Quantum algorithms for the hidden shift problem of Boolean functions

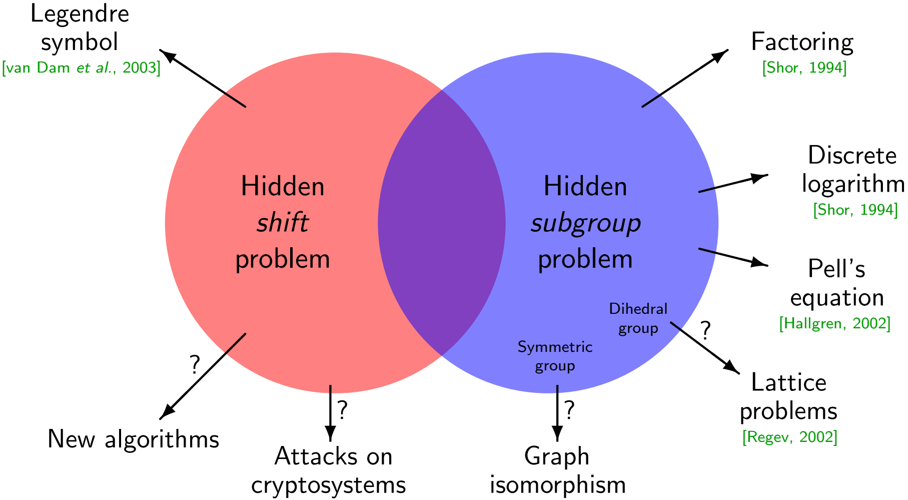
Maris Ozols
University of Waterloo, IQC
and NEC Labs

Joint work with: **Martin Rötteler** (NEC Labs)
Jérémie Roland (NEC Labs)
Andrew Childs (University of Waterloo, IQC)

arXiv:1103.2774 Quantum rejection sampling
arXiv:1103.3017 Quantum algorithm for the Boolean hidden shift problem

Motivation

Hidden *shift* and *subgroup* problems



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

Boolean hidden shift problem (BHSP)

Problem

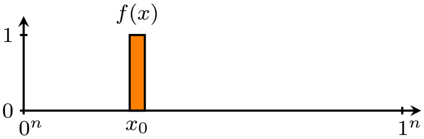
- **Given:** Complete knowledge of $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ and access to a black-box oracle for $f_s(x) := f(x+s)$

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

- **Determine:** The hidden shift s

Delta functions are hard

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



Boolean hidden shift problem (BHSP)

Problem

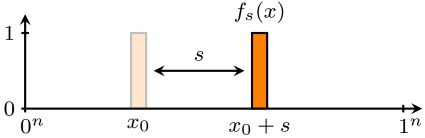
- **Given:** Complete knowledge of $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ and access to a black-box oracle for $f_s(x) := f(x+s)$

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

- **Determine:** The hidden shift s

Delta functions are hard

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



Boolean hidden shift problem (BHSP)

Problem

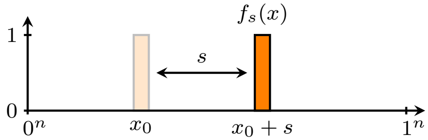
- **Given:** Complete knowledge of $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ and access to a black-box oracle for $f_s(x) := f(x+s)$

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

- **Determine:** The hidden shift s

Delta functions are hard

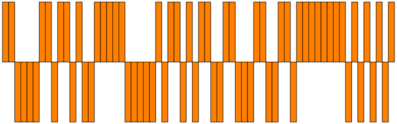
- $f(x) := \delta_{x,x_0}$
- Equivalent to Grover's search: $\Theta(\sqrt{2^n})$



Fourier transform of Boolean functions

The ± 1-function (normalized)

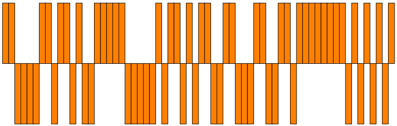
- $F(x) := \frac{1}{\sqrt{2^n}}(-1)^{f(x)}$



Fourier transform of Boolean functions

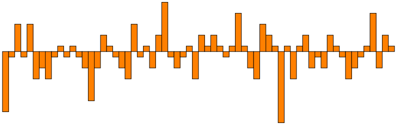
The ± 1-function (normalized)

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



Fourier transform $H := \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

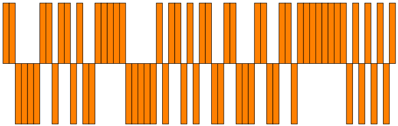
- $\hat{F}(w) := \langle w|H^{\otimes n}|F\rangle$



Fourier transform of Boolean functions

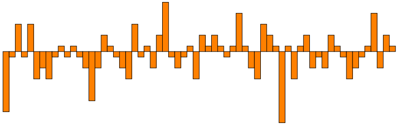
The ± 1-function (normalized)

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



Fourier transform $\quad H := \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

