcal{H}$ è sui complessi), $\mathcal{H}'$ è per definizione uno spazio vettoriale

sul campo dei numeri reali. Se $\mathcal{M}$ è lo spazio delle matrici 2x2 a coefficienti reali, definiamo l'isomorfismo φ di $\mathcal{M}$ in $\mathcal{H}'$ come segue:

$$\varphi(\mathbf{F}) = \frac{1}{\sqrt{2}}\left(|0\rangle \otimes \mathbf{F}|0\rangle + |1\rangle \otimes \mathbf{F}|1\rangle\right)$$

per ogni $\mathbf{F} \in \mathcal{M}$. Con $\mathbf{F}|x\rangle$ indichiamo l'azione della matrice **F**, riferita alla base $\{|0\rangle, |1\rangle\}$, come endomorfismo dello spazio delle combinazioni lineari a coefficienti reali degli stati $|0\rangle$ ed $|1\rangle$.

Posto ad esempio $\mathbf{F}_1 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$ avremo

$$\mathbf{F}_1|0\rangle \equiv \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \end{pmatrix} \equiv |0\rangle$$

e

$$\mathbf{F}_1|1\rangle \equiv \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \end{pmatrix} \equiv |0\rangle$$

e pertanto

$$\varphi(\mathbf{F}_1) = \frac{1}{\sqrt{2}}\left(|0\rangle \otimes \mathbf{F}_1|0\rangle + |1\rangle \otimes \mathbf{F}_1|1\rangle\right) = \frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|0\rangle\right) \equiv |f_1\rangle,$$

ovvero l'immagine di $\mathbf{F}_1$ attraverso φ è lo stato finale dei registri del calcolatore dopo la computazione di f_1 sulla sovrapposizione dei punti del dominio. Analogamente, possiamo associare agli stati $|f_2\rangle$, $|f_3\rangle$ ed $|f_4\rangle$ ottenuti calcolando le altre funzioni le seguenti matrici:

$$\mathbf{F}_2 = \varphi^{-1}\left(|f_2\rangle\right) = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$$

$$\mathbf{F}_3 = \varphi^{-1}\left(|f_3\rangle\right) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\mathbf{F}_4 = \varphi^{-1}\left(|f_4\rangle\right) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Poiché anche i quattro stati $|\text{UGUALI}\rangle$, $|\text{DIVERSI}\rangle$, $|\text{FALLIMENTO}\rangle$ ed $|\text{ERRORE}\rangle$ appartengono ad $\mathcal{H}'$, è possibile calcolarne la controimmagine mediante φ ottenendo così

$$\mathbf{U} = \varphi^{-1}\left(|\text{UGUALI}\rangle\right) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix},$$

$$\mathbf{D} = \varphi^{-1}\left(|\text{DIVERSI}\rangle\right) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix},$$

$$\mathbf{F} = \varphi^{-1}\left(|\text{FALLIMENTO}\rangle\right) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix},$$

ed

$$\mathbf{E} = \varphi^{-1}\left(|\text{ERRORE}\rangle\right) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}.$$

L'insieme {**U**, **D**, **F**, **E**} costituisce ovviamente una base per $\mathcal{M}$ sulla quale lo sviluppo delle matrici $\mathbf{F}_i$ è analogo a quello degli stati $|f_i\rangle$ sulla base {$|\text{UGUALI}\rangle$, $|\text{DIVERSI}\rangle$, $|\text{FALLIMENTO}\rangle$, $|\text{ERRORE}\rangle$ }:

$$\mathbf{F}_1 = \frac{1}{\sqrt{2}}(\mathbf{U} + \mathbf{F}), \qquad \mathbf{F}_2 = -\frac{1}{\sqrt{2}}(\mathbf{U} - \mathbf{F}),$$

$$\mathbf{F}_3 = \frac{1}{\sqrt{2}}(\mathbf{D} + \mathbf{F}), \qquad \mathbf{F}_4 = -\frac{1}{\sqrt{2}}(\mathbf{D} - \mathbf{F}).$$

(Come abbiamo osservato a pagina 40 per i corrispondenti vettori di stato, le quattro matrici non sono linearmente indipendenti).

