Quantum rejection sampling

Maris Ozols
University of Waterloo

Martin Rötteler
NEC Laboratories America

Jérémie Roland
Université Libre de Bruxelles

arXiv:1103.2774

Motivation

We started with. . . [recall Martin's talk yesterday]

An algorithm for the Boolean hidden shift problem:

- Might be useful for breaking cryptosystems (LFSRs)
- Potential insights into the dihedral hidden subgroup problem

Motivation

We started with... [recall Martin's talk yesterday]

An algorithm for the Boolean hidden shift problem:

- Might be useful for breaking cryptosystems (LFSRs)
- Potential insights into the dihedral hidden subgroup problem

...but ended up with

A useful primitive for constructing quantum algorithms:

- Quantum algorithm for linear systems of equations [HHL09]
- Quantum Metropolis algorithm [TOVPV11]
- Preparing PEPS [STV11]
- more...

Resampling

Classical $\boldsymbol{p} \to \boldsymbol{s}$ resampling problem

- **Given:** $\boldsymbol{p}, \boldsymbol{s} \in \mathbb{R}_+^n$ with $\|\boldsymbol{p}\|_1 = \|\boldsymbol{s}\|_1 = 1$
  Ability to sample from distribution $\boldsymbol{p}$
- **Task:** Sample from distribution $\boldsymbol{s}$

Resampling

Classical $\boldsymbol{p} \to \boldsymbol{s}$ resampling problem

- **Given:** $\boldsymbol{p}, \boldsymbol{s} \in \mathbb{R}_+^n$ with $\|\boldsymbol{p}\|_1 = \|\boldsymbol{s}\|_1 = 1$
  Ability to sample from distribution $\boldsymbol{p}$
- **Task:** Sample from distribution $\boldsymbol{s}$
- **Question:** How many samples from $\boldsymbol{p}$ we need to prepare one sample from $\boldsymbol{s}$?

Resampling

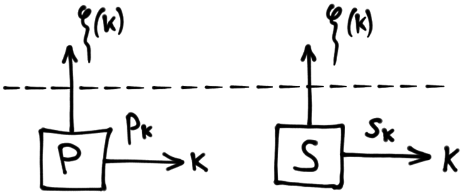
Classical $\boldsymbol{p} \to \boldsymbol{s}$ resampling problem

- **Given:** $\boldsymbol{p}, \boldsymbol{s} \in \mathbb{R}_+^n$ with $\|\boldsymbol{p}\|_1 = \|\boldsymbol{s}\|_1 = 1$
  Ability to sample from distribution $\boldsymbol{p}$
- **Task:** Sample from distribution $\boldsymbol{s}$
- **Question:** How many samples from $\boldsymbol{p}$ we need to prepare one sample from $\boldsymbol{s}$?

Resampling

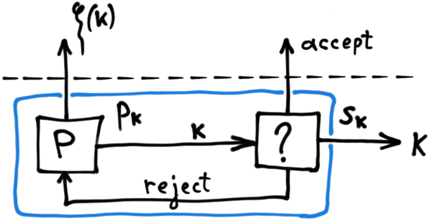
Classical $\boldsymbol{p} \to \boldsymbol{s}$ resampling problem

- **Given:** $\boldsymbol{p}, \boldsymbol{s} \in \mathbb{R}_+^n$ with $\|\boldsymbol{p}\|_1 = \|\boldsymbol{s}\|_1 = 1$
  Ability to sample from distribution $\boldsymbol{p}$
- **Task:** Sample from distribution $\boldsymbol{s}$
- **Question:** How many samples from $\boldsymbol{p}$ we need to prepare one sample from $\boldsymbol{s}$?
- **Note:** Samples are pairs $(k, \xi(k))$ where $\xi(k)$ is not accessible



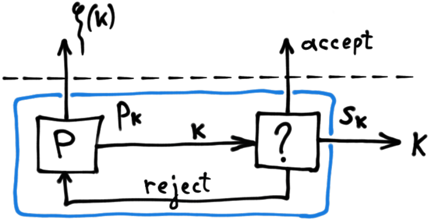
Classical rejection sampling

Algorithm



Classical rejection sampling

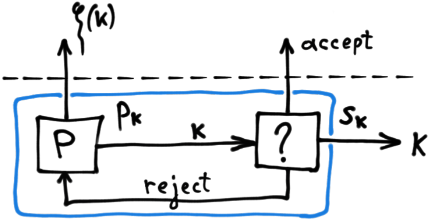
Algorithm



- Accept k with probability $\gamma s_k / p_k$

Classical rejection sampling

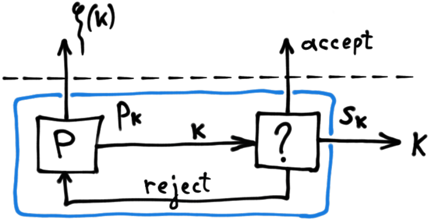
Algorithm

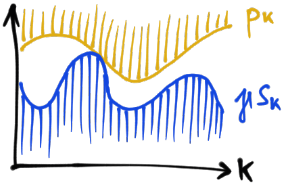


- Accept k with probability $\gamma s_k / p_k$
- Avg. prob. to accept: $\sum_k p_k \cdot \gamma s_k / p_k = \gamma$

Classical rejection sampling

Algorithm

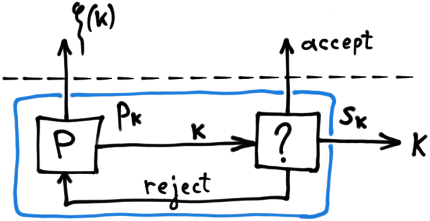


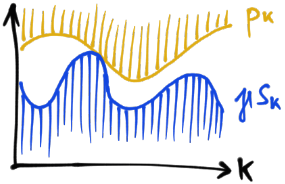


- Accept k with probability $\gamma s_k / p_k \ \leq 1$
- Avg. prob. to accept: $\sum_k p_k \cdot \gamma s_k / p_k = \gamma$

Classical rejection sampling

Algorithm

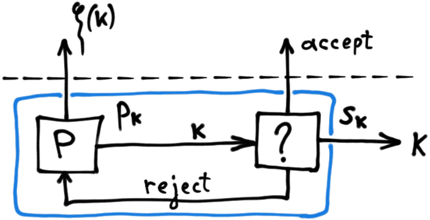


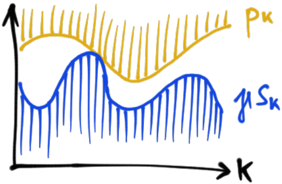


- Accept k with probability $\gamma s_k / p_k \ \leq 1$, so $\gamma = \min_k p_k / s_k$
- Avg. prob. to accept: $\sum_k p_k \cdot \gamma s_k / p_k = \gamma$

Classical rejection sampling

Algorithm

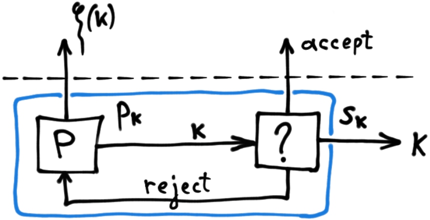


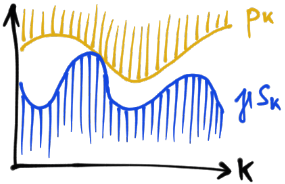


- Accept k with probability $\gamma s_k / p_k \le 1$, so $\gamma = \min_k p_k / s_k$
- Avg. prob. to accept: $\sum_k p_k \cdot \gamma s_k / p_k = \gamma$
- Query complexity: $\Theta(1/\gamma)$

Classical rejection sampling

Algorithm

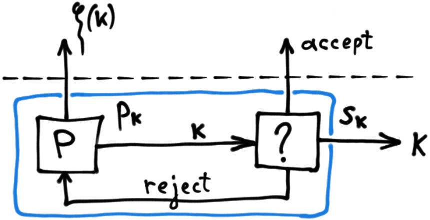


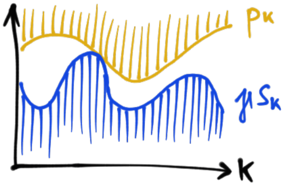


- Accept k with probability $\gamma s_k / p_k \; \leq 1$, so $\gamma = \min_k p_k / s_k$
- Avg. prob. to accept: $\sum_k p_k \cdot \gamma s_k / p_k = \gamma$
- Query complexity: $\Theta(1/\gamma)$
- Introduced by von Neumann in 1951

Classical rejection sampling

Algorithm





- Accept k with probability $\gamma s_k / p_k \ \leq 1$, so $\gamma = \min_k p_k / s_k$
- Avg. prob. to accept: $\sum_k p_k \cdot \gamma s_k / p_k = \gamma$
- Query complexity: $\Theta(1/\gamma)$
- Introduced by von Neumann in 1951
- Has numerous applications:
  - Metropolis algorithm [MRRTT53]
  - Monte-Carlo simulations
  - optimization (simulated annealing), etc.

Quantum resampling

Quantum $\boldsymbol{\pi} \to \boldsymbol{\sigma}$ resampling problem

- **Given:** $\boldsymbol{\pi}, \boldsymbol{\sigma} \in \mathbb{R}^n_+$ with $\|\boldsymbol{\pi}\|_2 = \|\boldsymbol{\sigma}\|_2 = 1$
  Oracle for preparing $|\pi\rangle = \sum_{k=1}^n \pi_k |k\rangle|\xi(k)\rangle$

Quantum resampling

Quantum $\boldsymbol{\pi} \to \boldsymbol{\sigma}$ resampling problem

- **Given:** $\boldsymbol{\pi}, \boldsymbol{\sigma} \in \mathbb{R}_+^n$ with $\|\boldsymbol{\pi}\|_2 = \|\boldsymbol{\sigma}\|_2 = 1$
  Oracle for preparing $|\pi\rangle = \sum_{k=1}^n \pi_k |k\rangle|\xi(k)\rangle$
- **Task:** Prepare $|\sigma\rangle = \sum_{k=1}^n \sigma_k |k\rangle|\xi(k)\rangle$

Quantum resampling

Quantum $\boldsymbol{\pi} \to \boldsymbol{\sigma}$ resampling problem

- **Given:** $\boldsymbol{\pi}, \boldsymbol{\sigma} \in \mathbb{R}_+^n$ with $\|\boldsymbol{\pi}\|_2 = \|\boldsymbol{\sigma}\|_2 = 1$
  Oracle for preparing $|\pi\rangle = \sum_{k=1}^n \pi_k |k\rangle|\xi(k)\rangle$
- **Task:** Prepare $|\sigma\rangle = \sum_{k=1}^n \sigma_k |k\rangle|\xi(k)\rangle$
- **Question:** How many $|\pi\rangle$s we need to produce one $|\sigma\rangle$?

