Quantum Fourier transform beyond Shor's algorithm

András Gilyén

Alfréd Rényi Institute of Mathematics
Budapest, Hungary

Quantum Computing Graduate Summer School, Park City Mathematics Institute, 2023 July 20

Quantum Gradient Computation Algorithm

Quantum Fourier Transform - extracting linear phase factors

Let $\varepsilon = \frac{1}{N}$ be the precision we want to achieve, and set

$$G_N := \left\{ \frac{0}{N}, \frac{1}{N}, \ldots, \frac{N-1}{N} \right\}.$$

Suppose $x, k \in G_N$ are quantum (basis) states, then

$$\sum_{x \in G_N} |x\rangle \frac{e^{2\pi i (Nxk)}}{\sqrt{N}} \xrightarrow{QFT_N} |k\rangle.$$

Quantum Gradient Computation Algorithm

Gradient computation - S. Jordan's algorithm (2004)

Input: phase oracle $O_f : |\vec{x}\rangle \to |\vec{x}\rangle e^{2\pi i f(\vec{x})}$, where $\vec{x} \in G_N^d$.
Output: gradient with (hopefully) $\varepsilon = 1/N$ coordinate-wise precision

Assumption: $f(\vec{x}) \approx f(\vec{0}) + \vec{x}\nabla f(\vec{0})$, then

$$\sum_{\vec{x}\in G_N^d} \frac{|\vec{x}\rangle}{N^{\frac{d}{2}}} \xrightarrow[N\times]{O_f} \sum_{\vec{x}\in G_N^d} |\vec{x}\rangle \frac{e^{2\pi i N f(\vec{x})}}{N^{\frac{d}{2}}} \approx e^{2\pi i N f(\vec{0})} \sum_{\vec{x}\in G_N^d} |\vec{x}\rangle \frac{e^{2\pi i \left(N\vec{x}\nabla f(\vec{0})\right)}}{N^{\frac{d}{2}}} \xrightarrow[\otimes d]{QFT_N} \approx \left|\nabla f(\vec{0})\right\rangle.$$

$$\sum_{\vec{x}\in G_N^d} \frac{|\vec{x}\rangle}{N^{\frac{d}{2}}} e^{2\pi i \left(N\vec{x}\nabla f(\vec{0})\right)} = \bigotimes_{i=1}^{d} \sum_{x_i\in G_N} \frac{|x_i\rangle}{\sqrt{N}} e^{2\pi i N x_i \nabla_i f(\vec{0})}$$

Exponential speed-up?

- If we have a circuit computing f gradient computation introduces small overheads.
- "Cheap gradient principle": $\leq 4\times$ overhead for classical gradient computation

Day 4 – Applications of &
Quantum Gradient Computation

Application to distribution estimation

How many samples do we need to estimate every probability to ε precision?

- Given a distribution $p \in \mathbb{R}_+^d$, we wish to estimate its entries p_i.
- Taking $\approx \log(1/\delta)/\varepsilon^2$ samples estimates p_1 to ε precision with success probability at least $1 - \delta$ (by the Chernoff bound).
- Taking $\approx \log(d)/\varepsilon^2$ samples, their histogram ε-approximates every p_i with high probability (by the union bound).

Can we improve this using amplitude estimation?

- Assume we can sample using the quantum computer:

$$V : \left|\bar{0}\right\rangle \to \sum_{i=1}^{d} \sqrt{p_i}|i\rangle|\psi_i\rangle$$

- Weakest natural assumption. E.g., implement your Monte Carlo sampler on a quant. comp. ($|\psi_i\rangle$ is arbitrary garbage, e.g., describing the state of the Monte Carlo sampler.)
- Can estimate p_1 to ε precision with $\approx 1/\varepsilon$ steps of amplitude estimation. But all of them?

Idea: build a probability oracle for a linear function $f(\vec{x}) = \langle \vec{x} | p \rangle$!