E' ora necessario considerare come la condizione di ortogonalità di due stati in $\mathcal{H}'$ si traduce in un vincolo sulle corrispondenti matrici in $\mathcal{M}$. Consideriamo a tal fine le matrici $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ e $\mathbf{B} = \begin{pmatrix} e & f \\ g & h \end{pmatrix}$. La loro immagine mediante φ è rispettivamente

$$\varphi\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle\big(a|0\rangle + c|1\rangle\big) + \frac{1}{\sqrt{2}}|1\rangle\big(b|0\rangle + d|1\rangle\big) \equiv |\eta\rangle$$

e

$$\varphi\begin{pmatrix} e & f \\ g & h \end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle\big(e|0\rangle + g|1\rangle\big) + \frac{1}{\sqrt{2}}|1\rangle\big(f|0\rangle + h|1\rangle\big) \equiv |\mu\rangle.$$

Il prodotto scalare di $|\eta\rangle$ e $|\mu\rangle$ in $\mathcal{H}'$ (restrizione di quello in $\mathcal{H}$) è

$$\langle\eta|\mu\rangle = \frac{1}{2}\big(ae + cg + bf + dh\big);$$

d'altra parte, si ha anche

$$\frac{1}{2}\mathrm{Tr}\,\mathbf{A}^{\mathrm{t}}\,\mathbf{B} = ae + cg + bf + dh.$$

Definendo quindi su $\mathcal{M}$ il prodotto scalare di due matrici come $(\mathbf{A},\mathbf{B}) = \frac{1}{2}\mathrm{Tr}\,\mathbf{A}^{\mathrm{t}}\,\mathbf{B}$ si ottiene che φ trasformi basi ortonormali di $\mathcal{M}$ in basi ortonormali di $\mathcal{H}'$ (e quindi di $\mathcal{H}$). E' immediato verificare che ad esempio {**U**, **D**, **F**, **E**} è una base ortonormale rispetto a questa nozione di prodotto scalare.

A questo punto, il problema di individuare un'osservabile che permetta di estrarre, con una certa probabilità, dallo stato finale

$$\frac{1}{\sqrt{2}}\Big(|0\rangle|\,f(0)\rangle + |1\rangle|\,f(1)\rangle\Big)$$

dei registri del calcolatore l'informazione se f sia o meno costante è equivalente al problema di determinare una base ortonormale di $\mathcal{M}$ tale che nello sviluppo della matrice **F** associata a ciascuna funzione costante compaia almeno un vettore che non figura nello sviluppo delle matrici associate alle funzioni che costanti non sono. Questo è chiaramente possibile se e soltanto se ogni **F** afferente ad una funzione costante ha una proiezione non nulla sull'ortogonale del sottospazio di $\mathcal{M}$ generato dalle matrici delle funzioni non costanti.

La generalizzazione è a questo punto immediata. Ci limiteremo qui a considerare il caso in cui il dominio ed il codominio della funzione assegnata abbiano tre elementi:

$$f: \{0, 1, 2\} \rightarrow \{0, 1, 2\}.$$

Lo spazio $\mathcal{M}$ deve ora essere inteso come lo spazio delle matrici 3x3 a coefficienti reali dotato del prodotto scalare $(\mathbf{A},\mathbf{B}) = \frac{1}{3}\mathrm{Tr}\,\mathbf{A}^{t}\,\mathbf{B}$. Vi sono 27 differenti funzioni, divisibili in tre categorie:

- Tre funzioni costanti, le cui matrici sono linearmente indipendenti:

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

- Sei permutazioni, le cui matrici generano un sottospazio a dimensione cinque:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

- Diciotto funzioni non iniettive e non costanti, le cui matrici generano uno spazio a dimensione sette.

Lo stato finale dei registri del calcolatore al termine di una computazione sarà ora della forma

$$\frac{1}{\sqrt{3}}\left(|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle + |2\rangle|f(2)\rangle\right).$$

Ci domandiamo, analogamente al caso precedente, se esiste una misura che permette con una certa probabilità, a partire da tale stato, di classificare con sicurezza la funzione f in una delle tre categorie sopra elencate. La risposta è negativa. Utilizzando un pacchetto di calcolo numerico è facile stabilire che le 27 matrici associate alle funzioni f generano un sottospazio di $\mathcal{M}$ a dimensione 7.

Poiché le sole matrici corrispondenti alle funzioni non iniettive e non costanti generano un sottospazio $\mathcal{N}$ a dimensione 7, tutte le altre matrici sono ottenibili come combinazione lineare di queste ed hanno quindi una componente nulla lungo l'ortogonale di $\mathcal{N}$. Di conseguenza non è possibile distinguere con certezza, utilizzando una sola volta il calcolatore quantistico, una funzione costante o una permutazione da una funzione non costante e non iniettiva.