Quantum resampling

Quantum $\boldsymbol{\pi} \to \boldsymbol{\sigma}$ resampling problem

- **Given:** $\boldsymbol{\pi}, \boldsymbol{\sigma} \in \mathbb{R}_+^n$ with $\|\boldsymbol{\pi}\|_2 = \|\boldsymbol{\sigma}\|_2 = 1$
  Oracle for preparing $|\pi\rangle = \sum_{k=1}^n \pi_k |k\rangle|\xi(k)\rangle$
- **Task:** Prepare $|\sigma\rangle = \sum_{k=1}^n \sigma_k |k\rangle|\xi(k)\rangle$
- **Question:** How many $|\pi\rangle$s we need to produce one $|\sigma\rangle$?
- **Note:** States $|\xi(k)\rangle$ are not known

Quantum resampling

Quantum $\boldsymbol{\pi} \to \boldsymbol{\sigma}$ resampling problem

- **Given:** $\boldsymbol{\pi}, \boldsymbol{\sigma} \in \mathbb{R}_+^n$ with $\|\boldsymbol{\pi}\|_2 = \|\boldsymbol{\sigma}\|_2 = 1$
  Oracle for preparing $|\pi\rangle = \sum_{k=1}^n \pi_k |k\rangle |\xi(k)\rangle$
- **Task:** Prepare $|\sigma\rangle = \sum_{k=1}^n \sigma_k |k\rangle |\xi(k)\rangle$
- **Question:** How many $|\pi\rangle$s we need to produce one $|\sigma\rangle$?
- **Note:** States $|\xi(k)\rangle$ are not known

Main theorem (exact case)

The quantum query complexity of the exact $\boldsymbol{\pi} \to \boldsymbol{\sigma}$ quantum resampling problem is $\Theta(1/\gamma)$ where $\gamma = \min_k |\pi_k / \sigma_k|$

Quantum resampling

Quantum $\boldsymbol{\pi} \to \boldsymbol{\sigma}$ resampling problem

- **Given:** $\boldsymbol{\pi}, \boldsymbol{\sigma} \in \mathbb{R}^n_+$ with $\|\boldsymbol{\pi}\|_2 = \|\boldsymbol{\sigma}\|_2 = 1$
  Oracle for preparing $|\pi\rangle = \sum_{k=1}^n \pi_k |k\rangle|\xi(k)\rangle$
- **Task:** Prepare $|\sigma\rangle = \sum_{k=1}^n \sigma_k |k\rangle|\xi(k)\rangle$
- **Question:** How many $|\pi\rangle$s we need to produce one $|\sigma\rangle$?
- **Note:** States $|\xi(k)\rangle$ are not known

Main theorem (exact case)

The quantum query complexity of the exact $\boldsymbol{\pi} \to \boldsymbol{\sigma}$ quantum resampling problem is $\Theta(1/\gamma)$ where $\gamma = \min_k |\pi_k/\sigma_k|$

Approximate preparation

Task: Prepare $\sqrt{1-\varepsilon}|\sigma\rangle + \sqrt{\varepsilon}|\mathrm{error}\rangle$

Quantum resampling

Quantum $\boldsymbol{\pi} \to \boldsymbol{\sigma}$ resampling problem

- **Given:** $\boldsymbol{\pi}, \boldsymbol{\sigma} \in \mathbb{R}_+^n$ with $\|\boldsymbol{\pi}\|_2 = \|\boldsymbol{\sigma}\|_2 = 1$
  Oracle for preparing $|\pi\rangle = \sum_{k=1}^n \pi_k |k\rangle|\xi(k)\rangle$
- **Task:** Prepare $|\sigma\rangle = \sum_{k=1}^n \sigma_k |k\rangle|\xi(k)\rangle$
- **Question:** How many $|\pi\rangle$s we need to produce one $|\sigma\rangle$?
- **Note:** States $|\xi(k)\rangle$ are not known

Main theorem (exact case)

The quantum query complexity of the exact $\boldsymbol{\pi} \to \boldsymbol{\sigma}$ quantum resampling problem is $\Theta(1/\gamma)$ where $\gamma = \min_k |\pi_k/\sigma_k|$

Approximate preparation

Task: Prepare $\sqrt{1-\varepsilon}|\sigma\rangle + \sqrt{\varepsilon}|\mathrm{error}\rangle$
$\iff$ Prepare $|\delta\rangle$ with $\boldsymbol{\sigma} \cdot \boldsymbol{\delta} \geq \sqrt{1-\varepsilon}$

Quantum rejection sampling algorithm

1. Use the oracle to prepare
$$|0\rangle|\pi\rangle = |0\rangle \sum_{k=1}^{n} \pi_k |k\rangle|\xi(k)\rangle$$

Quantum rejection sampling algorithm

1. Use the oracle to prepare
$$|0\rangle|\pi\rangle = |0\rangle \sum_{k=1}^{n} \pi_k |k\rangle|\xi(k)\rangle$$

2. Pick some $\boldsymbol{\delta} \in \mathbb{R}_+^n$ and rotate the state in the first register:
$$\sum_{k=1}^{n} \big(\sqrt{|\pi_k|^2 - |\delta_k|^2}\, |0\rangle + \delta_k\, |1\rangle\big)|k\rangle|\xi(k)\rangle$$

Quantum rejection sampling algorithm

1. Use the oracle to prepare
$$|0\rangle|\pi\rangle = |0\rangle \sum_{k=1}^{n} \pi_k |k\rangle|\xi(k)\rangle$$
2. Pick some $\boldsymbol{\delta} \in \mathbb{R}_+^n$ and rotate the state in the first register:
$$\sum_{k=1}^{n} \big(\sqrt{|\pi_k|^2 - |\delta_k|^2}\,|0\rangle + \delta_k\,|1\rangle\big)|k\rangle|\xi(k)\rangle$$
3. Measure the first register:

Quantum rejection sampling algorithm

1. Use the oracle to prepare
$$|0\rangle|\pi\rangle = |0\rangle \sum_{k=1}^{n} \pi_k |k\rangle|\xi(k)\rangle$$

2. Pick some $\boldsymbol{\delta} \in \mathbb{R}_+^n$ and rotate the state in the first register:
$$\sum_{k=1}^{n} \Big(\sqrt{|\pi_k|^2 - |\delta_k|^2}\, |0\rangle + \delta_k \, |1\rangle \Big) |k\rangle|\xi(k)\rangle$$

3. Measure the first register:
   - w.p. $\|\boldsymbol{\delta}\|_2^2$ the state collapses to
   $$\sum_{k=1}^{n} \hat{\delta}_k |k\rangle|\xi(k)\rangle$$
   where $\hat{\delta}_k = \delta_k / \|\boldsymbol{\delta}\|_2$

Quantum rejection sampling algorithm

Subroutine

$$\text{one copy of } |\pi\rangle \quad \mapsto \quad \sum_{k=1}^{n} \hat{\delta}_k |k\rangle |\xi(k)\rangle \quad \text{w.p.} \quad \|\boldsymbol{\delta}\|_2^2$$

Quantum rejection sampling algorithm

Subroutine

$$\text{one copy of } |\pi\rangle \quad \mapsto \quad \sum_{k=1}^{n} \hat{\delta}_k |k\rangle|\xi(k)\rangle \quad \text{w.p.} \quad \|\boldsymbol{\delta}\|_2^2$$

Amplification

- **Naïve:** repeat $1/\|\boldsymbol{\delta}\|_2^2$ times to succeed w.p. ≈ 1

Quantum rejection sampling algorithm

Subroutine

$$\text{one copy of } |\pi\rangle \quad \mapsto \quad \sum_{k=1}^{n} \hat{\delta}_k |k\rangle |\xi(k)\rangle \quad \text{w.p.} \quad \|\boldsymbol{\delta}\|_2^2$$

Amplification

- **Naïve:** repeat $1/\|\boldsymbol{\delta}\|_2^2$ times to succeed w.p. ≈ 1
- **Quantum:** $1/\|\boldsymbol{\delta}\|_2$ repetitions of amplitude amplification suffice [BHMT00]

Quantum rejection sampling algorithm

Subroutine

$$\text{one copy of } |\pi\rangle \quad \mapsto \quad \sum_{k=1}^{n} \hat{\delta}_k |k\rangle |\xi(k)\rangle \quad \text{w.p.} \quad \|\boldsymbol{\delta}\|_2^2$$

Amplification

- **Naïve:** repeat $1/\|\boldsymbol{\delta}\|_2^2$ times to succeed w.p. ≈ 1
- **Quantum:** $1/\|\boldsymbol{\delta}\|_2$ repetitions of amplitude amplification suffice [BHMT00]

Summary

We can prepare $\sum_{k=1}^{n} \hat{\delta}_k |k\rangle |\xi(k)\rangle$ with $O(1/\|\boldsymbol{\delta}\|_2)$ quantum queries

Quantum rejection sampling algorithm

Subroutine

$$\text{one copy of } |\pi\rangle \quad \mapsto \quad \sum_{k=1}^{n} \hat{\delta}_k |k\rangle |\xi(k)\rangle \quad \text{w.p.} \quad \|\boldsymbol{\delta}\|_2^2$$

Amplification

- **Naïve:** repeat $1/\|\boldsymbol{\delta}\|_2^2$ times to succeed w.p. ≈ 1
- **Quantum:** $1/\|\boldsymbol{\delta}\|_2$ repetitions of amplitude amplification suffice [BHMT00]

Summary

We can prepare $\sum_{k=1}^{n} \hat{\delta}_k |k\rangle |\xi(k)\rangle$ with $O(1/\|\boldsymbol{\delta}\|_2)$ quantum queries

Goal: preparing $|\sigma\rangle$

- What $\boldsymbol{\delta}$ should we choose?
- We are done if $\boldsymbol{\sigma} \cdot \hat{\boldsymbol{\delta}} \geq \sqrt{1 - \varepsilon}$ where $\hat{\boldsymbol{\delta}} = \boldsymbol{\delta}/\|\boldsymbol{\delta}\|_2$

Optimization

Problem

- $\min_{\boldsymbol{\delta}} 1/\|\boldsymbol{\delta}\|_2$ s.t. $\boldsymbol{\sigma} \cdot \hat{\boldsymbol{\delta}} \geq \sqrt{1-\varepsilon}$

Optimization

$$\sum_{k=1}^{n} \left(\sqrt{|\pi_k|^2 - |\delta_k|^2}\, |0\rangle + \delta_k\, |1\rangle \right) |k\rangle |\xi(k)\rangle$$

Problem

- $\min_{\boldsymbol{\delta}} 1/\|\boldsymbol{\delta}\|_2$ s.t. $\boldsymbol{\sigma} \cdot \hat{\boldsymbol{\delta}} \geq \sqrt{1-\varepsilon}$ and $0 \leq \delta_k \leq \pi_k$

Optimization

$$\sum_{k=1}^{n} \left(\sqrt{|\pi_k|^2 - |\delta_k|^2}\,|0\rangle + \delta_k\,|1\rangle\right)|k\rangle|\xi(k)\rangle$$

Problem

- $\min_{\boldsymbol{\delta}} 1/\|\boldsymbol{\delta}\|_2$ s.t. $\boldsymbol{\sigma} \cdot \hat{\boldsymbol{\delta}} \geq \sqrt{1-\varepsilon}$ and $0 \leq \delta_k \leq \pi_k$
- This can be stated as an SDP

Optimization

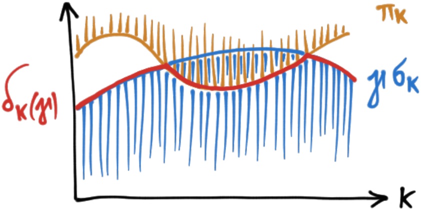
$$\sum_{k=1}^{n} \left(\sqrt{|\pi_k|^2 - |\delta_k|^2}\,|0\rangle + \delta_k\,|1\rangle\right)|k\rangle|\xi(k)\rangle$$

Problem

- $\min_{\boldsymbol{\delta}} 1/\|\boldsymbol{\delta}\|_2$ s.t. $\boldsymbol{\sigma} \cdot \hat{\boldsymbol{\delta}} \geq \sqrt{1-\varepsilon}$ and $0 \leq \delta_k \leq \pi_k$
- This can be stated as an SDP

Optimal solution

- Let $\delta_k(\gamma) = \min\{\pi_k, \gamma\sigma_k\}$



Optimization

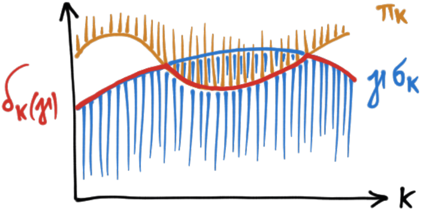
$$\sum_{k=1}^{n}\left(\sqrt{|\pi_k|^2-|\delta_k|^2}\,|0\rangle+\delta_k\,|1\rangle\right)|k\rangle|\xi(k)\rangle$$