Modifying the oracle to get probability oracle for f with $\nabla f = p$.

- Apply rotation controlled by $|i\rangle|x_i\rangle$ to "rejection sample"

$$R(x_i) = \begin{pmatrix} \sqrt{x_i} & -\sqrt{1-x_i} \\ \sqrt{1-x_i} & \sqrt{x_i} \end{pmatrix}$$

- This gives a "probability oracle" for every $\vec{x} \in [0,1]^d$:

$$U\colon |\bar{0}\rangle|\vec{x}\rangle \to \sum_{i=1}^{d} \left(\sqrt{p_i x_i}|0\rangle|i\rangle|\psi_i\rangle + \sqrt{p_i(1-x_i)}|1\rangle|i\rangle|\psi_i\rangle \right)|\vec{x}\rangle$$

- If the second register is in state $\vec{x} \in [0,1]^d$, then

$$\Pr(\text{first qubit is in state } 0) = \sum_{i=1}^{d} x_i p_i = \langle \vec{x} | p \rangle$$

- This is a probability oracle for the linear function $f(\vec{x}) := \langle \vec{x} | p \rangle$.

Probability oracle to phase oracle

Modifying the oracle

- Given a probability oracle for the function $f(\vec{x}) \in [0,1]$ (currently $f(\vec{x}) = \langle \vec{x} | p \rangle$)

$$U_f : \left|\bar{0}\right\rangle \left|\vec{x}\right\rangle \to \left(\sqrt{f(\vec{x})} |0\rangle \left|\psi_{\text{accept}}(\vec{x})\right\rangle + \sqrt{1 - f(\vec{x})} |1\rangle \left|\psi_{\text{reject}}(\vec{x})\right\rangle \right) \left|\vec{x}\right\rangle$$

- We wish to implement a phase oracle

$$O_f : \left|\vec{x}\right\rangle \to e^{if(\vec{x})} \left|\vec{x}\right\rangle$$

- First we create a block encoding $W := (I \otimes U_f^\dagger)(SWAP \otimes I)(I \otimes U_f)$

$$\operatorname{diag}(f(\vec{x})) = (\langle 0 | \left\langle \bar{0} \right| \otimes I) W (|0\rangle \left|\bar{0}\right\rangle \otimes I)$$

- We can think about $\operatorname{diag}(f(\vec{x}))$ as a Hamiltonian, and use Hamiltonian simulation.
- Use quantum signal processing to implement $\left|\vec{x}\right\rangle \to e^{iN \cdot f(\vec{x})} \left|\vec{x}\right\rangle$ with complexity $\widetilde{O}(N)$!

Distribution estimation (Apeldoorn 2020)

- Use Jordan's gradient computation algorithm for estimating p with $\widetilde{O}(1/\varepsilon)$ queries to V.

Application to purified mixed state tomography

Input model and problem statement

- Suppose we are given purified state preparation circuit
$$V: |\bar{0}\rangle \to |\psi\rangle_{AB}$$
such that $\mathrm{Tr}(|\psi\rangle\langle\psi|)_A = \rho$.
- We wish to estimate ρ to precision ε in trace distance

Idea: consider the linear function $X \to \mathrm{Tr}(X\rho)$

- Suppose the matrix elements of X are uniformly random $(-1, 1)$
- Worst case $\|X\| = d$ (all ones matrix)
- Apart from exponentially small probability: $\|X\| = \sqrt{d}$ (matrix Chernoff bound)
- We can build block-encoding of $\mathrm{diag}(\mathrm{Tr}(X\rho)/\sqrt{d}) = \mathrm{diag}(\langle X|\rho\rangle_{HS}/\sqrt{d})$ for most X.
- With $\sqrt{d}/\varepsilon$ uses of V we get ε coordinate-wise (almost) independent estimates of ρ
- If the estimator is unbiased we very likely get $\varepsilon\sqrt{d}$ estimate in $\|\cdot\|$
- Implies $\varepsilon r\sqrt{d}$ estimate of ρ in trace norm ($r = \mathrm{rank}(\rho)$) $\quad\Rightarrow\quad \widetilde{O}(dr/\varepsilon)$ complexity!