L'algoritmo di Shor

Presentiamo nel seguito l'algoritmo di fattorizzazione ideato da P.W. Shor, che costituisce al presente il massimo traguardo raggiunto nel campo degli algoritmi quantistici. Esso è in grado di fornire un fattore di un intero N in un tempo polinomiale nel logaritmo di N, con un enorme vantaggio sugli algoritmi classici già citati (vedi pagina 12 e bibliografia ivi indicata), che hanno complessità esponenziale.

Avvalendosi di teoremi classici della teoria dei numeri, Shor riconduce la determinazione di un fattore dell'ingresso al calcolo della periodicità di una funzione. L'esplosione esponenziale del problema diviene in tal modo superabile dal parallelismo quantistico, che rende possibile calcolare contemporaneamente la funzione in tutti i punti del dominio. La trasformata di Fourier discreta viene poi utilizzata per estrarre dalla sovrapposizione l'informazione richiesta sulla periodicità, che conduce direttamente alla determinazione di un fattore.

Prima di enunciare brevemente gli argomenti di teoria dei numeri coinvolti e di dettagliare l'algoritmo descriviamo qui l'implementazione quantistica della trasformata di Fourier discreta.

Un algoritmo quantistico per la DFT

La trasformata di Fourier discreta (DFT) costituisce, a meno di un fattore di normalizzazione e di un riordinamento degli indici, una discretizzazione diretta della trasformata di Fourier delle funzioni integrabili. La DFT q-dimensionale della funzione $f : \mathbf{N}^q \to \mathbf{C}$ è definita dalla sommatoria

$$\tilde{\mathrm{f}}(c) = \frac{1}{\sqrt{q}} \sum_{\mathrm{a}=0}^{\mathrm{q}-1} \mathrm{e}^{\frac{2\pi i a c}{q}} \mathrm{f}(a)$$

Equivalentemente, possiamo pensare alla successione f(a) come alle componenti di un vettore **v** rispetto ad una base ortogonale $\{|a\rangle\}$ di un **C**-spazio vettoriale di dimensione q. La successione $\tilde{\mathrm{f}}(c)$ risulta allora formata dalle componenti di **v** sulla base $\{|c\rangle\}$ definita da

$$|c\rangle = \frac{1}{\sqrt{q}} \sum_{\mathrm{a}=0}^{\mathrm{q}-1} \mathrm{e}^{-\frac{2\pi i a c}{q}} |a\rangle,$$

ovverosia

$$\sum_a \mathrm{f}(a)|a\rangle = \sum_c \tilde{\mathrm{f}}(c)|c\rangle$$

Essendo anche $\{|c\rangle\}$ una base ortogonale, la matrice di trasformazione fra le due basi è unitaria.

Il calcolo diretto della DFT richiede a prima vista q prodotti di numeri complessi per ogni componente di $\tilde{\mathrm{f}}$, per un totale di q^2 moltiplicazioni. Esiste però un algoritmo più efficiente, noto col nome di Fast Fourier Transform (FFT), che sfrutta la periodicità dell'esponenziale complesso per calcolare la DFT con un numero di moltiplicazioni dell'ordine di $q(\log_2 q)$ [Marple]. Il risparmio di tempo così realizzato è enorme, e le applicazioni sono tante e di tale importanza che la FFT può forse essere considerata il più importante algoritmo classico del nostro secolo.

Il più semplice degli algoritmi FFT risulta applicabile nel caso in cui il numero di componenti del vettore da trasformare sia una potenza di due. Esso si basa infatti sulla decomposizione della DFT originaria in due sommatorie Σ_1 e Σ_2 su q/2 termini che conservano la struttura di una DFT. Ciascuna di esse risulta perciò calcolabile con $(q/2)^2 = q^2/4$ prodotti, con un risparmio di q/2 operazioni. Questo procedimento di bisezione si applica ricorsivamente al calcolo di Σ_1 e Σ_2

fino a ricondursi alla trasformata (banale) di vettori ad una dimensione; le varie trasformate parziali vengono poi opportunamente ricombinate ad ottenere la DFT richiesta. Tale procedimento di bisezione da origine ai caratteristici "diagrammi a farfalla", che illustrano graficamente lo sviluppo della procedura di calcolo e costituiscono un fondamentale aiuto nello studio delle implementazioni (Figura 3). La versione quantistica della trasformata di Fourier discreta che qui descriviamo è una trasposizione immediata dell'algoritmo di Cooley e Tukey[I].