Problem

- $\min_{\boldsymbol{\delta}} 1/\|\boldsymbol{\delta}\|_2$ s.t. $\boldsymbol{\sigma}\cdot\hat{\boldsymbol{\delta}} \geq \sqrt{1-\varepsilon}$ and $0 \leq \delta_k \leq \pi_k$
- This can be stated as an SDP

Optimal solution

- Let $\delta_k(\gamma) = \min\{\pi_k, \gamma\sigma_k\}$
- Choose $\bar{\gamma} = \max\gamma$ s.t. $\boldsymbol{\sigma}\cdot\hat{\boldsymbol{\delta}}(\gamma) \geq \sqrt{1-\varepsilon}$



Optimization

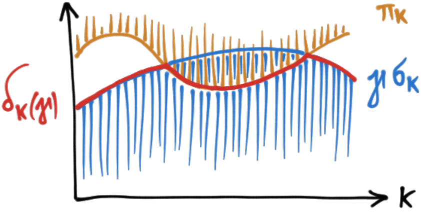
$$\sum_{k=1}^{n} \left(\sqrt{|\pi_k|^2 - |\delta_k|^2}\,|0\rangle + \delta_k\,|1\rangle\right)|k\rangle|\xi(k)\rangle$$

Problem

- $\min_{\boldsymbol{\delta}} 1/\|\boldsymbol{\delta}\|_2$ s.t. $\boldsymbol{\sigma} \cdot \hat{\boldsymbol{\delta}} \geq \sqrt{1-\varepsilon}$ and $0 \leq \delta_k \leq \pi_k$
- This can be stated as an SDP

Optimal solution

- Let $\delta_k(\gamma) = \min\{\pi_k, \gamma\sigma_k\}$
- Choose $\bar{\gamma} = \max \gamma$ s.t. $\boldsymbol{\sigma} \cdot \hat{\boldsymbol{\delta}}(\gamma) \geq \sqrt{1-\varepsilon}$



Main theorem

The quantum query complexity of the ε-approximate $\boldsymbol{\pi} \to \boldsymbol{\sigma}$ quantum resampling problem is $\Theta(1/\|\boldsymbol{\delta}(\bar{\gamma})\|_2)$

Weak vs. *strong* quantum rejection sampling

Weak quantum resampling problem

- **Given:** Description of $\boldsymbol{\pi}, \boldsymbol{\sigma} \in \mathbb{R}_+^n$
  Oracle $O : |0\rangle \mapsto |\pi\rangle = \sum_{k=1}^n \pi_k |k\rangle|\xi(k)\rangle$
- **Task:** Prepare $|\sigma\rangle = \sum_{k=1}^n \sigma_k |k\rangle|\xi(k)\rangle$

Strong quantum resampling problem

- **Given:** Description of entry-wise ratios $\boldsymbol{\sigma}/\boldsymbol{\pi}$
  Reflection $\mathrm{ref}_{|\pi\rangle} = I - 2|\pi\rangle\langle\pi|$
  One copy of $|\pi\rangle = \sum_{k=1}^n \pi_k |k\rangle|\xi(k)\rangle$
- **Task:** Prepare $|\sigma\rangle = \sum_{k=1}^n \sigma_k |k\rangle|\xi(k)\rangle$

Strong quantum rejection sampling algorithm

The $\boldsymbol{\tau}$-rotation

Let $\boldsymbol{\tau} = \sin\theta \cdot \boldsymbol{\sigma}/\boldsymbol{\pi}$ for θ such that $\max_k \tau_k \leq 1$. Define

$$R_{\boldsymbol{\tau}} = \sum_{k=1}^{n} \begin{pmatrix} \sqrt{1-\tau_k^2} & -\tau_k \\ \tau_k & \sqrt{1-\tau_k^2} \end{pmatrix} \otimes |k\rangle\langle k| \otimes I$$

Strong quantum rejection sampling algorithm

The $\boldsymbol{\tau}$-rotation

Let $\boldsymbol{\tau} = \sin\theta \cdot \boldsymbol{\sigma}/\boldsymbol{\pi}$ for θ such that $\max_k \tau_k \leq 1$. Define

$$R_{\boldsymbol{\tau}} = \sum_{k=1}^{n} \begin{pmatrix} \sqrt{1-\tau_k^2} & -\tau_k \\ \tau_k & \sqrt{1-\tau_k^2} \end{pmatrix} \otimes |k\rangle\langle k| \otimes I$$

Recall that $|\pi\rangle = \sum_{k=1}^{n} \pi_k |k\rangle|\xi(k)\rangle$.

Strong quantum rejection sampling algorithm

The $\boldsymbol{\tau}$-rotation

Let $\boldsymbol{\tau} = \sin\theta \cdot \boldsymbol{\sigma}/\boldsymbol{\pi}$ for θ such that $\max_k \tau_k \leq 1$. Define

$$R_{\boldsymbol{\tau}} = \sum_{k=1}^{n} \begin{pmatrix} \sqrt{1-\tau_k^2} & -\tau_k \\ \tau_k & \sqrt{1-\tau_k^2} \end{pmatrix} \otimes |k\rangle\langle k| \otimes I$$

Recall that $|\pi\rangle = \sum_{k=1}^{n} \pi_k |k\rangle|\xi(k)\rangle$. Then

$$R_{\boldsymbol{\tau}} \cdot |0\rangle|\pi\rangle = \sum_{k=1}^{n} \Big(\sqrt{1-\tau_k^2}\pi_k\,|0\rangle + \tau_k\pi_k\,|1\rangle\Big)|k\rangle|\xi(k)\rangle$$

Strong quantum rejection sampling algorithm

The $\boldsymbol{\tau}$-rotation

Let $\boldsymbol{\tau} = \sin\theta \cdot \boldsymbol{\sigma}/\boldsymbol{\pi}$ for θ such that $\max_k \tau_k \leq 1$. Define

$$R_{\boldsymbol{\tau}} = \sum_{k=1}^{n} \begin{pmatrix} \sqrt{1-\tau_k^2} & -\tau_k \\ \tau_k & \sqrt{1-\tau_k^2} \end{pmatrix} \otimes |k\rangle\langle k| \otimes I$$

Recall that $|\pi\rangle = \sum_{k=1}^{n} \pi_k |k\rangle|\xi(k)\rangle$. Then

$$R_{\boldsymbol{\tau}} \cdot |0\rangle|\pi\rangle = \sum_{k=1}^{n} \Big(\sqrt{1-\tau_k^2}\pi_k \,|0\rangle + \tau_k\pi_k \,|1\rangle\Big)|k\rangle|\xi(k)\rangle$$

Note that $\tau_k\pi_k = \sin\theta \cdot \sigma_k$.

Strong quantum rejection sampling algorithm

The $\boldsymbol{\tau}$-rotation

Let $\boldsymbol{\tau} = \sin\theta \cdot \boldsymbol{\sigma}/\boldsymbol{\pi}$ for θ such that $\max_k \tau_k \leq 1$. Define

$$R_{\boldsymbol{\tau}} = \sum_{k=1}^{n} \begin{pmatrix} \sqrt{1-\tau_k^2} & -\tau_k \\ \tau_k & \sqrt{1-\tau_k^2} \end{pmatrix} \otimes |k\rangle\langle k| \otimes I$$

Recall that $|\pi\rangle = \sum_{k=1}^{n} \pi_k |k\rangle|\xi(k)\rangle$. Then

$$\begin{aligned} R_{\boldsymbol{\tau}} \cdot |0\rangle|\pi\rangle &= \sum_{k=1}^{n} \Big(\sqrt{1-\tau_k^2}\pi_k\,|0\rangle + \tau_k\pi_k\,|1\rangle\Big)|k\rangle|\xi(k)\rangle \\ &= \cos\theta|0\rangle|\text{🐵}\rangle + \sin\theta|1\rangle|\sigma\rangle \end{aligned}$$

Note that $\tau_k\pi_k = \sin\theta \cdot \sigma_k$.

Strong quantum rejection sampling algorithm

Amplitude amplification

Let $|\Psi\rangle = R_{\boldsymbol{\tau}} \cdot |0\rangle|\pi\rangle = \cos\theta|0\rangle|\text{🐵}\rangle + \sin\theta|1\rangle|\sigma\rangle$. One step of amplitude amplification is given by

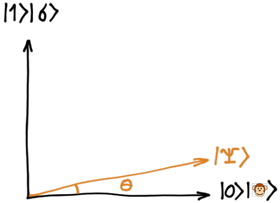
$$\mathcal{A} = \mathrm{ref}_{|\Psi\rangle} \cdot \mathrm{ref}_{|1\rangle\otimes I} = (R_{\boldsymbol{\tau}} \cdot \mathrm{ref}_{|0\rangle|\pi\rangle} \cdot R_{\boldsymbol{\tau}}^{\dagger}) \cdot (Z \otimes I)$$

Strong quantum rejection sampling algorithm

Amplitude amplification

Let $|\Psi\rangle = R_{\boldsymbol{\tau}} \cdot |0\rangle|\pi\rangle = \cos\theta|0\rangle|\text{🐵}\rangle + \sin\theta|1\rangle|\sigma\rangle$. One step of amplitude amplification is given by

$$\mathcal{A} = \mathrm{ref}_{|\Psi\rangle} \cdot \mathrm{ref}_{|1\rangle\otimes I} = (R_{\boldsymbol{\tau}} \cdot \mathrm{ref}_{|0\rangle|\pi\rangle} \cdot R_{\boldsymbol{\tau}}^{\dagger}) \cdot (Z \otimes I)$$



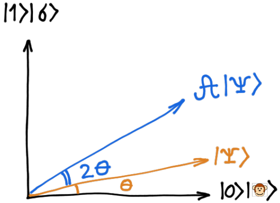
Strong quantum rejection sampling algorithm

Amplitude amplification

Let $|\Psi\rangle = R_{\boldsymbol{\tau}} \cdot |0\rangle|\pi\rangle = \cos\theta|0\rangle|\text{🐵}\rangle + \sin\theta|1\rangle|\sigma\rangle$. One step of amplitude amplification is given by

$$\mathcal{A} = \mathrm{ref}_{|\Psi\rangle} \cdot \mathrm{ref}_{|1\rangle\otimes I} = (R_{\boldsymbol{\tau}} \cdot \mathrm{ref}_{|0\rangle|\pi\rangle} \cdot R_{\boldsymbol{\tau}}^{\dagger}) \cdot (Z \otimes I)$$

This is a rotation by 2θ in the 2-dim subspace $\{|0\rangle|\text{🐵}\rangle, |1\rangle|\sigma\rangle\}$.



Strong quantum rejection sampling algorithm

Amplitude amplification

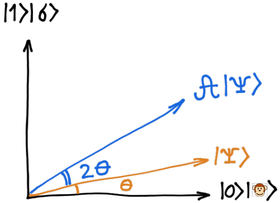
Let $|\Psi\rangle = R_{\boldsymbol{\tau}} \cdot |0\rangle|\pi\rangle = \cos\theta|0\rangle|\text{🐵}\rangle + \sin\theta|1\rangle|\sigma\rangle$. One step of amplitude amplification is given by

$$\mathcal{A} = \mathrm{ref}_{|\Psi\rangle} \cdot \mathrm{ref}_{|1\rangle\otimes I} = (R_{\boldsymbol{\tau}} \cdot \mathrm{ref}_{|0\rangle|\pi\rangle} \cdot R_{\boldsymbol{\tau}}^{\dagger}) \cdot (Z \otimes I)$$

This is a rotation by 2θ in the 2-dim subspace $\{|0\rangle|\text{🐵}\rangle, |1\rangle|\sigma\rangle\}$.