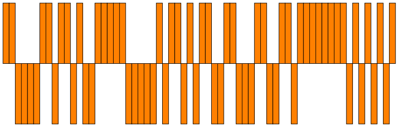
- $\hat{F}(w) := \langle w|H^{\otimes n}|F\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{Z}_2^n} (-1)^{w \cdot x} F(x)$



Fourier transform of Boolean functions

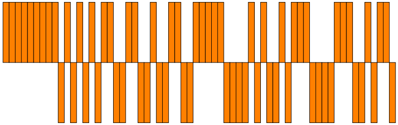
The ± 1-function (normalized)

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



Fourier transform $\quad H := \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

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



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

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

Bent functions are easy

Preparing the "phase state"

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

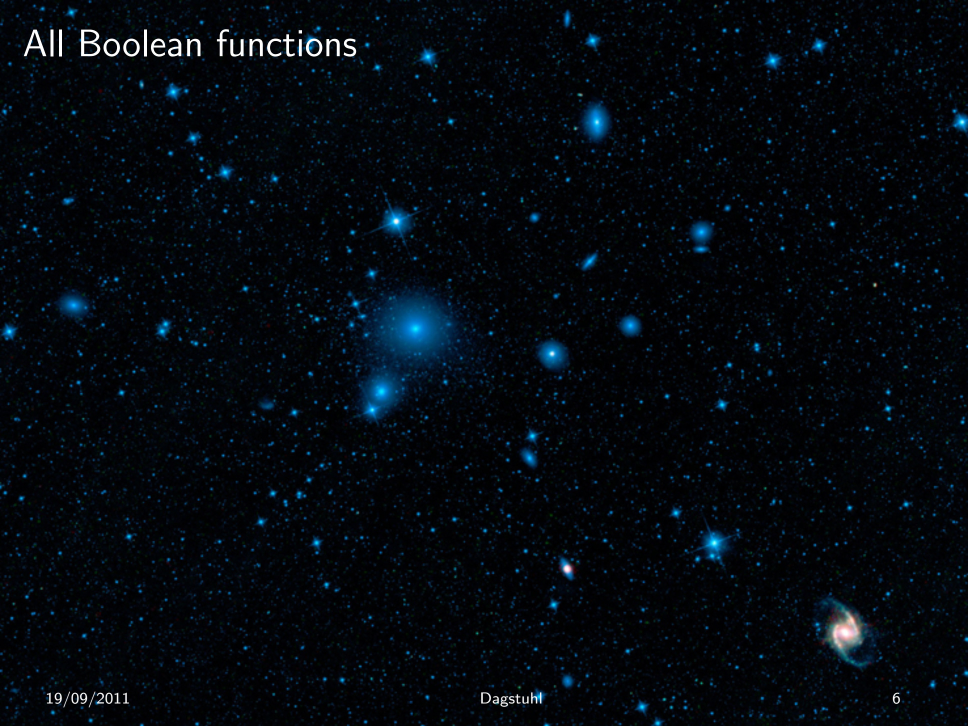
$$|0\rangle^{\otimes n} \; - \boxed{H^{\otimes n}} - \boxed{O_{f_s}} - \boxed{H^{\otimes n}} - \; |\Phi(s)\rangle$$

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

Algorithm [Rötteler'10]

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

All Boolean functions



All Boolean functions

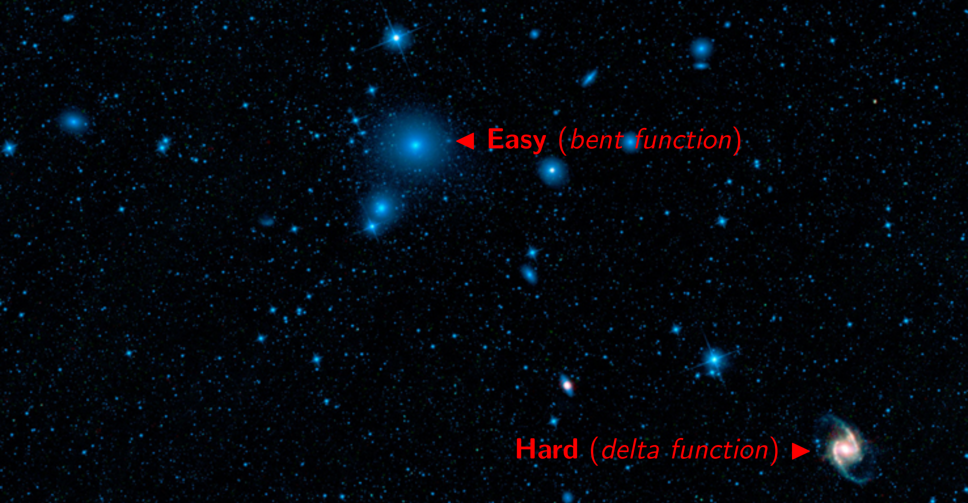
In total there are 2^{2^n} Boolean functions with n arguments.
For $n = 8$ this is roughly 10^{77}.

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}.

What about the rest?