La DFT quantistica in dimensione $q=2^L$ opera sugli stati della base computazionale di registri a L bit, che possono contenere qualunque intero da zero a 2^L-1. I numeri a e c sono rappresentati dai ket $|a\rangle$ e $|c\rangle$ che possono essere espressi in forma binaria[II] come

$$|a\rangle = |a_{L-1}, a_{L-2}, \ldots a_0\rangle, \quad a = \sum_{i=0}^{L-1} a_i 2^i$$
$$|c\rangle = |c_{L-1}, c_{L-2}, \ldots c_0\rangle, \quad c = \sum_{i=0}^{L-1} c_i 2^i$$

Definiamo anche l'intero di L bit b formato dai bit di c presi in ordine inverso,

$$|b\rangle \equiv |b_{L-1}, b_{L-2}, \ldots b_0\rangle = |c_0, c_1, \ldots c_{L-1}\rangle .$$

Per costruire la DFT occorrono solo due tipi di trasformazioni unitarie. La prima è una "rotazione di spin" A_j operante sul j-mo bit di un registro:

$$\mathbf{A}_j = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix};$$

la seconda è una trasformazione $\mathbf{B}_{jk}$ che opera sui bit in posizione j e k,

[I] FFT radice 2 a decimazione in frequenza.

[II] La scrittura in base due è funzionale al procedimento di bisezione.

$$\mathbf{B}_{jk} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \mathrm{e}^{i\theta_{jk}} \end{pmatrix}$$

dove $\theta_{jk} = \pi / 2^{k-j}$. La matrice **A** è rappresentata nella base $\left\{ \left| a_j \right\rangle \right\}$, la matrice **B** nella base $\left\{ \left| a_j, a_k \right\rangle \right\}$ (a_i=0,1). $\mathbf{B}_{jk}$ ha effetto solo sugli stati che hanno un 1 sia nella posizione *j* sia nella posizione *k*, indipendentemente dallo stato degli altri bit.

Analogamente alla FFT classica, la trasformazione si svolge in L passi, indicizzati dall'indice *j* che va da L−1 a 0. Nel primo passo poniamo *j*=L−1 e applichiamo $\mathbf{A}_j$ al bit più significativo a_{L-1}. Notiamo che, in questo passaggio, abbiamo calcolato l'intero primo stadio del diagramma a farfalla.

Successivamente, nel secondo passo, riduciamo *j* di uno, *j*=L−2, ed applichiamo la seguente serie di trasformazioni (da leggere da sinistra a destra):

$$\mathbf{B}_{j,L-1}, \mathbf{A}_j .$$

Analogamente ad ogni stadio seguente riduciamo *j* di uno ed applichiamo (sempre da sinistra a destra)

$$\mathbf{B}_{j,L-1}, \mathbf{B}_{j,L-2}, \ldots, \mathbf{B}_{j,j+1}, \mathbf{A}_j$$

fino ad arrivare a *j*=0, concludendo così la procedura con $\mathbf{A}_0$. Il risultato finale degli L passi è la trasformazione

$$|a\rangle \rightarrow \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} \mathrm{e}^{\frac{2\pi i a c}{q}} |b\rangle$$

che è equivalente alla DFT in dimensione q, se i bit di $|b\rangle$ vengono letti in ordine inverso a dare $|c\rangle$. A titolo di esempio, la DFT in dimensione 16 richiede le seguenti dieci operazioni elementari:

$$\left(\mathbf{A}_3\right)\left(\mathbf{B}_{2,3}\mathbf{A}_2\right)\left(\mathbf{B}_{1,3}\mathbf{B}_{1,2}\mathbf{A}_1\right)\left(\mathbf{B}_{0,3}\mathbf{B}_{0,2}\mathbf{B}_{0,1}\mathbf{A}_0\right)$$

dove le parentesi separano i singoli stadi della computazione come evidenziati dal diagramma (Figura 4).

Nel caso generale, per trasformare il contenuto di un registro di L bit sono necessarie L operazioni $\mathbf{A}_j$ e L(L−1)/2 operazioni $\mathbf{B}_{jk}$, per un totale di L(L+1)/2 operazioni elementari. Il numero di operatori coinvolti cresce come una funzione quadratica della dimensione L dell'ingresso, cosicché l'algoritmo risulta efficiente.

Osserviamo infine che la DFT quantistica opera sulla sovrapposizione di stati di un solo registro di ingresso trasformandola in una nuova sovrapposizione i cui coefficienti sono la trasformata del vettore delle ampiezze originale.