Algorithm

1. Start with $|0\rangle|\pi\rangle$ and $l = 0$



Strong quantum rejection sampling algorithm

Amplitude amplification

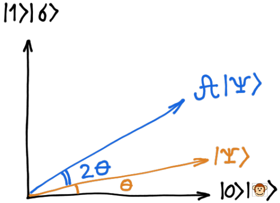
Let $|\Psi\rangle = R_{\boldsymbol{\tau}} \cdot |0\rangle|\pi\rangle = \cos\theta|0\rangle|\text{🐵}\rangle + \sin\theta|1\rangle|\sigma\rangle$. One step of amplitude amplification is given by

$$\mathcal{A} = \mathrm{ref}_{|\Psi\rangle} \cdot \mathrm{ref}_{|1\rangle\otimes I} = (R_{\boldsymbol{\tau}} \cdot \mathrm{ref}_{|0\rangle|\pi\rangle} \cdot R_{\boldsymbol{\tau}}^{\dagger}) \cdot (Z \otimes I)$$

This is a rotation by 2θ in the 2-dim subspace $\{|0\rangle|\text{🐵}\rangle, |1\rangle|\sigma\rangle\}$.

Algorithm

1. Start with $|0\rangle|\pi\rangle$ and $l = 0$
2. Apply $R_{\boldsymbol{\tau}}$ and get $|\Psi\rangle$



Strong quantum rejection sampling algorithm

Amplitude amplification

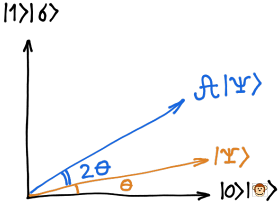
Let $|\Psi\rangle = R_{\boldsymbol{\tau}} \cdot |0\rangle|\pi\rangle = \cos\theta|0\rangle|\text{🐵}\rangle + \sin\theta|1\rangle|\sigma\rangle$. One step of amplitude amplification is given by

$$\mathcal{A} = \mathrm{ref}_{|\Psi\rangle} \cdot \mathrm{ref}_{|1\rangle\otimes I} = (R_{\boldsymbol{\tau}} \cdot \mathrm{ref}_{|0\rangle|\pi\rangle} \cdot R_{\boldsymbol{\tau}}^{\dagger}) \cdot (Z \otimes I)$$

This is a rotation by 2θ in the 2-dim subspace $\{|0\rangle|\text{🐵}\rangle, |1\rangle|\sigma\rangle\}$.

Algorithm

1. Start with $|0\rangle|\pi\rangle$ and $l = 0$
2. Apply $R_{\boldsymbol{\tau}}$ and get $|\Psi\rangle$
3. Measure the first register:



Strong quantum rejection sampling algorithm

Amplitude amplification

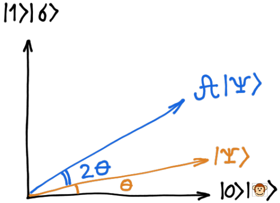
Let $|\Psi\rangle = R_{\boldsymbol{\tau}} \cdot |0\rangle|\pi\rangle = \cos\theta|0\rangle|\text{🐵}\rangle + \sin\theta|1\rangle|\sigma\rangle$. One step of amplitude amplification is given by

$$\mathcal{A} = \mathrm{ref}_{|\Psi\rangle} \cdot \mathrm{ref}_{|1\rangle\otimes I} = (R_{\boldsymbol{\tau}} \cdot \mathrm{ref}_{|0\rangle|\pi\rangle} \cdot R_{\boldsymbol{\tau}}^{\dagger}) \cdot (Z \otimes I)$$

This is a rotation by 2θ in the 2-dim subspace $\{|0\rangle|\text{🐵}\rangle, |1\rangle|\sigma\rangle\}$.

Algorithm

1. Start with $|0\rangle|\pi\rangle$ and $l = 0$
2. Apply $R_{\boldsymbol{\tau}}$ and get $|\Psi\rangle$
3. Measure the first register:
   - $|1\rangle \Rightarrow$ done



Strong quantum rejection sampling algorithm

Amplitude amplification

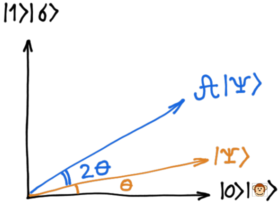
Let $|\Psi\rangle = R_{\boldsymbol{\tau}} \cdot |0\rangle|\pi\rangle = \cos\theta|0\rangle|\text{🐵}\rangle + \sin\theta|1\rangle|\sigma\rangle$. One step of amplitude amplification is given by

$$\mathcal{A} = \mathrm{ref}_{|\Psi\rangle} \cdot \mathrm{ref}_{|1\rangle\otimes I} = (R_{\boldsymbol{\tau}} \cdot \mathrm{ref}_{|0\rangle|\pi\rangle} \cdot R_{\boldsymbol{\tau}}^{\dagger}) \cdot (Z \otimes I)$$

This is a rotation by 2θ in the 2-dim subspace $\{|0\rangle|\text{🐵}\rangle, |1\rangle|\sigma\rangle\}$.

Algorithm

1. Start with $|0\rangle|\pi\rangle$ and $l = 0$
2. Apply $R_{\boldsymbol{\tau}}$ and get $|\Psi\rangle$
3. Measure the first register:
   - $|1\rangle \Rightarrow$ done
   - $|0\rangle \Rightarrow$ increase l by 1



Strong quantum rejection sampling algorithm

Amplitude amplification

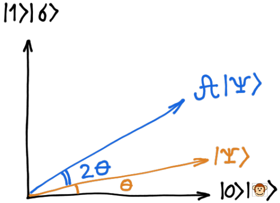
Let $|\Psi\rangle = R_{\boldsymbol{\tau}} \cdot |0\rangle|\pi\rangle = \cos\theta|0\rangle|\text{🐵}\rangle + \sin\theta|1\rangle|\sigma\rangle$. One step of amplitude amplification is given by

$$\mathcal{A} = \mathrm{ref}_{|\Psi\rangle} \cdot \mathrm{ref}_{|1\rangle\otimes I} = (R_{\boldsymbol{\tau}} \cdot \mathrm{ref}_{|0\rangle|\pi\rangle} \cdot R_{\boldsymbol{\tau}}^{\dagger}) \cdot (Z \otimes I)$$

This is a rotation by 2θ in the 2-dim subspace $\{|0\rangle|\text{🐵}\rangle, |1\rangle|\sigma\rangle\}$.

Algorithm

1. Start with $|0\rangle|\pi\rangle$ and $l = 0$
2. Apply $R_{\boldsymbol{\tau}}$ and get $|\Psi\rangle$
3. Measure the first register:
   - $|1\rangle \Rightarrow$ done
   - $|0\rangle \Rightarrow$ increase l by 1
4. Pick a random $t \in \{1, \ldots, 2^l\}$



Strong quantum rejection sampling algorithm

Amplitude amplification

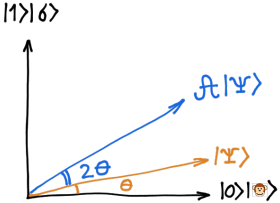
Let $|\Psi\rangle = R_{\boldsymbol{\tau}} \cdot |0\rangle|\pi\rangle = \cos\theta|0\rangle|\text{🐵}\rangle + \sin\theta|1\rangle|\sigma\rangle$. One step of amplitude amplification is given by

$$\mathcal{A} = \mathrm{ref}_{|\Psi\rangle} \cdot \mathrm{ref}_{|1\rangle\otimes I} = (R_{\boldsymbol{\tau}} \cdot \mathrm{ref}_{|0\rangle|\pi\rangle} \cdot R_{\boldsymbol{\tau}}^{\dagger}) \cdot (Z \otimes I)$$

This is a rotation by 2θ in the 2-dim subspace $\{|0\rangle|\text{🐵}\rangle, |1\rangle|\sigma\rangle\}$.

Algorithm

1. Start with $|0\rangle|\pi\rangle$ and $l = 0$
2. Apply $R_{\boldsymbol{\tau}}$ and get $|\Psi\rangle$
3. Measure the first register:
   - $|1\rangle \Rightarrow$ done
   - $|0\rangle \Rightarrow$ increase l by 1
4. Pick a random $t \in \{1, \ldots, 2^l\}$
5. Apply $\mathcal{A}^t$ and go to step 3



Applications

Implicit use

- Synthesis of quantum states [Grover, 2000]
- Linear systems of equations [Harrow, Hassidim and Lloyd 2009]
- Fast amplification of QMA [Nagaj, Wocjan, Zhang, 2009]

Applications

Implicit use

- Synthesis of quantum states [Grover, 2000]
- Linear systems of equations [Harrow, Hassidim and Lloyd 2009]
- Fast amplification of QMA [Nagaj, Wocjan, Zhang, 2009]

New applications

- Speed up quantum Metropolis sampling algorithm by [Temme, Osborne, Vollbrecht, Poulin, Verstraete, 2011]
- New quantum algorithm for the hidden shift problem of any Boolean function

Applications

Implicit use

- Synthesis of quantum states [Grover, 2000]
- Linear systems of equations [Harrow, Hassidim and Lloyd 2009]
- Fast amplification of QMA [Nagaj, Wocjan, Zhang, 2009]

New applications

- Speed up quantum Metropolis sampling algorithm by [Temme, Osborne, Vollbrecht, Poulin, Verstraete, 2011]
- New quantum algorithm for the hidden shift problem of any Boolean function

Future applications

- Preparing PEPS [Schwarz, Temme, Verstraete, 2011]
- More...

Applications

Implicit use

- Synthesis of quantum states [Grover, 2000]
- Linear systems of equations [Harrow, Hassidim and Lloyd 2009]
- Fast amplification of QMA [Nagaj, Wocjan, Zhang, 2009]

New applications

- Speed up quantum Metropolis sampling algorithm by [Temme, Osborne, Vollbrecht, Poulin, Verstraete, 2011]
- New quantum algorithm for the hidden shift problem of any Boolean function [Martin's talk yesterday]

Future applications

- Preparing PEPS [Schwarz, Temme, Verstraete, 2011]
- More...

Linear systems of equations [HHL09]

Problem

- **Given:** Invertible matrix $A \in \mathbb{C}^{d \times d}$, one copy of $|b\rangle \in \mathbb{C}^d$
- **Task:** Prepare $|x\rangle / \| |x\rangle \|_2$ where $A|x\rangle = |b\rangle$

Linear systems of equations [HHL09]

Problem

- **Given:** Invertible matrix $A \in \mathbb{C}^{d \times d}$, one copy of $|b\rangle \in \mathbb{C}^d$
- **Task:** Prepare $|x\rangle / \| |x\rangle \|_2$ where $A|x\rangle = |b\rangle$

Main idea

- W.l.o.g. A is Hermitian: $A = \sum_{j=1}^{d} \lambda_j |\psi_j\rangle\langle\psi_j|$

Linear systems of equations [HHL09]

Problem

- **Given:** Invertible matrix $A \in \mathbb{C}^{d\times d}$, one copy of $|b\rangle \in \mathbb{C}^d$
- **Task:** Prepare $|x\rangle/\||x\rangle\|_2$ where $A|x\rangle = |b\rangle$

Main idea

- W.l.o.g. A is Hermitian: $A = \sum_{j=1}^{d} \lambda_j |\psi_j\rangle\langle\psi_j|$
- Let $|b\rangle = \sum_{j=1}^{d} b_j |\psi_j\rangle$

Linear systems of equations [HHL09]

Problem

- **Given:** Invertible matrix $A \in \mathbb{C}^{d\times d}$, one copy of $|b\rangle \in \mathbb{C}^d$
- **Task:** Prepare $|x\rangle/\||x\rangle\|_2$ where $A|x\rangle = |b\rangle$

Main idea

- W.l.o.g. A is Hermitian: $A = \sum_{j=1}^{d} \lambda_j |\psi_j\rangle\langle\psi_j|$
- Let $|b\rangle = \sum_{j=1}^{d} b_j |\psi_j\rangle$
- Then $|x\rangle = A^{-1}|b\rangle = \sum_{j=1}^{d} b_j/\lambda_j |\psi_j\rangle$