◄ **Easy** (*bent function*)

Hard (*delta function*) ►

All Boolean functions

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

What about the rest?

◀ **Easy** (*bent function*)

Three approaches:

1. Grover-like [Grover'00] / quantum rejection sampling [ORR'11]
2. Pretty good measurement
3. Simon-like [Rötteler'10, GRR'11]

Hard (*delta function*) ▶

Algorithm 1: Grover-like / quantum rejection sampling

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

Algorithm 1: Grover-like / quantum rejection sampling

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

- Pick $\boldsymbol{\varepsilon} \in \mathbb{R}^{2^n}$ such that $\forall w : 0 \leq \varepsilon_w \leq |\hat{F}(w)|$

Algorithm 1: Grover-like / quantum rejection sampling

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

- Pick $\boldsymbol{\varepsilon} \in \mathbb{R}^{2^n}$ such that $\forall w : 0 \leq \varepsilon_w \leq |\hat{F}(w)|$
- Apply $R_{\boldsymbol{\varepsilon}} : |w\rangle|0\rangle \mapsto |w\rangle \frac{1}{\hat{F}(w)} \left(\sqrt{\hat{F}(w)^2 - \varepsilon_w^2}|0\rangle + \varepsilon_w|1\rangle \right)$

Algorithm 1: Grover-like / quantum rejection sampling

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

- Pick $\boldsymbol{\varepsilon} \in \mathbb{R}^{2^n}$ such that $\forall w : 0 \leq \varepsilon_w \leq |\hat{F}(w)|$
- Apply $R_{\boldsymbol{\varepsilon}} : |w\rangle|0\rangle \mapsto |w\rangle \frac{1}{\hat{F}(w)} \Big(\sqrt{\hat{F}(w)^2 - \varepsilon_w^2} |0\rangle + \varepsilon_w |1\rangle \Big)$
- If we would measure the last qubit, we would get outcome "1" w.p. $\|\boldsymbol{\varepsilon}\|_2^2$ and the post-measurement state would be

$$\frac{1}{\|\boldsymbol{\varepsilon}\|_2} \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \varepsilon_w |w\rangle$$

Algorithm 1: Grover-like / quantum rejection sampling

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

- Pick $\boldsymbol{\varepsilon} \in \mathbb{R}^{2^n}$ such that $\forall w : 0 \leq \varepsilon_w \leq |\hat{F}(w)|$
- Apply $R_{\boldsymbol{\varepsilon}} : |w\rangle|0\rangle \mapsto |w\rangle \frac{1}{\hat{F}(w)} \Big(\sqrt{\hat{F}(w)^2 - \varepsilon_w^2} |0\rangle + \varepsilon_w |1\rangle \Big)$
- If we would measure the last qubit, we would get outcome "1" w.p. $\|\boldsymbol{\varepsilon}\|_2^2$ and the post-measurement state would be

$$\frac{1}{\|\boldsymbol{\varepsilon}\|_2} \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \varepsilon_w |w\rangle$$

- Instead of measuring, amplify the amplitude on $|1\rangle$

Algorithm 1: Grover-like / quantum rejection sampling

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

- Pick $\boldsymbol{\varepsilon} \in \mathbb{R}^{2^n}$ such that $\forall w : 0 \leq \varepsilon_w \leq |\hat{F}(w)|$
- Apply $R_{\boldsymbol{\varepsilon}} : |w\rangle|0\rangle \mapsto |w\rangle \frac{1}{\hat{F}(w)} \Big(\sqrt{\hat{F}(w)^2 - \varepsilon_w^2} |0\rangle + \varepsilon_w |1\rangle \Big)$
- If we would measure the last qubit, we would get outcome "1" w.p. $\|\boldsymbol{\varepsilon}\|_2^2$ and the post-measurement state would be

$$\frac{1}{\|\boldsymbol{\varepsilon}\|_2} \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \varepsilon_w |w\rangle$$

- Instead of measuring, amplify the amplitude on $|1\rangle$
- Complexity: $O(1/\|\boldsymbol{\varepsilon}\|_2)$

Algorithm 1: Grover-like / quantum rejection sampling

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

- Pick $\boldsymbol{\varepsilon} \in \mathbb{R}^{2^n}$ such that $\forall w : 0 \leq \varepsilon_w \leq |\hat{F}(w)|$
- Apply $R_{\boldsymbol{\varepsilon}} : |w\rangle|0\rangle \mapsto |w\rangle \frac{1}{\hat{F}(w)} \Big(\sqrt{\hat{F}(w)^2 - \varepsilon_w^2} |0\rangle + \varepsilon_w |1\rangle \Big)$
- If we would measure the last qubit, we would get outcome "1" w.p. $\|\boldsymbol{\varepsilon}\|_2^2$ and the post-measurement state would be

$$\frac{1}{\|\boldsymbol{\varepsilon}\|_2} \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \varepsilon_w |w\rangle$$