Bounding non-linear phase errors for non-linear functions

Want: $\left|\vec{x}\right\rangle \to \left|\vec{x}\right\rangle e^{\frac{2\pi i}{\varepsilon}\vec{x}\nabla f(0)}$ for $\vec{x} \in [0,1]^d$. Have $O_f : \left|\vec{x}\right\rangle \to \left|\vec{x}\right\rangle e^{2\pi i f(\vec{x})}$.

Rescaling the function

Suppose $f(x) = \sum_{j=0}^{\infty} b_j x^j$, then

$$R \cdot f(x/R) = \sum_{j=0}^{\infty} b_j R^{1-j} x^j.$$

Note that 1 phase query to the rescaled function costs R original queries!

Trick: Using higher order numerical differential formulas

$$xf'(0) = \frac{f(x) - f(-x)}{2} + \mathcal{O}\left(x^3\right)$$

$$\vec{x}\nabla f(0) = \sum_{k=-m}^{m} a_k f(k\vec{x}) + \mathcal{O}\left(\left\|\vec{x}\right\|^{(2m+1)}\right)$$

We need $\left\|\vec{x}\right\| < 1$! We set $R \approx \sqrt{d}$, $m \approx \log(d/\varepsilon)$

Optimal query complexity of smoothed

c-smooth functions (cf. Gevrey-class $\sigma = 1/2$)

We say that an analytic function f is c-smooth if all k-fold partial derivatives are bounded by $c^k \cdot \sqrt{k!}$ in absolute value for all $k \in \mathbb{N}$.

Query complexity for c-smooth functions

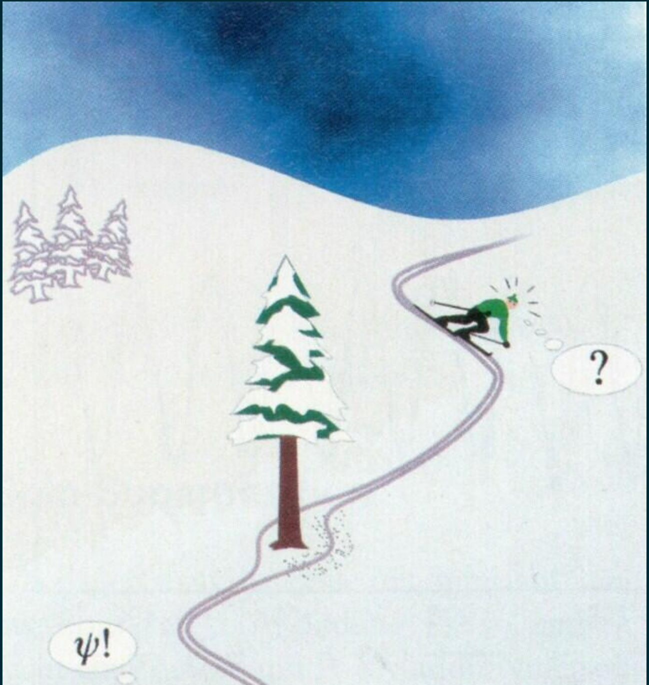
The quantum query complexity of calculating an ε-$\|.\|_\infty$-apx. gradient is

$$\widetilde{\Theta}\left(\frac{c\sqrt{d}}{\varepsilon}\right).$$

Query complexity of calculating an ε-approximate gradient in $\|\cdot\|_\infty$

Classical	Coord.-wise	**Smoothed**	**Degree-k**
$\widetilde{O}\left(\frac{d}{\varepsilon^2}\right)$	$\widetilde{O}\left(\frac{d}{\varepsilon}\right)$	$\widetilde{O}\left(\frac{\sqrt{d}}{\varepsilon}\right)$	$\widetilde{O}\left(\frac{k}{\varepsilon}\right)$

Faster quantum gradient descent!