Linear systems of equations [HHL09]

Problem

- **Given:** Invertible matrix $A \in \mathbb{C}^{d\times d}$, one copy of $|b\rangle \in \mathbb{C}^d$
- **Task:** Prepare $|x\rangle/\||x\rangle\|_2$ where $A|x\rangle = |b\rangle$

Main idea

- W.l.o.g. A is Hermitian: $A = \sum_{j=1}^{d} \lambda_j |\psi_j\rangle\langle\psi_j|$
- Let $|b\rangle = \sum_{j=1}^{d} b_j|\psi_j\rangle$
- Then $|x\rangle = A^{-1}|b\rangle = \sum_{j=1}^{d} b_j/\lambda_j|\psi_j\rangle$

Algorithm

1. Apply phase estimation of e^{iAt} on $|b\rangle$ and get $\sum_{j=1}^{d} b_j|\psi_j\rangle|\lambda_j\rangle$

Linear systems of equations [HHL09]

Problem

- **Given:** Invertible matrix $A \in \mathbb{C}^{d\times d}$, one copy of $|b\rangle \in \mathbb{C}^d$
- **Task:** Prepare $|x\rangle/\||x\rangle\|_2$ where $A|x\rangle = |b\rangle$

Main idea

- W.l.o.g. A is Hermitian: $A = \sum_{j=1}^d \lambda_j |\psi_j\rangle\langle\psi_j|$
- Let $|b\rangle = \sum_{j=1}^d b_j |\psi_j\rangle$
- Then $|x\rangle = A^{-1}|b\rangle = \sum_{j=1}^d b_j/\lambda_j |\psi_j\rangle$

Algorithm

1. Apply phase estimation of e^{iAt} on $|b\rangle$ and get $\sum_{j=1}^d b_j|\psi_j\rangle|\lambda_j\rangle$
2. Convert this state to $c \cdot \sum_{j=1}^d b_j/\lambda_j|\psi_j\rangle|\lambda_j\rangle$

Linear systems of equations [HHL09]

Problem

- **Given:** Invertible matrix $A \in \mathbb{C}^{d \times d}$, one copy of $|b\rangle \in \mathbb{C}^d$
- **Task:** Prepare $|x\rangle / \||x\rangle\|_2$ where $A|x\rangle = |b\rangle$

Main idea

- W.l.o.g. A is Hermitian: $A = \sum_{j=1}^{d} \lambda_j |\psi_j\rangle\langle\psi_j|$
- Let $|b\rangle = \sum_{j=1}^{d} b_j |\psi_j\rangle$
- Then $|x\rangle = A^{-1}|b\rangle = \sum_{j=1}^{d} b_j/\lambda_j |\psi_j\rangle$

Algorithm

1. Apply phase estimation of e^{iAt} on $|b\rangle$ and get $\sum_{j=1}^{d} b_j |\psi_j\rangle|\lambda_j\rangle$
2. Convert this state to $c \cdot \sum_{j=1}^{d} b_j/\lambda_j |\psi_j\rangle|\lambda_j\rangle$
3. Undo phase estimation and get $c \cdot \sum_{j=1}^{d} b_j/\lambda_j |\psi_j\rangle = |x\rangle$

Classical Metropolis sampling [MRRTT53]

Problem

- **Given:** A set of configurations S where $j \in S$ has energy E_j
- **Task:** Sample from $p(j) = \exp(-\beta E_j)/Z(\beta)$
  (Gibbs distribution) where $Z(\beta) = \sum_j \exp(-\beta E_j)$

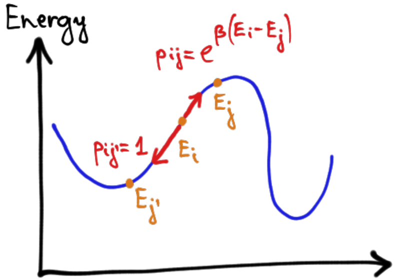
Classical Metropolis sampling [MRRTT53]

Problem

- **Given:** A set of configurations S where $j \in S$ has energy E_j
- **Task:** Sample from $p(j) = \exp(-\beta E_j)/Z(\beta)$
  (Gibbs distribution) where $Z(\beta) = \sum_j \exp(-\beta E_j)$

Algorithm

1. Start from a random $i \in S$



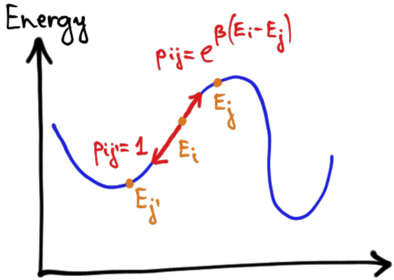
Classical Metropolis sampling [MRRTT53]

Problem

- **Given:** A set of configurations S where $j \in S$ has energy E_j
- **Task:** Sample from $p(j) = \exp(-\beta E_j)/Z(\beta)$ (Gibbs distribution) where $Z(\beta) = \sum_j \exp(-\beta E_j)$

Algorithm

1. Start from a random $i \in S$
2. Repeat several times:



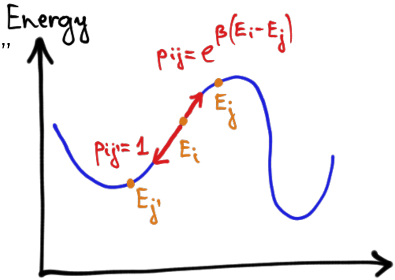
Classical Metropolis sampling [MRRTT53]

Problem

- **Given:** A set of configurations S where $j \in S$ has energy E_j
- **Task:** Sample from $p(j) = \exp(-\beta E_j)/Z(\beta)$
  (Gibbs distribution) where $Z(\beta) = \sum_j \exp(-\beta E_j)$

Algorithm

1. Start from a random $i \in S$
2. Repeat several times:
   - Let $j := i +$ "loc. rand. perturb."



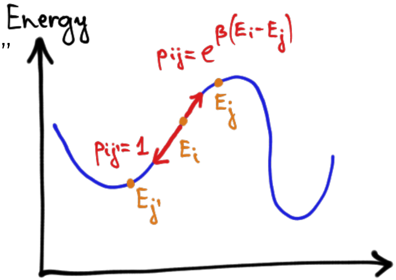
Classical Metropolis sampling [MRRTT53]

Problem

- **Given:** A set of configurations S where $j \in S$ has energy E_j
- **Task:** Sample from $p(j) = \exp(-\beta E_j)/Z(\beta)$ (Gibbs distribution) where $Z(\beta) = \sum_j \exp(-\beta E_j)$

Algorithm

1. Start from a random $i \in S$
2. Repeat several times:
   - Let $j := i +$ "loc. rand. perturb."
   - Set $i := j$ with probability $p_{ij} = \min\{1, e^{\beta(E_i - E_j)}\}$



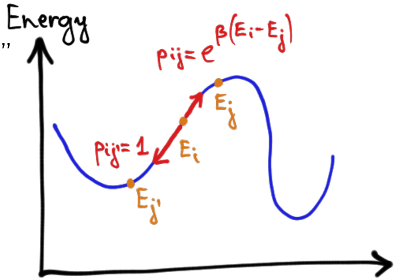
Classical Metropolis sampling [MRRTT53]

Problem

- **Given:** A set of configurations S where $j \in S$ has energy E_j
- **Task:** Sample from $p(j) = \exp(-\beta E_j)/Z(\beta)$
  (Gibbs distribution) where $Z(\beta) = \sum_j \exp(-\beta E_j)$

Algorithm

1. Start from a random $i \in S$
2. Repeat several times:
   - Let $j := i +$ "loc. rand. perturb."
   - Set $i := j$ with probability
     $p_{ij} = \min\{1, e^{\beta(E_i - E_j)}\}$
     - if $E_j \leq E_i$ then $i := j$
     - if $E_j > E_i$ then $i := j$
       with prob. $e^{\beta(E_i - E_j)}$



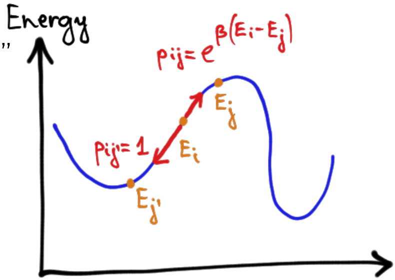
Classical Metropolis sampling [MRRTT53]

Problem

- **Given:** A set of configurations S where $j \in S$ has energy E_j
- **Task:** Sample from $p(j) = \exp(-\beta E_j)/Z(\beta)$ (Gibbs distribution) where $Z(\beta) = \sum_j \exp(-\beta E_j)$

Algorithm

1. Start from a random $i \in S$
2. Repeat several times:
   - Let $j := i +$ "loc. rand. perturb."
   - Set $i := j$ with probability $p_{ij} = \min\{1, e^{\beta(E_i - E_j)}\}$
     - if $E_j \leq E_i$ then $i := j$
     - if $E_j > E_i$ then $i := j$ with prob. $e^{\beta(E_i - E_j)}$
3. Output the final configuration i



Quantum Metropolis sampling [TOVPV11] + QR sampling

Problem

- **Given:** Ability to implement Hamiltonian H
- **Task:** Prepare the thermal state
  $\rho = \exp(-\beta H)/Z(\beta)$

Quantum Metropolis sampling [TOVPV11] + QR sampling

Problem

- **Given:** Ability to implement Hamiltonian H
- **Task:** Prepare the thermal state $\rho = \exp(-\beta H)/Z(\beta)$

Note: if $H = \sum_j E_j |\psi_j\rangle\langle\psi_j|$ for some *unknown* E_j and $|\psi_j\rangle$, then we want to prepare $|\psi_j\rangle$ w.p. $p(j) = \exp(-\beta E_j)/Z(\beta)$

Quantum Metropolis sampling [TOVPV11] + QR sampling

Problem

- **Given:** Ability to implement Hamiltonian H
- **Task:** Prepare the thermal state
  $\rho = \exp(-\beta H)/Z(\beta) = \sum_j e^{-\beta E_j} |\psi_j\rangle\langle\psi_j|/Z(\beta)$

Note: if $H = \sum_j E_j |\psi_j\rangle\langle\psi_j|$ for some *unknown* E_j and $|\psi_j\rangle$, then we want to prepare $|\psi_j\rangle$ w.p. $p(j) = \exp(-\beta E_j)/Z(\beta)$

Quantum Metropolis sampling [TOVPV11] + QR sampling

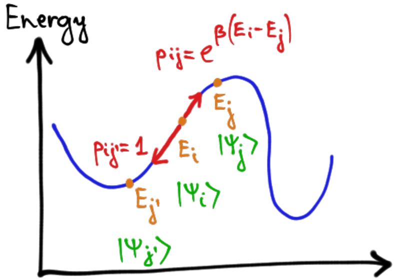
Problem

- **Given:** Ability to implement Hamiltonian H
- **Task:** Prepare the thermal state $\rho = \exp(-\beta H)/Z(\beta) = \sum_j e^{-\beta E_j} |\psi_j\rangle\langle\psi_j|/Z(\beta)$

Note: if $H = \sum_j E_j |\psi_j\rangle\langle\psi_j|$ for some *unknown* E_j and $|\psi_j\rangle$, then we want to prepare $|\psi_j\rangle$ w.p. $p(j) = \exp(-\beta E_j)/Z(\beta)$

Main idea

Set up the same *classical* random walk, but use a *quantum* subroutine to implement each steps and also keep track of the current eigenvector $|\psi_i\rangle$



Quantum Metropolis sampling [TOVPV11] + QR sampling

Recall, $H = \sum_j E_j |\psi_j\rangle\langle\psi_j|$. Let $\mathcal{U}$ be a universal set of quantum gates and let $U_k \in \mathcal{U}$ act as $U_k|\psi_i\rangle = \sum_j u_{ij}^{(k)} |\psi_j\rangle$.