- Instead of measuring, amplify the amplitude on $|1\rangle$
- Complexity: $O(1/\|\boldsymbol{\varepsilon}\|_2)$
- Take $\varepsilon_w = \hat{F}_{\min}$ to get s with certainty in $O\Big(\frac{1}{\sqrt{2^n} \hat{F}_{\min}}\Big)$ queries

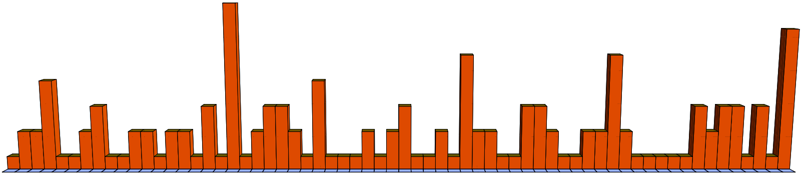
Algorithm 1: "Demo"

Algorithm

Algorithm 1: "Demo"

Algorithm

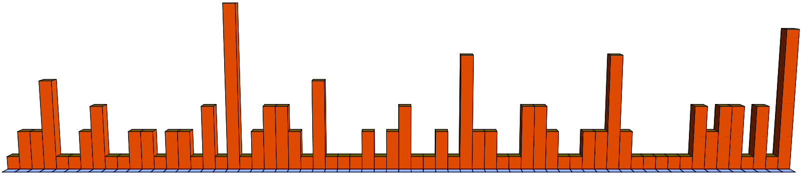
1. Prepare $|\Phi(s)\rangle$



Algorithm 1: "Demo"

Algorithm

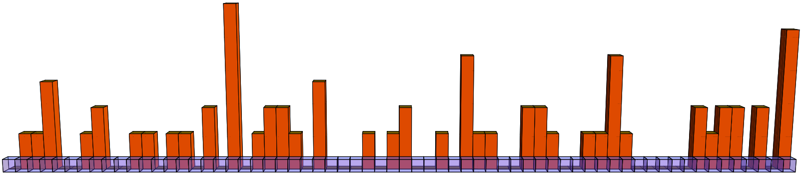
1. Prepare $|\Phi(s)\rangle$
2. Perform an ε-rotation



Algorithm 1: "Demo"

Algorithm

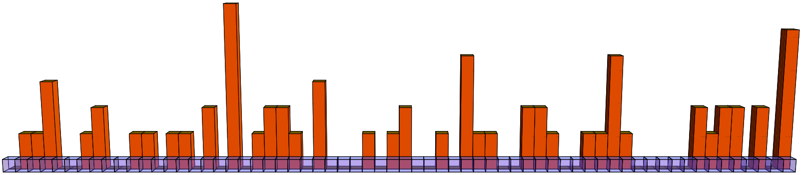
1. Prepare $|\Phi(s)\rangle$
2. Perform an ε-rotation



Algorithm 1: “Demo”

Algorithm

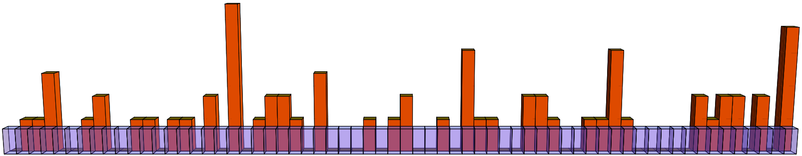
1. Prepare $|\Phi(s)\rangle$
2. Perform an ε-rotation
3. Do amplitude amplification



Algorithm 1: "Demo"

Algorithm

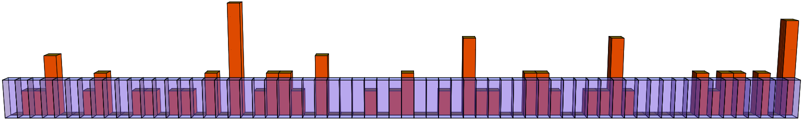
1. Prepare $|\Phi(s)\rangle$
2. Perform an ε-rotation
3. Do amplitude amplification



Algorithm 1: “Demo”

Algorithm

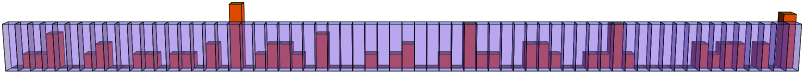
1. Prepare $|\Phi(s)\rangle$
2. Perform an ε-rotation
3. Do amplitude amplification



Algorithm 1: "Demo"

Algorithm

1. Prepare $|\Phi(s)\rangle$
2. Perform an ε-rotation
3. Do amplitude amplification



Algorithm 1: "Demo"

Algorithm

1. Prepare $|\Phi(s)\rangle$
2. Perform an ε-rotation
3. Do amplitude amplification

Algorithm 1: "Demo"

Algorithm

1. Prepare $|\Phi(s)\rangle$
2. Perform an ε-rotation
3. Do amplitude amplification
4. Measure the resulting state in Fourier basis

