Applicazione della Stima della Fase

Stima autovalore

Dato U unitario che agisce su n qbits e $|u\rangle$ autovettore di U, $U|u\rangle = e^{2\pi i \varphi}|u\rangle$

$0 \leq \varphi < 1$ vogliamo stimare l'autovalore ossia φ.

Prima di tutto creiamo una sovrapposizione di $|u\rangle$ che permetta di eseguire la stima della fase.

la sappiamo stimare per $|\psi\rangle = \frac{1}{\sqrt{n}} \sum e^{2\pi i \varphi k} |k\rangle$

Quindi, dato $|u\rangle$, "inseriamo" φ nell'autovettore dello stato.

Per costruire la nuova sovrapposizione applichiamo U. Definiamo

$$CU\,|0\rangle|u\rangle = |0\rangle|u\rangle$$

$$CU\,|1\rangle|u\rangle = |1\rangle|Uu\rangle = e^{2\pi i\varphi}\,|1\rangle|u\rangle$$

Da cui si calcoliamo

$$(CU)\left(\frac{|0\rangle}{\sqrt{2}} + \frac{|1\rangle}{\sqrt{2}}\right)|u\rangle = \frac{|0\rangle + e^{2\pi i\varphi}\,|1\rangle}{\sqrt{2}}\,|u\rangle$$

Vogliamo creare uno stato che "assomigli" alla QFT

Costruiamo allora $|\psi_j\rangle$

nel seguente modo

$$|\psi_j\rangle|u\rangle = (CU)^{2^j}\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|u\rangle$$

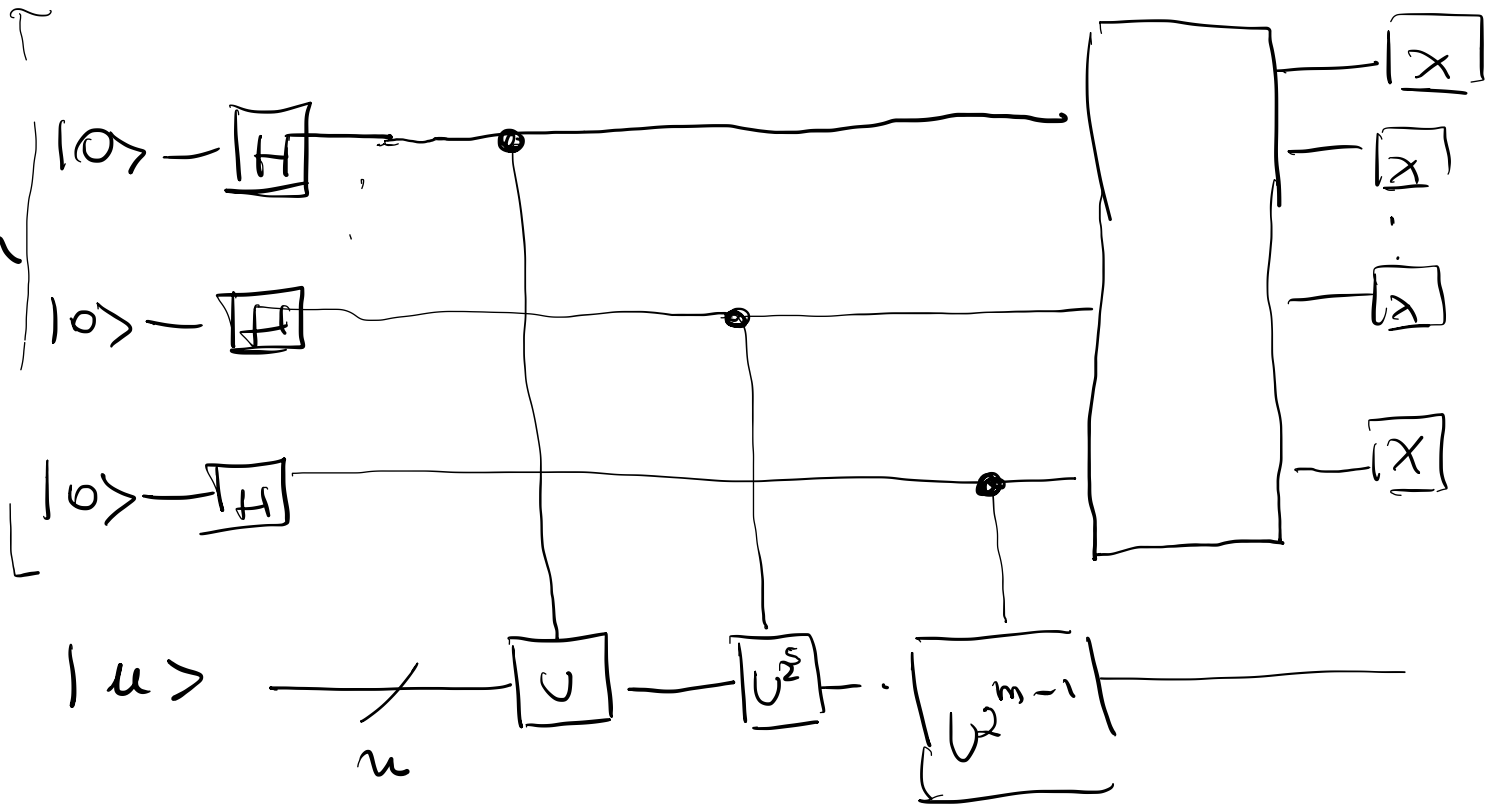
$$= \frac{|0\rangle + e^{2\pi i 2^j \varphi}|1\rangle}{\sqrt{2}}|u\rangle$$