Quantum Metropolis sampling [TOVPV11] + QR sampling

Recall, $H = \sum_j E_j |\psi_j\rangle\langle\psi_j|$. Let $\mathcal{U}$ be a universal set of quantum gates and let $U_k \in \mathcal{U}$ act as $U_k|\psi_i\rangle = \sum_j u_{ij}^{(k)}|\psi_j\rangle$.

Algorithm

Metropolis move from i to j with prob. $p_{ij} = \min\{1, e^{\beta(E_i - E_j)}\}$:

1. $|\psi_i\rangle|E_i\rangle \leftarrow$ prepare for random i using QPE

Quantum Metropolis sampling [TOVPV11] + QR sampling

Recall, $H = \sum_j E_j |\psi_j\rangle\langle\psi_j|$. Let $\mathcal{U}$ be a universal set of quantum gates and let $U_k \in \mathcal{U}$ act as $U_k|\psi_i\rangle = \sum_j u_{ij}^{(k)} |\psi_j\rangle$.

Algorithm

Metropolis move from i to j with prob. $p_{ij} = \min\{1, e^{\beta(E_i - E_j)}\}$:

1. $|\psi_i\rangle|E_i\rangle \leftarrow$ prepare for random i using QPE
2. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_k |k\rangle|\psi_i\rangle|E_i\rangle \leftarrow$ add a uniform superposition over $\mathcal{U}$

Quantum Metropolis sampling [TOVPV11] + QR sampling

Recall, $H = \sum_j E_j |\psi_j\rangle\langle\psi_j|$. Let $\mathcal{U}$ be a universal set of quantum gates and let $U_k \in \mathcal{U}$ act as $U_k|\psi_i\rangle = \sum_j u_{ij}^{(k)} |\psi_j\rangle$.

Algorithm

Metropolis move from i to j with prob. $p_{ij} = \min\{1, e^{\beta(E_i - E_j)}\}$:

1. $|\psi_i\rangle|E_i\rangle \leftarrow$ prepare for random i using QPE
2. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_k |k\rangle|\psi_i\rangle|E_i\rangle \leftarrow$ add a uniform superposition over $\mathcal{U}$
3. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_j \left[\sum_k u_{ij}^{(k)} |k\rangle\right] |\psi_j\rangle|E_i\rangle \leftarrow$ apply U_k controlled on k

Quantum Metropolis sampling [TOVPV11] + QR sampling

Recall, $H = \sum_j E_j |\psi_j\rangle\langle\psi_j|$. Let $\mathcal{U}$ be a universal set of quantum gates and let $U_k \in \mathcal{U}$ act as $U_k|\psi_i\rangle = \sum_j u_{ij}^{(k)} |\psi_j\rangle$.

Algorithm

Metropolis move from i to j with prob. $p_{ij} = \min\{1, e^{\beta(E_i - E_j)}\}$:

1. $|\psi_i\rangle|E_i\rangle \leftarrow$ prepare for random i using QPE
2. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_k |k\rangle|\psi_i\rangle|E_i\rangle \leftarrow$ add a uniform superposition over $\mathcal{U}$
3. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_j \left[\sum_k u_{ij}^{(k)} |k\rangle\right] |\psi_j\rangle|E_i\rangle \leftarrow$ apply U_k controlled on k
4. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_j \left[\sum_k u_{ij}^{(k)} |k\rangle\right] |\psi_j\rangle|E_i\rangle|E_j\rangle \leftarrow$ attach $|E_j\rangle$ using QPE

Quantum Metropolis sampling [TOVPV11] + QR sampling

Recall, $H = \sum_j E_j |\psi_j\rangle\langle\psi_j|$. Let $\mathcal{U}$ be a universal set of quantum gates and let $U_k \in \mathcal{U}$ act as $U_k|\psi_i\rangle = \sum_j u_{ij}^{(k)}|\psi_j\rangle$.

Algorithm

Metropolis move from i to j with prob. $p_{ij} = \min\{1, e^{\beta(E_i - E_j)}\}$:

1. $|\psi_i\rangle|E_i\rangle \leftarrow$ prepare for random i using QPE
2. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_k |k\rangle|\psi_i\rangle|E_i\rangle \leftarrow$ add a uniform superposition over $\mathcal{U}$
3. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_j \left[\sum_k u_{ij}^{(k)}|k\rangle\right]|\psi_j\rangle|E_i\rangle \leftarrow$ apply U_k controlled on k
4. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_j \left[\sum_k u_{ij}^{(k)}|k\rangle\right]|\psi_j\rangle|E_i\rangle|E_j\rangle \leftarrow$ attach $|E_j\rangle$ using QPE
5. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_j \sqrt{p_{ij}}\left[\sum_k u_{ij}^{(k)}|k\rangle\right]|\psi_j\rangle|E_i\rangle|E_j\rangle \leftarrow$ using QRS

Quantum Metropolis sampling [TOVPV11] + QR sampling

Recall, $H = \sum_j E_j |\psi_j\rangle\langle\psi_j|$. Let $\mathcal{U}$ be a universal set of quantum gates and let $U_k \in \mathcal{U}$ act as $U_k|\psi_i\rangle = \sum_j u_{ij}^{(k)} |\psi_j\rangle$.

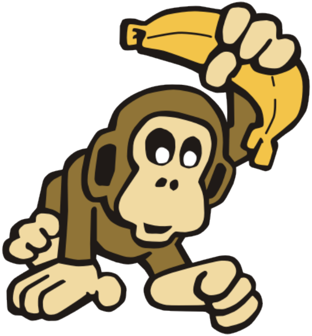
Algorithm

Metropolis move from i to j with prob. $p_{ij} = \min\{1, e^{\beta(E_i - E_j)}\}$:

1. $|\psi_i\rangle|E_i\rangle \leftarrow$ prepare for random i using QPE
2. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_k |k\rangle|\psi_i\rangle|E_i\rangle \leftarrow$ add a uniform superposition over $\mathcal{U}$
3. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_j \left[\sum_k u_{ij}^{(k)} |k\rangle\right] |\psi_j\rangle|E_i\rangle \leftarrow$ apply U_k controlled on k
4. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_j \left[\sum_k u_{ij}^{(k)} |k\rangle\right] |\psi_j\rangle|E_i\rangle|E_j\rangle \leftarrow$ attach $|E_j\rangle$ using QPE
5. $\frac{1}{\sqrt{|\mathcal{U}|}} \sum_j \sqrt{p_{ij}} \left[\sum_k u_{ij}^{(k)} |k\rangle\right] |\psi_j\rangle|E_i\rangle|E_j\rangle \leftarrow$ using QRS
6. $|\psi_j\rangle|E_j\rangle \leftarrow$ after discarding $|k\rangle$ and $|E_i\rangle$

Conclusion

- Classical rejection sampling has many applications
- Quantum rejection sampling could be as useful
- Tight characterization of query complexity
- Three diverse applications:
  - Boolean hidden shift problem
  - Quantum Metropolis algorithm [TOVPV11]
  - Quantum algorithm for linear systems of equations [HHL09]



Thank You!

Funding:

Boolean hidden shift problem (BHSP)

Problem

- **Given:** Complete knowledge of $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ and access to a black-box oracle for $f_s(x) := f(x+s)$

$$x \Rightarrow \boxed{} \Rightarrow f_s(x)$$

- **Determine:** The hidden shift s

Boolean hidden shift problem (BHSP)

Problem

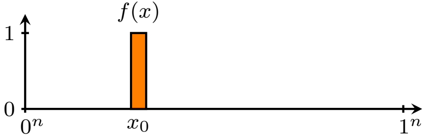
- **Given:** Complete knowledge of $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ and access to a black-box oracle for $f_s(x) := f(x+s)$

$$x \Rightarrow \blacksquare \Rightarrow f_s(x)$$

- **Determine:** The hidden shift s

Delta functions are hard

- $f(x) := \delta_{x,x_0}$



Boolean hidden shift problem (BHSP)

Problem

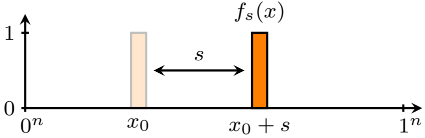
- **Given:** Complete knowledge of $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ and access to a black-box oracle for $f_s(x) := f(x+s)$

$$x \Rightarrow \boxed{} \Rightarrow f_s(x)$$

- **Determine:** The hidden shift s

Delta functions are hard

- $f(x) := \delta_{x,x_0}$



Boolean hidden shift problem (BHSP)

Problem

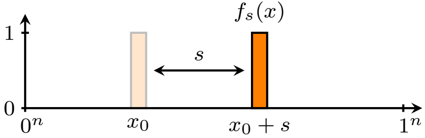
- **Given:** Complete knowledge of $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ and access to a black-box oracle for $f_s(x) := f(x+s)$

$$x \Rightarrow \blacksquare \Rightarrow f_s(x)$$

- **Determine:** The hidden shift s

Delta functions are hard

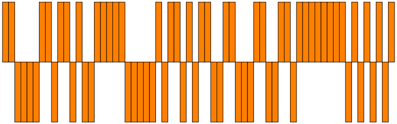
- $f(x) := \delta_{x,x_0}$
- Equivalent to Grover's search: $\Theta(\sqrt{2^n})$



Fourier transform of Boolean functions

The ± 1-function (normalized)

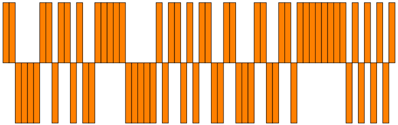
- $F(x) := \frac{1}{\sqrt{2^n}}(-1)^{f(x)}$



Fourier transform of Boolean functions

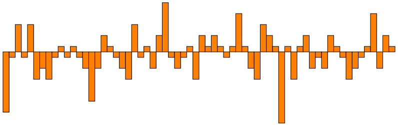
The ± 1-function (normalized)

- $F(x) := \frac{1}{\sqrt{2^n}}(-1)^{f(x)}$



Fourier transform

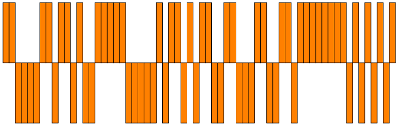
- $\hat{F}(w) := \langle w|H^{\otimes n}|F\rangle$



Fourier transform of Boolean functions

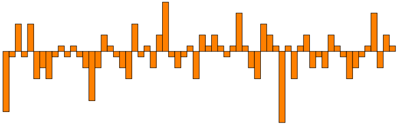
The ± 1-function (normalized)

- $F(x) := \frac{1}{\sqrt{2^n}}(-1)^{f(x)}$



Fourier transform

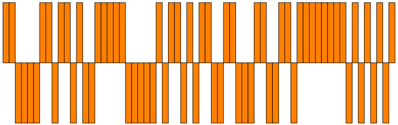
- $\hat{F}(w) := \langle w|H^{\otimes n}|F\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\mathbb{Z}_2^n}(-1)^{w\cdot x}F(x)$



Fourier transform of Boolean functions

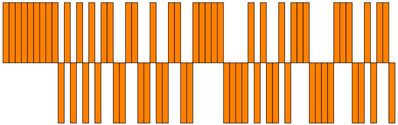
The ± 1-function (normalized)

- $F(x) := \frac{1}{\sqrt{2^n}}(-1)^{f(x)}$



Fourier transform

- $\hat{F}(w) := \langle w|H^{\otimes n}|F\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\mathbb{Z}_2^n}(-1)^{w\cdot x}F(x)$



Function f is **bent** if $\forall w : |\hat{F}(w)| = \frac{1}{\sqrt{2^n}}$

Bent functions are easy

Preparing the "phase state"

- Phase oracle $O_{f_s} : |x\rangle \mapsto (-1)^{f_s(x)}|x\rangle$

Bent functions are easy

Preparing the "phase state"