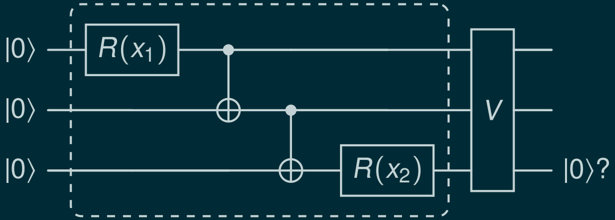


A generic model of quantum optimization algorithms

Quantum circuits are powerful → use them for optimization

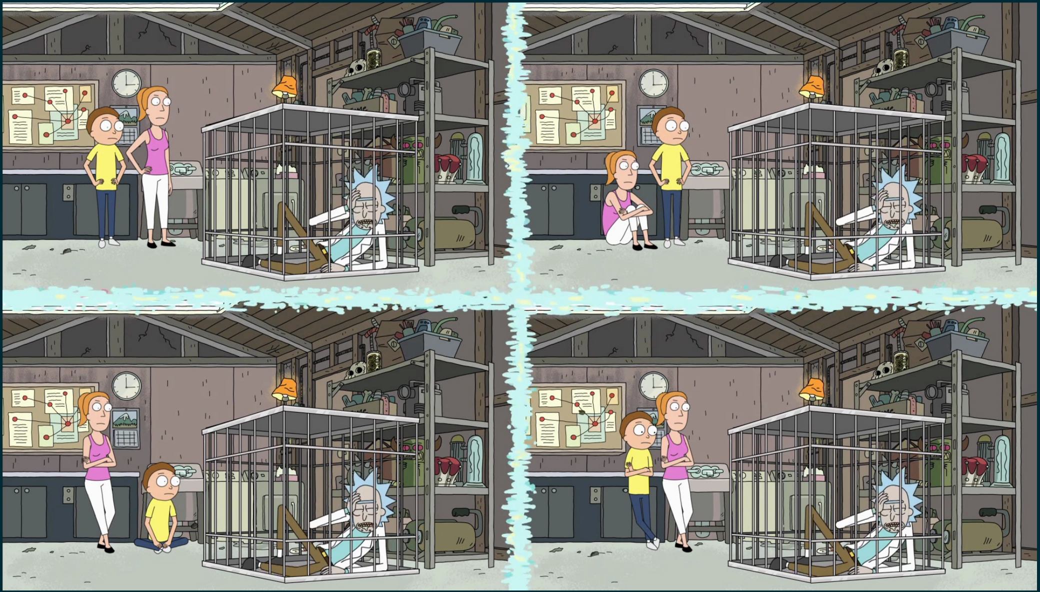
Tuning an inherently quantum model

- Quantum variational eigensolver – for finding a ground state
- Quantum approximate optimization algorithm
- Quantum machine learning, etc.

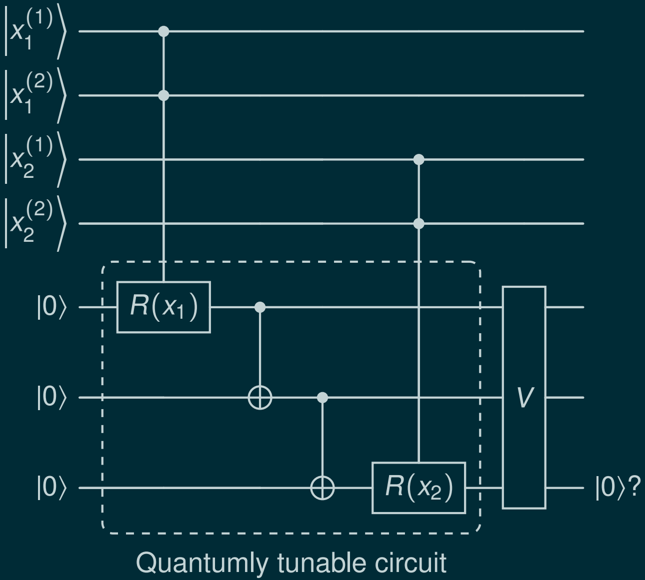


Tunable circuit - Find: $\arg\min_{\vec{x}}(p(\vec{x}))$

Quantum trick: tuning parameters in superposition!