in questo modo:

$$|\psi\rangle|u\rangle = \left(\bigotimes_{j=1}^{m}\left(\frac{|0\rangle+e^{2\pi i 2^j\varphi}|1\rangle}{\sqrt{2}}\right)\right)|u\rangle =$$

$$= \frac{1}{\sqrt{2^m}}\sum_k e^{2\pi i k \varphi_u}|k\rangle|u\rangle .$$

$\underline{m}$ dipende dal numero di cifre di accuratezza che vogliamo per φ



Due osservazioni:

1. È efficiente se costruiamo il gate U in modo efficiente

2. Non funziona avremo un autovettore! Che succede se usiamo uno stato generale $|\psi\rangle$? se Scriviamo

$|\psi\rangle = \sum \alpha_s |u_s\rangle$, $|u_s\rangle$ autovettori

dopo aver applicato $\bar{F}^{-1}$

lo stato è $|\psi\rangle = \sum \alpha_s |\tilde{\varphi}_s\rangle |u_s\rangle$

con probabilità $|\alpha_k|^2$ osserviamo una buona approssimazione di φ_k. Con $\bar{F}$ determiniamo una stima di un qualche autovalore. Se sia utile o meno dipende dalla applicazione.

Alcune applicazioni:

- Una implementazione dell'algoritmo di Grover
- Approssimare la FFT su $\mathbb{Z}/N$ per $N \neq 2^n$
- Risoluzione sistemi lineari sparsi
- Trovare l'ordine di un elemento mod N
- Calcolare il logaritmo discreto.

Per concludere Shor vediamo come trovare l'ordine $r = O(x)$ nell'elemento $x \bmod N$.

Innanzitutto $\underline{(x, N) = 1}$

Per trovare r vogliamo usare la stima degli autovalori di un operatore unitario.

Definiamo U come ($L = \lceil \log_2 N \rceil^2$)

$$U|y\rangle = |xy \ (N)\rangle \quad 0 \le y \le N$$

$$U|y\rangle = |y\rangle \quad N \le y <$$

Osservazioni 1. U è unitario

(x è invertibile mod N)

2. $U^k |y\rangle = |x^k \cdot y \ (N)\rangle$, quindi dato che $x^k \bmod(N)$ è una esponenziazione mod N può essere calcolata in # passi polinomiali in $\log(k+N)$ e calcolare U^k è relativamente "facile" (dopo)