- Phase oracle $O_{f_s} : |x\rangle \mapsto (-1)^{f_s(x)}|x\rangle$

$$|0\rangle^{\otimes n} \; — \boxed{H^{\otimes n}} — \boxed{O_{f_s}} — \boxed{H^{\otimes n}} — \; |\Phi(s)\rangle$$

- $|\Phi(s)\rangle := \sum_{w\in\mathbb{Z}_2^n}(-1)^{s\cdot w}\hat{F}(w)|w\rangle$

Bent functions are easy

Preparing the “phase state”

- Phase oracle $O_{f_s} : |x\rangle \mapsto (-1)^{f_s(x)}|x\rangle$

$$|0\rangle^{\otimes n} \; - \boxed{H^{\otimes n}} - \boxed{O_{f_s}} - \boxed{H^{\otimes n}} - \; |\Phi(s)\rangle$$

- $|\Phi(s)\rangle := \sum_{w\in\mathbb{Z}_2^n}(-1)^{s\cdot w}\hat{F}(w)|w\rangle$

Algorithm [Rötteler’10]

- Prepare $|\Phi(s)\rangle$

Bent functions are easy

Preparing the "phase state"

- Phase oracle $O_{f_s} : |x\rangle \mapsto (-1)^{f_s(x)}|x\rangle$

$$|0\rangle^{\otimes n} \; - \boxed{H^{\otimes n}} - \boxed{O_{f_s}} - \boxed{H^{\otimes n}} - \; |\Phi(s)\rangle$$

- $|\Phi(s)\rangle := \sum_{w\in\mathbb{Z}_2^n}(-1)^{s\cdot w}\hat{F}(w)|w\rangle$

Algorithm [Rötteler'10]

- Prepare $|\Phi(s)\rangle$
- Apply $D := \mathrm{diag}\left(\frac{|\hat{F}(w)|}{\hat{F}(w)}\right)$ [Curtis & Meyer'04] and get $D|\Phi(s)\rangle = \sum_{w\in\mathbb{Z}_2^n}(-1)^{s\cdot w}|\hat{F}(w)||w\rangle$

Bent functions are easy

Preparing the "phase state"

- Phase oracle $O_{f_s} : |x\rangle \mapsto (-1)^{f_s(x)}|x\rangle$

$$|0\rangle^{\otimes n} \; - \boxed{H^{\otimes n}} - \boxed{O_{f_s}} - \boxed{H^{\otimes n}} - \; |\Phi(s)\rangle$$

- $|\Phi(s)\rangle := \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \hat{F}(w)|w\rangle$

Algorithm [Rötteler'10]

- Prepare $|\Phi(s)\rangle$
- Apply $D := \mathrm{diag}\left(\frac{|\hat{F}(w)|}{\hat{F}(w)}\right)$ [Curtis & Meyer'04] and get $D|\Phi(s)\rangle = \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} |\hat{F}(w)||w\rangle$
- If f is bent then $H^{\otimes n} D|\Phi(s)\rangle = |s\rangle$

Bent functions are easy

Preparing the "phase state"

- Phase oracle $O_{f_s} : |x\rangle \mapsto (-1)^{f_s(x)}|x\rangle$

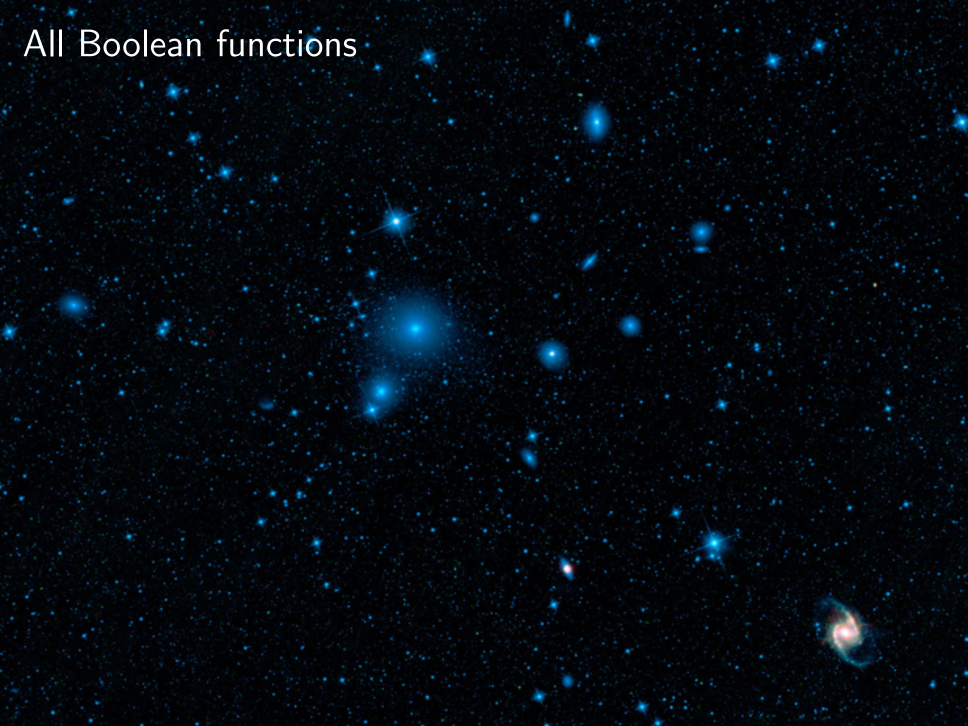
$$|0\rangle^{\otimes n} \;—\; \boxed{H^{\otimes n}} — \boxed{O_{f_s}} — \boxed{H^{\otimes n}} \;—\; |\Phi(s)\rangle$$

- $|\Phi(s)\rangle := \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \hat{F}(w)|w\rangle$

Algorithm [Rötteler'10]

- Prepare $|\Phi(s)\rangle$
- Apply $D := \mathrm{diag}\left(\frac{|\hat{F}(w)|}{\hat{F}(w)}\right)$ [Curtis & Meyer'04] and get $D|\Phi(s)\rangle = \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} |\hat{F}(w)||w\rangle$
- If f is bent then $H^{\otimes n} D|\Phi(s)\rangle = |s\rangle$
- Complexity: $\Theta(1)$

All Boolean functions

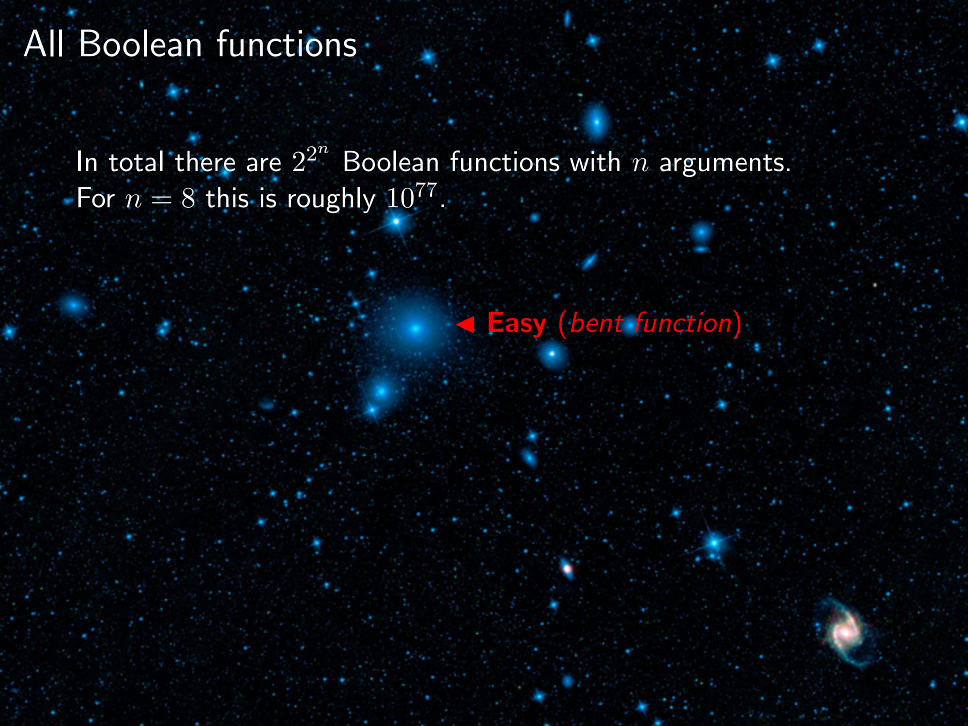


All Boolean functions

In total there are 2^{2^n} Boolean functions with n arguments.
For $n = 8$ this is roughly 10^{77}.

All Boolean functions

In total there are 2^{2^n} Boolean functions with n arguments.
For $n = 8$ this is roughly 10^{77}.



All Boolean functions

In total there are 2^{2^n} Boolean functions with n arguments.
For $n = 8$ this is roughly 10^{77}.

All Boolean functions

In total there are 2^{2^n} Boolean functions with n arguments.
For $n = 8$ this is roughly 10^{77}.

◄ **Easy** (*bent function*)

What about the rest?

Hard (*delta function*) ►

Algorithm for any Boolean function

Resampling approach

$$\sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \hat{F}(w) |w\rangle \mapsto \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \frac{1}{\sqrt{2^n}} |w\rangle$$

Algorithm for any Boolean function

Resampling approach

$$\sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \hat{F}(w) |w\rangle \mapsto \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \frac{1}{\sqrt{2^n}} |w\rangle$$

This is a quantum $\boldsymbol{\pi} \to \boldsymbol{\sigma}$ resampling problem with

$$\pi_w = \hat{F}(w) \qquad \sigma_w = \frac{1}{\sqrt{2^n}} \qquad |\xi(w)\rangle = (-1)^{s \cdot w}$$

Algorithm for any Boolean function

Resampling approach

$$\sum_{w\in\mathbb{Z}_2^n}(-1)^{s\cdot w}\hat{F}(w)|w\rangle \mapsto \sum_{w\in\mathbb{Z}_2^n}(-1)^{s\cdot w}\frac{1}{\sqrt{2^n}}|w\rangle$$

This is a quantum $\boldsymbol{\pi} \to \boldsymbol{\sigma}$ resampling problem with

$$\pi_w = \hat{F}(w) \qquad \sigma_w = \frac{1}{\sqrt{2^n}} \qquad |\xi(w)\rangle = (-1)^{s\cdot w}$$

Quantum query complexity

Recall that this can be solved using quantum rejection sampling in $O(1/\gamma)$ queries where $\gamma = \min_w \pi_w/\sigma_w$. In our case this is:

$$O\left(\frac{1}{\sqrt{2^n}\hat{F}_{\min}}\right)$$

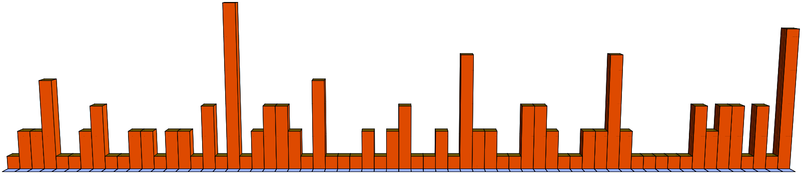
“Demo”

Algorithm

"Demo"

Algorithm

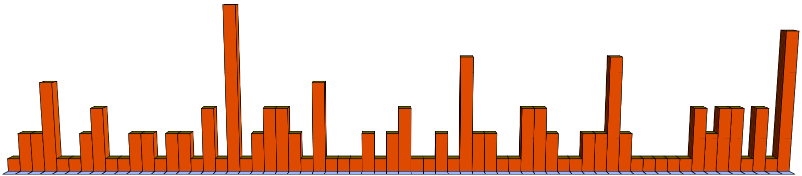
1. Prepare $|\Phi(s)\rangle = H^{\otimes n} O_{f_s} H^{\otimes n} |0\rangle^{\otimes n} = \sum_w (-1)^{s \cdot w} \hat{F}(w) |w\rangle$



“Demo”