Algorithm 1: Pros / cons

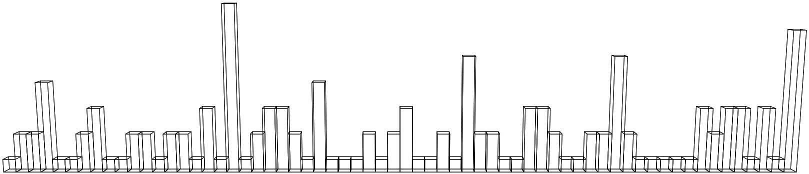
Performance

- Delta functions: $O(\sqrt{2^n})$
- Bent functions: $O(1)$

Issues

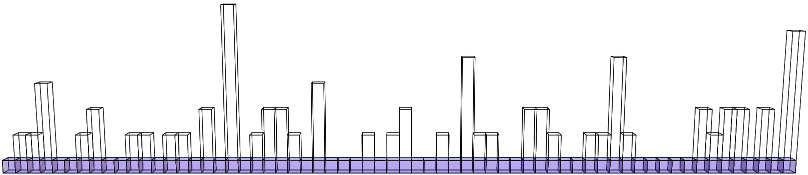
- What if $\hat{F}_{\min} = 0$?
- Undetectable anti-shifts: $f(x+s) = f(x) + 1$