3. $U^r = I$ quindi $\forall$ autovalore λ di U, $\lambda^r = 1$ ossia

$$\lambda = 2^{\frac{2\pi i j}{r}} \qquad 0 \le j < r$$

Abbiamo già visto che gli elementi

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i s k}{r}} |x^k(N)\rangle$$

sono autovettori di

$$U|u_s\rangle = e^{\frac{2\pi i s}{r}} |u_s\rangle$$

per $0 \leq s \leq r-1$

e che $\frac{1}{\sqrt{r}} \sum_{0}^{r-1} |u_s\rangle = |1\rangle$

Quindi se applichiamo la stima dell'autovalore a U partendo da $|1\rangle$ otteniamo

$$|1\rangle \longrightarrow \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |\tilde{\varphi}_s\rangle |u_s\rangle$$

se nella stima della fase usiamo 1 registro con

$$t = 2L + 1 + \left\lceil \log\left(2 + \frac{1}{2\varepsilon}\right) \right\rceil$$

q-bits per l'azione della F^{-1} e prepariamo il

secondo registro con $|1\rangle$

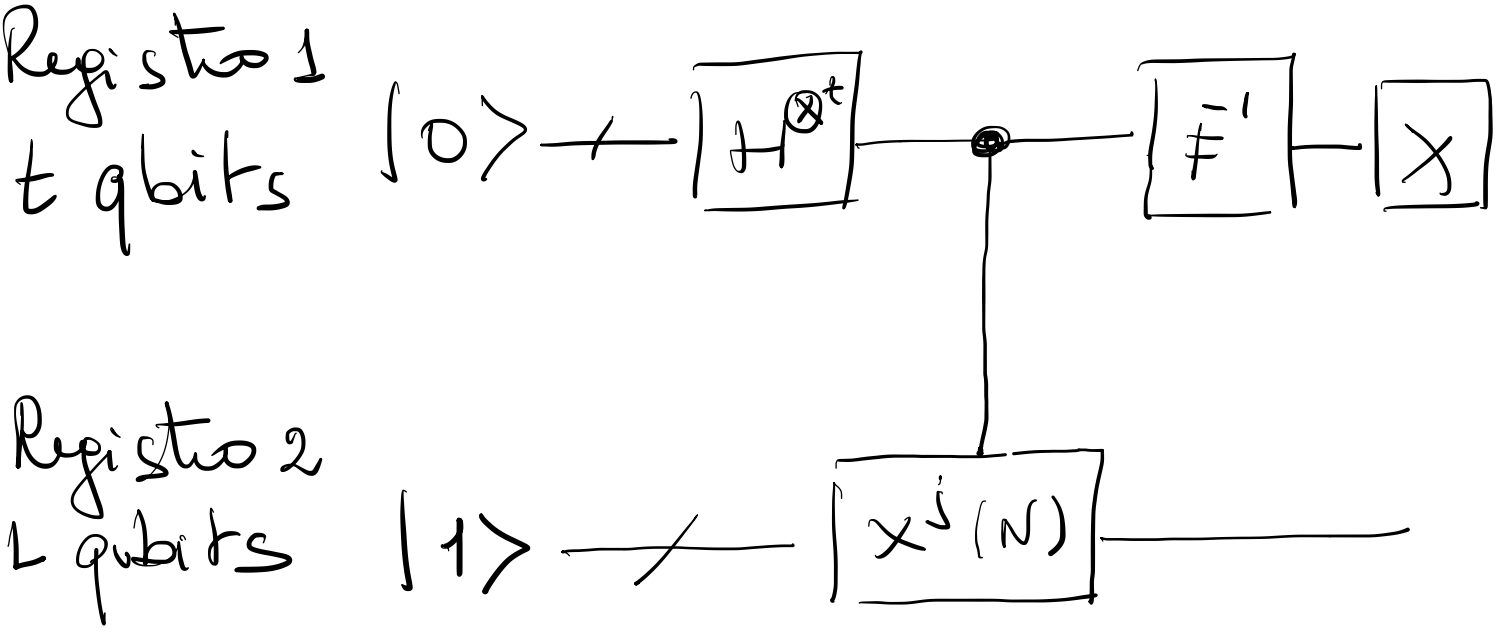
avremo che $\forall s$, $0 \leq s \leq r-1$

otteremo una stima della

fase $\varphi_s \simeq \frac{s}{r}$ accurata a

$2L+1$ bits con probabilità

almeno $\frac{1-\varepsilon}{r}$.



order finding

Per completare la procedura dobbiamo capire come ottenere r da $\varphi \simeq \frac{s}{r}$.