The quantumly tunable version of the circuit



Abstract model of the optimization circuits

What we have

A probability oracle

$$U_p : \left|\vec{x}\right\rangle|0\rangle \to \left|\vec{x}\right\rangle\left(\sqrt{p(\vec{x})}|\psi_0\rangle|0\rangle + \sqrt{1 - p(\vec{x})}|\psi_1\rangle|1\rangle\right).$$

Filling the gap – proving smoothness of $p(\vec{x})$

If each tunable gate in the quantum optimization circuit can be written as

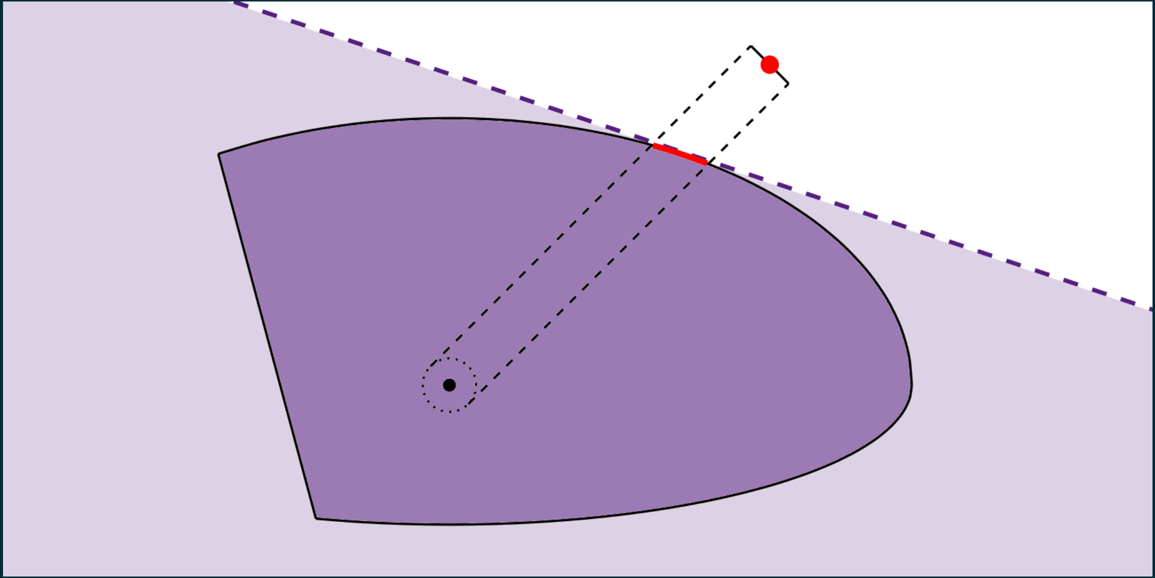
$$e^{ix_jH_j}, \text{ where } \left\|H_j\right\| \leq 1,$$

then $p(\vec{x})$ is 2-smooth.

Convert it to phase oracle and use Jordan's algorithm

The smoothed version of Jordan's algorithm computes the gradient in time $\mathcal{O}\left(\sqrt{d}/\varepsilon\right)$.

Application to a classical problem: black-box convex optimization



A separating hyperplane can be found by making $\widetilde{O}(1)$ membership queries in superposition.
Classically d queries are necessary (can be seen by information theoretic lower bound)