Algorithm 1: Approximate version



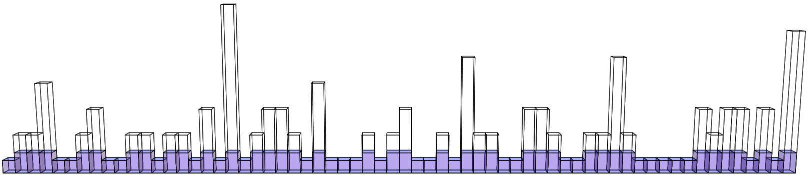
Algorithm 1: Approximate version

- Instead of the flat state



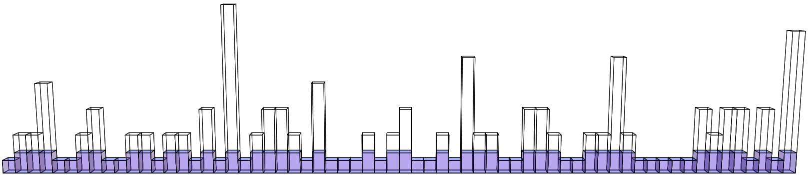
Algorithm 1: Approximate version

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



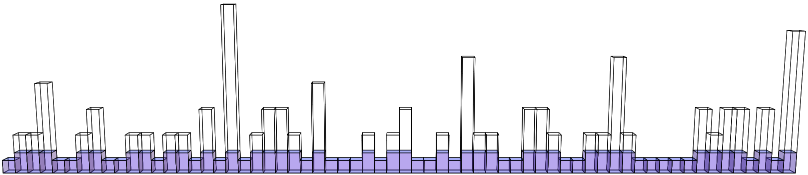
Algorithm 1: Approximate version

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



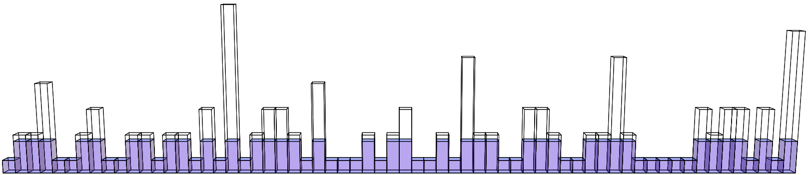
Algorithm 1: Approximate version

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



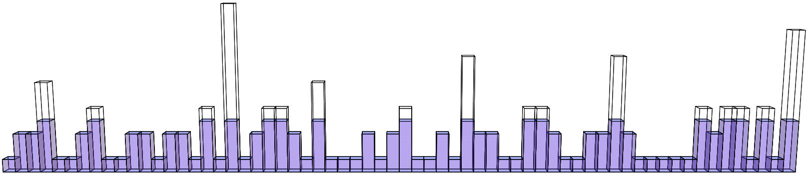
Algorithm 1: Approximate version

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



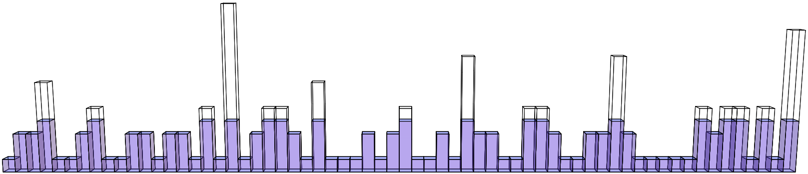
Algorithm 1: Approximate version

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

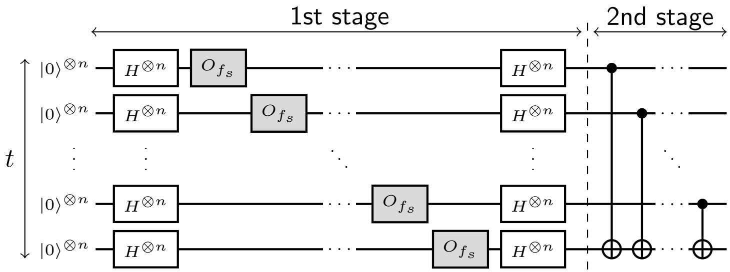


Algorithm 1: Approximate version

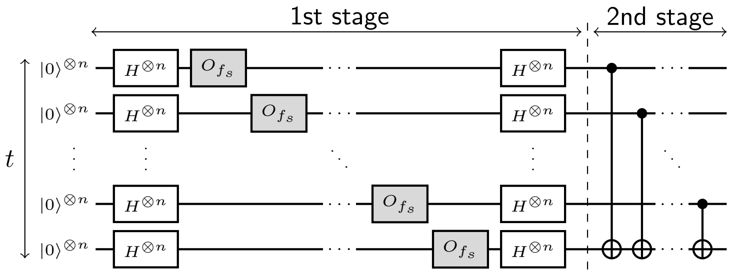
- Instead of the flat state aim for *approximately* flat state
- Fix success probability p
- Optimal choice of ε is given by the "water filling" vector ε_p such that $\boldsymbol{\mu}^{\mathsf{T}} \cdot \boldsymbol{\varepsilon}_p / \|\boldsymbol{\varepsilon}_p\|_2 \geq \sqrt{p}$ where $\mu_w = \frac{1}{\sqrt{2^n}}$
- Queries: $O(1/\|\boldsymbol{\varepsilon}_p\|_2)$



Algorithm 2: Pretty good measurement

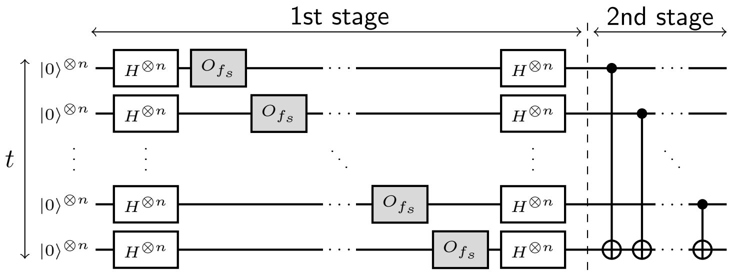


Algorithm 2: Pretty good measurement



After stage 1: $\quad |\Phi(s)\rangle^{\otimes t} = \left(\sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \hat{F}(w) |w\rangle\right)^{\otimes t}$

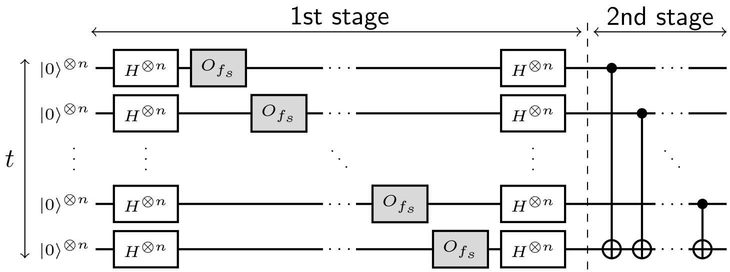
Algorithm 2: Pretty good measurement



After stage 1: $\quad |\Phi(s)\rangle^{\otimes t} = \left(\sum_{w\in\mathbb{Z}_2^n}(-1)^{s\cdot w}\hat{F}(w)|w\rangle\right)^{\otimes t}$

After stage 2: $\quad |\Phi^t(s)\rangle := \sum_{w\in\mathbb{Z}_2^n}(-1)^{s\cdot w}|\mathcal{F}_w^t\rangle|w\rangle$

Algorithm 2: Pretty good measurement

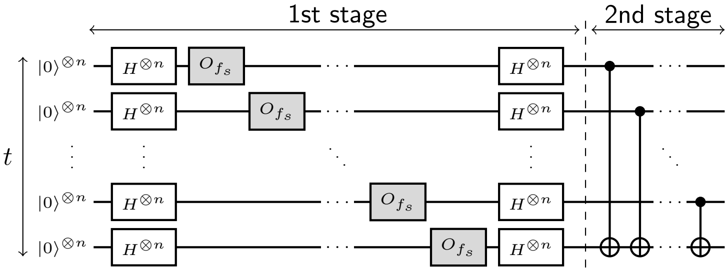


After stage 1: $|\Phi(s)\rangle^{\otimes t} = \left(\sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \hat{F}(w) |w\rangle\right)^{\otimes t}$

After stage 2: $|\Phi^t(s)\rangle := \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} |\mathcal{F}_w^t\rangle |w\rangle$

PGM: $|E_s^t\rangle := \frac{1}{\sqrt{2^n}} \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} \frac{|\mathcal{F}_w^t\rangle}{\||\mathcal{F}_w^t\rangle\|_2} |w\rangle$