Conosciamo solo $2L+1$ bits di φ ma sappiamo che $\varphi \in \mathbb{Q}$

Allora possiamo calcolare la frazione più vicina a φ e trovare r.

Vediamo l'algoritmo delle frazioni continue e il seguente teorema.

th Sia $\frac{s}{r} \in \mathbb{Q}$ t.c.

$$\left| \frac{s}{r} - \varphi \right| \leq \frac{1}{2r^2}$$

Allora $\frac{s}{r}$ è un convergente della frazione continua di φ e quindi può essere calcolato in $O(L^3)$ operazioni usando l'algoritmo delle frazioni continue.

Dal momento che φ è un'approssimazione di $\frac{s}{r}$

con $2L+1$ bits e dato

che $r \leq N \leq 2^L$

$$\left| \frac{s}{r} - \varphi \right| \leq \frac{1}{2^{L+1}} \leq \frac{1}{2r^2}$$

Quindi possiamo applicare il teorema e ottenere

$$\frac{s'}{r'} = \frac{s}{r} \quad \text{con } (s', r') = 1 .$$

r' è il nostro candidato se $x^{r'} \equiv 1 \ (N)$ (ok)

Riassumendo

Algoritmo OF

Input: 1) $U_{x,N}$ black box che calcola

$$|j\rangle|k\rangle \longrightarrow |j\rangle|x^j k \ (N)\rangle$$

per $(x, N) = 1$, $\lceil \text{Log}_2 (N) \rceil = L$

2) $t = 2L + 1 + \lceil \log\left(2 + \frac{1}{2\varepsilon}\right) \rceil$ qbits inizializzati a $|0\rangle$

3) L qbits inizializzati a $|1\rangle$

output $r = \Theta(x) \mod N$.

Tempo $\Theta(L^3)$ op.

1. $|0\rangle|1\rangle$ stato iniziale

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

3. $\frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle |x^j (N)\rangle$ applica $U_{x,N}$

$$\simeq \frac{1}{\sqrt{2^t r}} \sum_{s=0}^{r-1} \sum_{j=0}^{2^t-1} e^{2\pi i s j / r} |j\rangle |u_s\rangle$$

4. $\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left| \widetilde{\frac{s}{r}} \right\rangle |u_s\rangle$ applica $\overline{F}^{-1}$ al 1° registro

5. $\widetilde{\frac{s}{r}}$ misura il 1° registro

6. (r) applica alg frazioni continue.

Algoritmo di Fattorizzazione

Input: N composto

output: p, $p \mid N$, $p \neq 1, N$

$O((\log N)^3)$ operazioni

1. $N \equiv 0 \ (2) \Rightarrow 2$
2. $N = a^b \quad a > 1, b \geq 1 \Rightarrow a$
3. Scegli $1 < x \leq N-1$

   $\gcd(x, N) = d > 1 \Rightarrow d$
4. Trova $r = O(x) \ (N)$
5. se $r \equiv 0 \ (2)$ e $x^{\frac{r}{2}} \not\equiv -1 \ (N)$

   $p = \gcd(x^{r/2} - 1, N)$

   se $p \neq 1, N$, $p \mid N \Rightarrow$ ok

   Fail.

Sono rimaste le "sospese"
un po' di cose.

1. Questo è facile trovare
   x t.c. se $r = O(x)(N)$
   $r \equiv 0 (2)$ e $x^{r/2} \not\equiv -1 (N)$

2. Descrivere $U_{x,N}$.

3. Algoritmo frazioni continue

4. Controllare o dalle
   operazioni e le
   probabilità di successo.