I fondamenti teorici dell'algoritmo di Shor

L'algoritmo di Shor riconduce la fattorizzazione del numero N alla soluzione di un problema equivalente, specificamente la determinazione della periodicità della funzione

$$\mathrm{F_N}(a) = y^a \bmod \mathrm{N}$$

dove *y* è un numero naturale scelto a caso tale che N ed *y* siano primi tra loro. In questa sezione spiegheremo la relazione esistente fra la periodicità di $\mathrm{F_N}(a)$ e la fattorizzazione di N [Ekert].

Consideriamo l'equazione quadratica

$$x^2 \equiv 1 \bmod \mathrm{N}.$$

Essa ammette sempre le soluzioni $x = \pm 1 \bmod \mathrm{N}$, le cosiddette soluzioni banali. Se N è primo queste sono le sole soluzioni esistenti, in quanto $\mathbf{Z}_\mathrm{p}$ è un campo numerico, e in un campo da $x^2 - 1 \equiv (x+1)(x-1) \equiv 0 \bmod \mathrm{p}$ si può dedurre che $x - 1 = 0$

o $x+1=0$, non essendovi divisori di zero. Se N invece è composto esistono anche coppie di soluzioni non banali della forma $x=\pm a \bmod \mathrm{N}$.

Supponiamo infatti che si abbia $\mathrm{N}=\mathrm{n}_1\cdot\mathrm{n}_2$ con n_1 ed n_2 primi tra loro. Consideriamo i quattro sistemi di equazioni:

$$\begin{cases} x_1 \equiv 1 \bmod \mathrm{n}_1 \\ x_1 \equiv 1 \bmod \mathrm{n}_2 \end{cases} \qquad \begin{cases} x_2 \equiv -1 \bmod \mathrm{n}_1 \\ x_2 \equiv -1 \bmod \mathrm{n}_2 \end{cases}$$

$$\begin{cases} x_3 \equiv 1 \bmod \mathrm{n}_1 \\ x_3 \equiv -1 \bmod \mathrm{n}_2 \end{cases} \qquad \begin{cases} x_4 \equiv -1 \bmod \mathrm{n}_1 \\ x_4 \equiv 1 \bmod \mathrm{n}_2 \end{cases}$$

In ogni caso x_i^2=1 mod n_1 e mod n_2, cosicché $x_i^2 \equiv 1 \bmod \mathrm{N}$. Esiste inoltre un teorema, noto come *teorema cinese dei residui* [Schroeder], che afferma che un sistema di *s* congruenze modulo i numeri $\mathrm{n}_1, \mathrm{n}_2, \ldots \mathrm{n}_s$ primi tra loro ammette una ed una sola soluzione modulo $\mathrm{n}_1\cdot\mathrm{n}_2\cdot\ldots\mathrm{n}_s$. Questo ci assicura che ognuno dei quattro sistemi ha una ed una sola soluzione modulo N. I due sulla prima riga forniscono chiaramente le soluzioni banali x_1=1 e x_2=−1, mentre i rimanenti hanno le soluzioni non banali x_3=*a* mod N ed x_4 = −*a* mod N.

Abbiamo così che $(a+1)(a-1)\equiv 0 \bmod \mathrm{N}$, mentre $a\pm 1$ è diverso da zero. Così N divide (*a*+1)(*a*−1) senza dividere nessuno dei due fattori, che sono minori di N. Ne segue che il massimo comun divisore di N ed *a*+1 o *a*−1 è un fattore non banale di N. Il massimo comun divisore di due numeri è calcolabile efficientemente mediante l'algoritmo euclideo.

In sintesi, data una soluzione non banale *x* di $x^2 \equiv 1 \bmod \mathrm{N}$ siamo in grado di determinare efficientemente un fattore di N. La funzione $\mathrm{F_N}$ introdotta all'inizio del paragrafo è appunto utile nella determinazione di *x*.

Dato N, scegliamo a caso un *y* minore di N. Se *y* ed N sono primi fra loro, sia *r* il periodo di

$$\mathrm{F_N}(a) = y^a \bmod \mathrm{N}.$$

Questo significa che $F_N(r) = F_N(0)$, ovvero $y^r \equiv 1 \bmod N$ (r viene anche detto ordine di y modulo N). Perciò, nell'ipotesi che r sia pari, ponendo $x = y^{r/2}$ abbiamo $x^2 \equiv 1 \bmod N$, ovvero x è un candidato per una soluzione non banale. Risulta dunque evidente la connessione fra la periodicità di F_N ed il calcolo di un fattore di N.