Algorithm 2: Pretty good measurement



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

After stage 2: $|\Phi^t(s)\rangle := \sum_{w\in\mathbb{Z}_2^n}(-1)^{s\cdot w}|\mathcal{F}_w^t\rangle|w\rangle$

PGM: $|E_s^t\rangle := \frac{1}{\sqrt{2^n}}\sum_{w\in\mathbb{Z}_2^n}(-1)^{s\cdot w}\frac{|\mathcal{F}_w^t\rangle}{\||\mathcal{F}_w^t\rangle\|_2}|w\rangle$

E.g., for $t=1$: $|E_s^1\rangle := \frac{1}{\sqrt{2^n}}\sum_{w\in\mathbb{Z}_2^n}(-1)^{s\cdot w}\frac{\hat{F}(w)}{|\hat{F}(w)|}|w\rangle$

Algorithm 2: Pretty good measurement

Why does it work?

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

Algorithm 2: Pretty good measurement

Why does it work?

- States: $|\Phi^t(s)\rangle := \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} |\mathcal{F}_w^t\rangle |w\rangle$
  where $\| |\mathcal{F}_w^t\rangle \|_2^2 = \left[\hat{F}^2\right]^{*t}(w) = \frac{1}{\sqrt{2^n}} \widehat{(F * F)^t}(w)$

Algorithm 2: Pretty good measurement

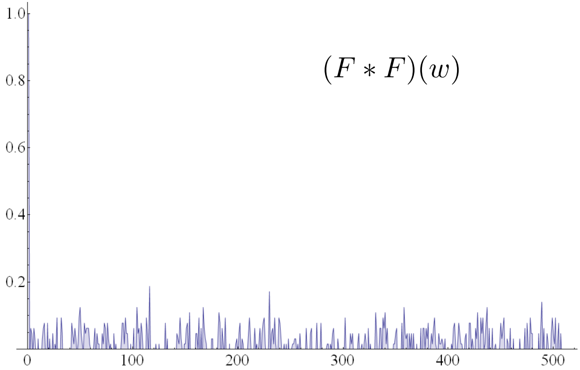
Why does it work?

- States: $|\Phi^t(s)\rangle := \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} |\mathcal{F}_w^t\rangle |w\rangle$
  where $\| |\mathcal{F}_w^t\rangle \|_2^2 = \left[\hat{F}^2\right]^{*t}(w) = \frac{1}{\sqrt{2^n}} \widehat{(F * F)^t}(w)$
- Convolution: $(F * F)(w) = \sum_{x \in \mathbb{Z}_2^n} F(x) F(w - x)$

Algorithm 2: Pretty good measurement

Why does it work?

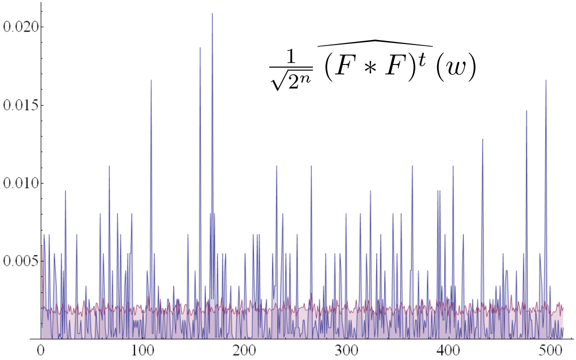
- States: $|\Phi^t(s)\rangle := \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} |\mathcal{F}_w^t\rangle |w\rangle$
  where $\| |\mathcal{F}_w^t\rangle \|_2^2 = \left[\hat{F}^2\right]^{*t}(w) = \frac{1}{\sqrt{2^n}} \widehat{(F * F)^t}(w)$
- Convolution: $(F * F)(w) = \sum_{x \in \mathbb{Z}_2^n} F(x) F(w - x)$



Algorithm 2: Pretty good measurement

Why does it work?

- States: $|\Phi^t(s)\rangle := \sum_{w \in \mathbb{Z}_2^n} (-1)^{s \cdot w} |\mathcal{F}_w^t\rangle |w\rangle$
  where $\| |\mathcal{F}_w^t\rangle \|_2^2 = \left[\hat{F}^2\right]^{*t}(w) = \frac{1}{\sqrt{2^n}} \widehat{(F * F)^t}(w)$
- Convolution: $(F * F)(w) = \sum_{x \in \mathbb{Z}_2^n} F(x) F(w - x)$



Algorithm 2: Pros / cons

Performance

- Bent functions: $O(1)$
- Random functions: $O(1)$
- No issues with undetectable anti-shifts

Issues

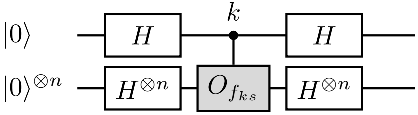
- Delta functions: $O(2^n)$, no speedup

Note

- For some $t \leq n$ there will be no zero amplitudes!