Algorithm

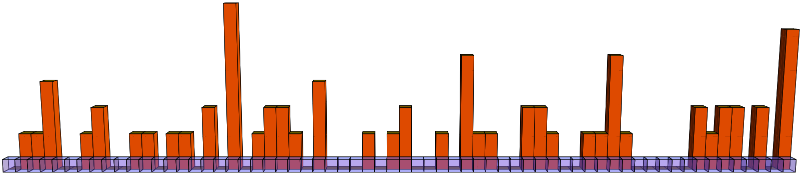
1. Prepare $|\Phi(s)\rangle = H^{\otimes n} O_{f_s} H^{\otimes n} |0\rangle^{\otimes n} = \sum_w (-1)^{s \cdot w} \hat{F}(w) |w\rangle$
2. Perform a $\boldsymbol{\delta}$-rotation where $\delta_w = \hat{F}_{\min}$ for all $w \in \mathbb{Z}_2^n$



“Demo”

Algorithm

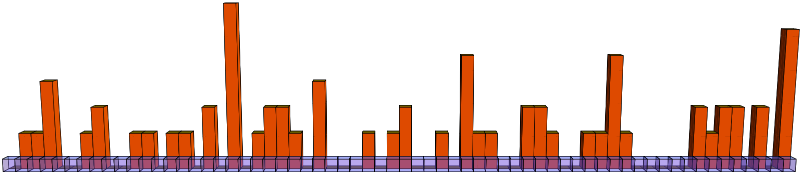
1. Prepare $|\Phi(s)\rangle = H^{\otimes n} O_{f_s} H^{\otimes n} |0\rangle^{\otimes n} = \sum_w (-1)^{s \cdot w} \hat{F}(w) |w\rangle$
2. Perform a $\boldsymbol{\delta}$-rotation where $\delta_w = \hat{F}_{\min}$ for all $w \in \mathbb{Z}_2^n$



"Demo"

Algorithm

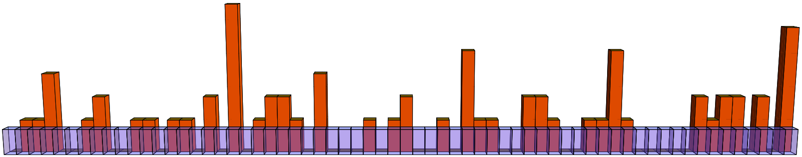
1. Prepare $|\Phi(s)\rangle = H^{\otimes n} O_{f_s} H^{\otimes n} |0\rangle^{\otimes n} = \sum_w (-1)^{s \cdot w} \hat{F}(w) |w\rangle$
2. Perform a $\boldsymbol{\delta}$-rotation where $\delta_w = \hat{F}_{\min}$ for all $w \in \mathbb{Z}_2^n$
3. Do amplitude amplification



"Demo"

Algorithm

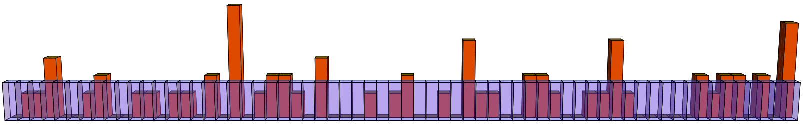
1. Prepare $|\Phi(s)\rangle = H^{\otimes n} O_{f_s} H^{\otimes n} |0\rangle^{\otimes n} = \sum_w (-1)^{s \cdot w} \hat{F}(w) |w\rangle$
2. Perform a $\boldsymbol{\delta}$-rotation where $\delta_w = \hat{F}_{\min}$ for all $w \in \mathbb{Z}_2^n$
3. Do amplitude amplification



"Demo"

Algorithm

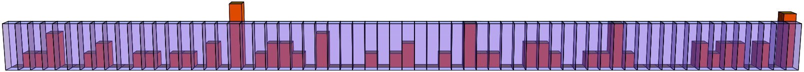
1. Prepare $|\Phi(s)\rangle = H^{\otimes n} O_{f_s} H^{\otimes n} |0\rangle^{\otimes n} = \sum_w (-1)^{s \cdot w} \hat{F}(w) |w\rangle$
2. Perform a $\boldsymbol{\delta}$-rotation where $\delta_w = \hat{F}_{\min}$ for all $w \in \mathbb{Z}_2^n$
3. Do amplitude amplification



"Demo"

Algorithm

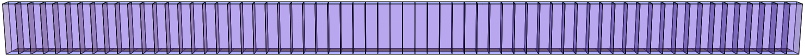
1. Prepare $|\Phi(s)\rangle = H^{\otimes n} O_{f_s} H^{\otimes n} |0\rangle^{\otimes n} = \sum_w (-1)^{s \cdot w} \hat{F}(w) |w\rangle$
2. Perform a $\boldsymbol{\delta}$-rotation where $\delta_w = \hat{F}_{\min}$ for all $w \in \mathbb{Z}_2^n$
3. Do amplitude amplification



“Demo”

Algorithm

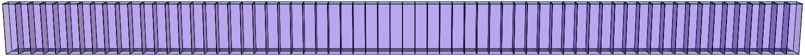
1. Prepare $|\Phi(s)\rangle = H^{\otimes n} O_{f_s} H^{\otimes n} |0\rangle^{\otimes n} = \sum_w (-1)^{s \cdot w} \hat{F}(w) |w\rangle$
2. Perform a $\boldsymbol{\delta}$-rotation where $\delta_w = \hat{F}_{\min}$ for all $w \in \mathbb{Z}_2^n$
3. Do amplitude amplification



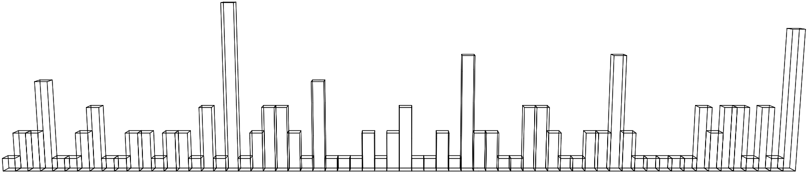
“Demo”

Algorithm

1. Prepare $|\Phi(s)\rangle = H^{\otimes n} O_{f_s} H^{\otimes n} |0\rangle^{\otimes n} = \sum_w (-1)^{s \cdot w} \hat{F}(w) |w\rangle$
2. Perform a $\boldsymbol{\delta}$-rotation where $\delta_w = \hat{F}_{\min}$ for all $w \in \mathbb{Z}_2^n$
3. Do amplitude amplification
4. Measure the resulting state in Fourier basis

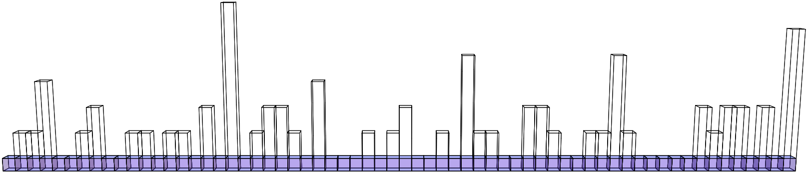


“Demo” (approximate version)



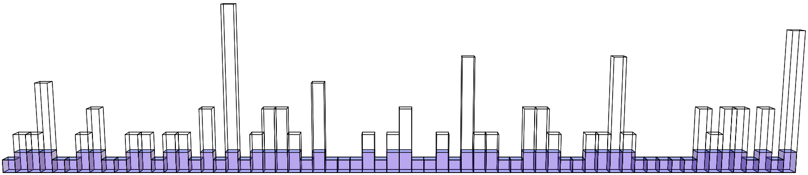
"Demo" (approximate version)

- Instead of the "flat" state



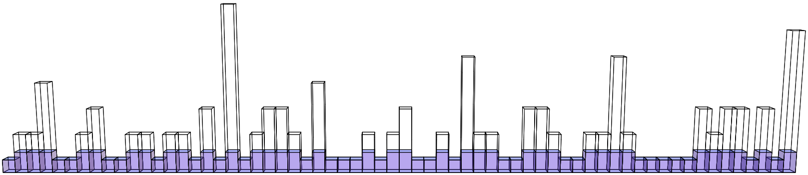
“Demo” (approximate version)

- Instead of the “flat” state aim for “*approximately* flat” state



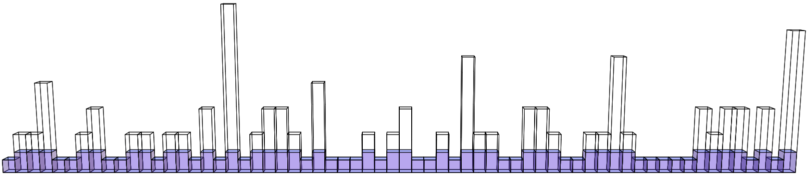
"Demo" (approximate version)

- Instead of the "flat" state aim for "*approximately* flat" state
- Fix the desired success probability p



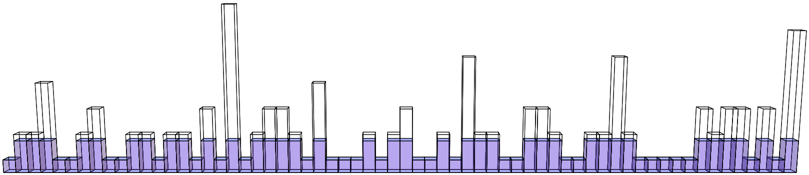
“Demo” (approximate version)

- Instead of the “flat” state aim for “*approximately* flat” state
- Fix the desired success probability p
- Optimal choice of $\boldsymbol{\delta}$ is given by the “water filling” vector $\boldsymbol{\delta}_p$ such that $\boldsymbol{\sigma}^{\mathsf{T}} \cdot \boldsymbol{\delta}_p / \|\boldsymbol{\delta}_p\|_2 \geq \sqrt{p}$ where $\sigma_w = \frac{1}{\sqrt{2^n}}$



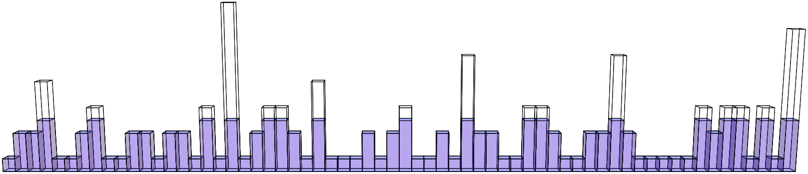
"Demo" (approximate version)

- Instead of the "flat" state aim for "*approximately* flat" state
- Fix the desired success probability p
- Optimal choice of $\boldsymbol{\delta}$ is given by the "water filling" vector $\boldsymbol{\delta}_p$ such that $\boldsymbol{\sigma}^{\mathsf{T}} \cdot \boldsymbol{\delta}_p / \|\boldsymbol{\delta}_p\|_2 \geq \sqrt{p}$ where $\sigma_w = \frac{1}{\sqrt{2^n}}$



"Demo" (approximate version)

- Instead of the "flat" state aim for "*approximately* flat" state
- Fix the desired success probability p
- Optimal choice of $\boldsymbol{\delta}$ is given by the "water filling" vector $\boldsymbol{\delta}_p$ such that $\boldsymbol{\sigma}^\mathsf{T} \cdot \boldsymbol{\delta}_p / \|\boldsymbol{\delta}_p\|_2 \geq \sqrt{p}$ where $\sigma_w = \frac{1}{\sqrt{2^n}}$



“Demo” (approximate version)

- Instead of the “flat” state aim for “*approximately* flat” state
- Fix the desired success probability p
- Optimal choice of $\boldsymbol{\delta}$ is given by the “water filling” vector $\boldsymbol{\delta}_p$ such that $\boldsymbol{\sigma}^{\mathsf{T}} \cdot \boldsymbol{\delta}_p / \|\boldsymbol{\delta}_p\|_2 \geq \sqrt{p}$ where $\sigma_w = \frac{1}{\sqrt{2^n}}$
- Query complexity: $O(1/\|\boldsymbol{\delta}_p\|_2)$