Il procedimento suesposto può fallire se il valore di y scelto da origine ad un periodo r dispari o se r è pari ma $y^{r/2}$ è una delle soluzioni banali di $x^2 \equiv 1 \bmod N$. Si può comunque dimostrare che queste situazioni si verificano con probabilità sufficientemente piccola se r è scelto a caso. In particolare, vale il seguente teorema: se N è dispari con scomposizione in fattori

$$N = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_k^{\alpha_k},$$

y è scelto a caso e y ed N sono primi tra loro allora la probabilità che r sia pari e che $y^{r/2}$ sia una soluzione non banale di $x^2 \equiv 1 \bmod N$ è maggiore o uguale di $1 - \frac{1}{2^{k-1}}$. Il fatto che questo teorema sussista solo per N dispari non costituisce alcuna limitazione pratica, in quanto i fattori due sono facilmente identificabili e rimovibili.

Notiamo inoltre che se y è scelto a caso fra 1 ed N allora la probabilità che y ed N siano primi fra loro è maggiore di 1/log N. Così, anche nel caso in cui k=2 (le potenze di primi $N=p^{\alpha}$ sono efficientemente riconoscibili da un algoritmo classico probabilistico), assumendo di essere in grado di determinare la periodicità di F_N possiamo ottenere un fattore non banale di N, scelto un y a caso, con probabilità maggiore di 1/(2 log N).

Fattorizzazione efficiente degli interi

Shor descrive un algoritmo quantistico che fornisce l'ordine *r* di un intero scelto a caso *y*, quando *y* ed N sono primi fra loro. L'algoritmo esegue in tempo polinomiale, ovvero richiede poly(log N) passi; è inoltre probabilistico, e fornisce il valore di *r* con una probabilità di successo arbitraria fissata $1-\varepsilon$, con $\varepsilon>0$.

Sia dato un intero N. Costruiamo due registri di $q=2^L$ bit, con q compreso tra N^2 e $2N^2$. La scelta di una potenza di due per q ci permette di adoperare l'algoritmo quantistico per la FFT esposto nel paragrafo precedente, mentre i criteri che determinano la scelta dell'intervallo saranno esposti nel seguito.

Incominciamo con il primo registro nello stato $|0\rangle$ (ovvero con tutti gli L bit azzerati) e ne calcoliamo la DFT. Lo stato finale del registro sarà quindi

$$\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle.$$

Successivamente scegliamo a caso un numero naturale *y* minore di N, e calcoliamo $\mathrm{F_N}(a)=y^a \bmod \mathrm{N}$ sulla sovrapposizione degli stati del primo registro, memorizzando il risultato in un registro ausiliario. Otteniamo così lo stato correlato

$$\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle\left|y^a \bmod \mathrm{N}\right\rangle.$$

Questo calcolo può essere compiuto efficientemente ad esempio elevando ripetutamente *y* al quadrato modulo N per ottenere i valori di y^{2^i} e moltiplicando poi fra loro quelli che compaiono nell'espansione binaria di *a*.

Misuriamo successivamente i contenuti del secondo registro. Dopo la misura, esso si viene a trovare in uno stato della base computazionale, ovverosia contiene un ben preciso numero naturale. Supponiamo che il risultato sia *z* mod

N, e sia l il minimo intero positivo per cui $z = y^l \bmod \mathrm{N} = \mathrm{F_N}(l)$. Se r è l'ordine di y modulo N allora $y^l \equiv y^{jr+l}$ per ogni j. Così, in seguito alla misura, il primo registro si troverà in una sovrapposizione di valori di a pari a l, l+r, l+2r, ..., l+Ar, dove A è il massimo intero minore di $(q-l)/r$. Notiamo che il valore di $l \le r <$ N è fissato ed è determinato probabilisticamente dal processo di misura. Notiamo inoltre che *il contenuto z del registro ausiliario è assolutamente irrilevante,* e non è neppure necessario che esso sia noto. Da questo punto in poi, il registro ausiliario non viene più utilizzato. L'informazione utile è completamente codificata nel primo registro, che si trova ora, come visto, nello stato