Algorithm 3: Simon-like

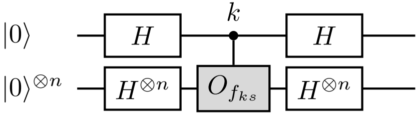
- Oracle $O_{f_{ks}}$: $|k\rangle|w\rangle \mapsto (-1)^{f(x+ks)}|k\rangle|w\rangle$



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

Algorithm 3: Simon-like

- Oracle $O_{f_{ks}} : |k\rangle|w\rangle \mapsto (-1)^{f(x+ks)}|k\rangle|w\rangle$

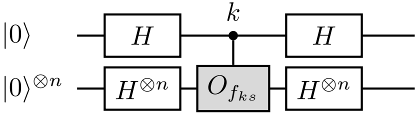


$$|\Psi(s)\rangle := \sum_{w \in \mathbb{Z}_2^n} \hat{F}(w)|s \cdot w\rangle|w\rangle$$

- Complexity: $O(n/\sqrt{I_f})$

Algorithm 3: Simon-like

- Oracle $O_{f_{ks}} : |k\rangle|w\rangle \mapsto (-1)^{f(x+ks)}|k\rangle|w\rangle$



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

- Complexity: $O(n/\sqrt{I_f})$
- Where $I_f(w)$ is the *influence* of $w \in \mathbb{Z}_2^n$ on f:

$$I_f(w) := \Pr_x\left[f(x) \neq f(x+w)\right]$$

and $I_f := \min_w I_f(w)$

Comparison

	delta	bent	random
Grover-like	$O(\sqrt{2^n})$	$O(1)$	$O(1)$
PGM	$O(2^n)$	$O(1)$	$O(1)$
Simon-like	$O(n\sqrt{2^n})$	$O(n)$	$O(n)$

Open problems

- What is the best quantum algorithm for solving BHSP?

Open problems

- What is the best quantum algorithm for solving BHSP?
- Quantum query lower bound?

Open problems

- What is the best quantum algorithm for solving BHSP?
- Quantum query lower bound?
- Related problems:

Open problems

- What is the best quantum algorithm for solving BHSP?
- Quantum query lower bound?
- Related problems:
  - Verification of s: $O\left(1/\sqrt{I_f}\right)$

Open problems

- What is the best quantum algorithm for solving BHSP?
- Quantum query lower bound?
- Related problems:
  - Verification of s: $O\big(1/\sqrt{I_f}\big)$
  - Extracting parity $w \cdot s$: $O\big(1/\hat{F}(w)\big)$

Open problems

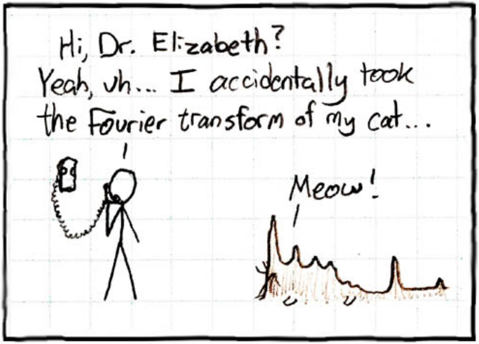
- What is the best quantum algorithm for solving BHSP?
- Quantum query lower bound?
- Related problems:
  - Verification of s: $O\big(1/\sqrt{I_f}\big)$
  - Extracting parity $w \cdot s$: $O\big(1/\hat{F}(w)\big)$
- What is the classical query complexity of this problem?

Open problems

- What is the best quantum algorithm for solving BHSP?
- Quantum query lower bound?
- Related problems:
  - Verification of s: $O\big(1/\sqrt{I_f}\big)$
  - Extracting parity $w \cdot s$: $O\big(1/\hat{F}(w)\big)$
- What is the classical query complexity of this problem?
- Generalize from $\mathbb{Z}_2$ to $\mathbb{Z}_d$

Open problems

- What is the best quantum algorithm for solving BHSP?
- Quantum query lower bound?
- Related problems:
  - Verification of s: $O\big(1/\sqrt{I_f}\big)$
  - Extracting parity $w \cdot s$: $O\big(1/\hat{F}(w)\big)$
- What is the classical query complexity of this problem?
- Generalize from $\mathbb{Z}_2$ to $\mathbb{Z}_d$
- Applications



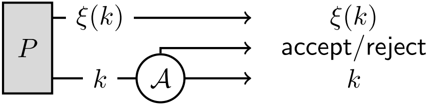
Thank you for your attention!

Classical rejection sampling

Classical resampling problem

- **Given:** Ability to sample from distribution p
- **Task:** Sample from distribution q

Classical algorithm



Quantum rejection sampling

Quantum resampling problem

- **Given:** Oracle $O : |0\rangle \mapsto \sum_{k=1}^{n} \pi_k |\xi_k\rangle |k\rangle$
- **Task:** Perform transformation

$$\sum_{k=1}^{n} \pi_k |\xi_k\rangle |k\rangle \mapsto \sum_{k=1}^{n} \sigma_k |\xi_k\rangle |k\rangle$$

- **Note:** Amplitudes π_k and σ_k are known, but states $|\xi_k\rangle$ are not known