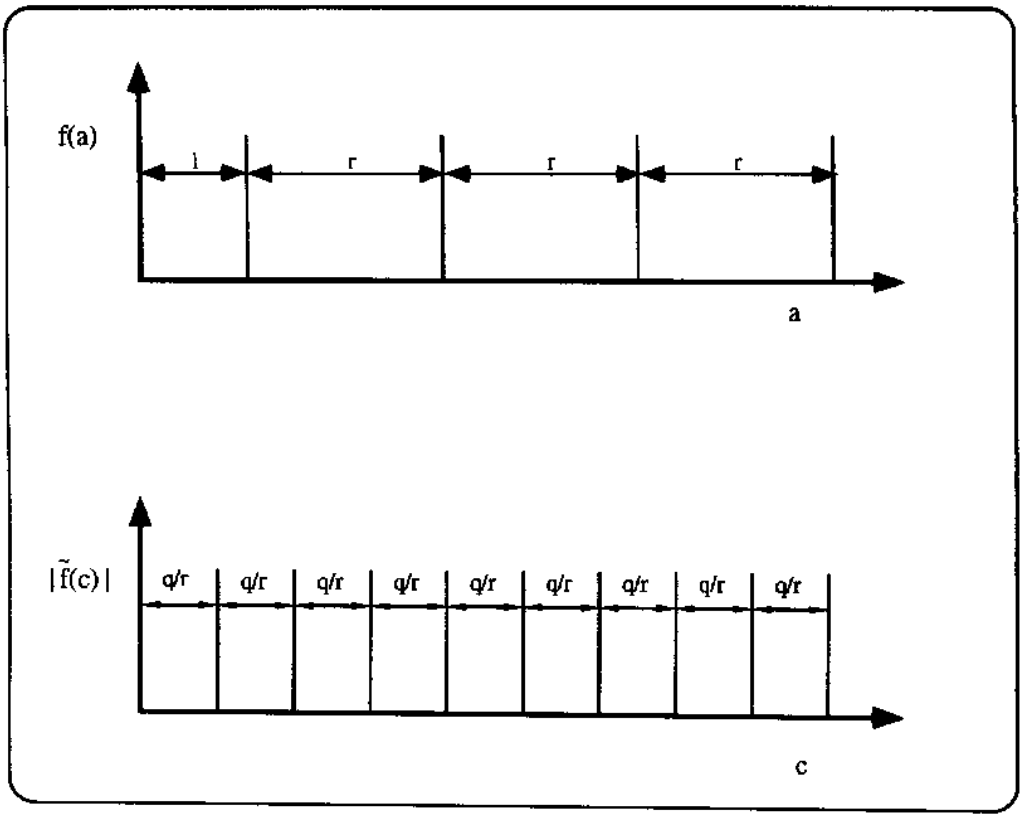
$$\left|\phi_l\right\rangle = \frac{1}{\sqrt{A+1}} \sum_{j=0}^{A} \left| jr + l \right\rangle .$$

Questo stadio della computazione è l'analogo dello stato del calcolatore a due registri dell'esempio di pagina 37 e seguenti dopo il calcolo della funzione f. Il problema che si pone è ora quello di fare interferire i diversi valori di a che compaiono nella sovrapposizione per estrarne r, indipendentemente da quale l il caso abbia determinato per noi. Occorrerà in particolare che la probabilità di determinare r non decresca troppo rapidamente con le dimensioni di N; più precisamente, vogliamo estrarre r con probabilità costante ripetendo il procedimento un numero di volte al più polinomiale nel logaritmo di N (notiamo che ad ogni ripetizione con y fissato il valore di l cambia, ma r resta costante).

Lo "schema di interferenze" richiesto per ottenere r è fornito esattamente dalla DFT. Ci limiteremo qui ad analizzare il caso più semplice, quello in cui r divide esattamente q, cosicché A=q/r−1. Lo stato finale del primo registro è in tal caso della forma

$$\left|\phi_l\right\rangle = \sqrt{\frac{r}{q}} \sum_{j=0}^{\frac{q}{r}-1} \left| jr + l \right\rangle = \sum_{a=0}^{q-1} f(a) \left| a \right\rangle ,$$

Figura 5: La funzione f(a) e il modulo della sua trasformata di Fourier $\tilde{f}(c)$.



dove $f(a)=\sum_{j=0}^{q/r-1}\sqrt{(r/q)}\,\delta^{a}_{jr+l}$. La successione infinita *f(i* mod q*)* ottenuta percorrendo ciclicamente i q valori possibili di *a* viene così ad essere esattamente periodica di periodo *r*, mentre nel caso generale il raccordo tra i due estremi della sequenza "rovina" la periodicità. Trasformando $|\phi_l\rangle$ otteniamo $|\tilde{\phi}_l\rangle=\sum_c \tilde{f}(c)|c\rangle$, dove $\tilde{f}(c)$ è la DFT di *f(a)*:

$$\tilde{f}(c)=\frac{\sqrt{r}}{q}\sum_{j=0}^{\frac{q}{r}-1}\exp\left(\frac{2\pi i(jr+l)c}{q}\right)=\frac{\sqrt{r}}{q}\left[\sum_{j=0}^{\frac{q}{r}-1}\exp\left(2\pi i\frac{jrc}{q}\right)\right]\exp\left(2\pi i\frac{lc}{q}\right).$$

Il termine in parentesi quadre nel membro di destra è diverso da zero solo se *c* è un multiplo di q/*r*, e in tal caso vale q/*r*. Abbiamo perciò

$$\tilde{f}(c)=\begin{cases}\exp(2\pi ilc/q)/\sqrt{r} & \textit{se } c \textit{ è un multiplo di } M\\ 0 & \textit{altrimenti}\end{cases}$$

ovvero la trasformata di Fourier di una sequenza di periodo r e lunghezza q ha periodo M=q/r (Figura 5). Ponendo c=jq/r otteniamo

$$\left|\tilde{\phi}_l\right\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} \exp\left(2\pi i l j / r\right) \left| j \frac{q}{r} \right\rangle$$

Una misura dell'"etichetta" c dello stato fornirà perciò un multiplo del periodo λM=λq/r, con λ casuale e uniformemente distribuito tra 0 ed r−1. Notiamo che né il risultato della misura né la sua probabilità dipendono più dallo "spiazzamento" l della sequenza $f(a)$. Questo avviene per la ben nota proprietà della trasformata di Fourier di convertire le traslazioni in moltiplicazioni per un fattore di fase[I].

In definitiva, dopo la misura abbiamo un valore di c che soddisfa la relazione c/q=λ/r. Ora c e q sono noti; se λ ed r sono primi tra loro possiamo determinare r semplicemente riducendo c/q ai minimi termini. Dal momento che λ è determinato casualmente dalla misura la probabilità che λ ed r siano primi tra loro è maggiore di 1/log r per r abbastanza grande. Così, ripetendo la misura O(log r) < O(log N) volte possiamo ottenere una probabilità di successo arbitrariamente vicina ad uno. Questo esposto è dunque un algoritmo efficiente per determinare r.

Torniamo ora a considerare il caso generale

$$\left|\phi_l\right\rangle = \frac{1}{\sqrt{A+1}} \sum_{j=0}^{A} \left| jr + l \right\rangle,$$

senza particolari ipotesi sul valore di A. In tal caso, la continuazione periodica della successione $f(a)$ delle q componenti sulla base $\{|a\rangle\}$ non presenta una ben definita periodicità r, in quanto l'inizio e il termine della successione non sono

[I] Lo scopo della DFT è unicamente quello di eliminare la dipendenza da l. In effetti, se potessimo preparare ripetutamente $\left|\phi_l\right\rangle$ con lo stesso valore di l sarebbe sufficiente misurare il valore di c su tre copie di $\left|\phi_l\right\rangle$ per ottenere r con una probabilità sufficientemente alta.

esattamente raccordabili. La situazione è esattamente trattabile sulla base dell'approccio probabilistico alla teoria dei numeri; anche in questo caso si dimostra che la probabilità di successo dell'algoritmo è amplificabile a piacere ripetendo la procedura $O(\log N)$ volte.

Una buona comprensione qualitativa delle difficoltà coinvolte e del modo di superarle è ottenibile semplicemente sulla base di argomenti standard di teoria del campionamento dei segnali. Da questo punto di vista, la successione $f(a)$ si può pensare come costituita dai punti di un segnale di periodo r campionato su una finestra temporale di lunghezza q non multipla del periodo. La finestratura causa perciò il ben noto fenomeno del *leakage*, con la comparsa di componenti spurie nello spettro del segnale. In particolare, avremo probabilità non nulla di misurare valori di c vicini ma non uguali a multipli di $M=q/r$.

Questa dispersione nelle frequenze è conseguenza di un effetto di bordo dovuto alla continuazione periodica del segnale implicita del calcolo della DFT; la sua entità diminuisce perciò con l'aumento del numero q di "punti campionati". Siccome un q elevato rende più onerosa la DFT, la scelta ottimale è il minimo valore di q che consente di ottenere r dalla misura di c con una probabilità sufficientemente elevata. Argomenti di teoria dei numeri mostrano che $q \geq N^2$ è sufficiente. Dovendo q essere una potenza di due, questo giustifica la nostra scelta iniziale di un q compreso tra N^2 e $2N^2$.