Search via quantum walk

Jérémie Roland

UC Berkeley

Joint work with

Frédéric Magniez[1] Ashwin Nayak[2] Miklos Santha[1]

[1]LRI-CNRS, France

[2]Univ. Waterloo/Perimeter Institute, Canada

STOC 2007

Abstract search problem

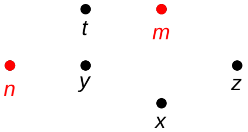
The problem

Input:

- a set of elements X
- a set of marked elements $M \subseteq X$ $\left(\varepsilon = \frac{|M|}{|X|}\right)$

Output:

- a marked element $x \in M$



Available procedures

- Setup (cost S): pick a random $x \in X$
- Check (cost C): check whether $x \in M$
- Update (cost U): make a random walk P

- here: assume P ergodic, symmetric
- δ = e-v gap of P

Abstract search problem

The problem

Input:

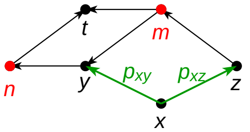
- a set of elements X
- a set of marked elements $M \subseteq X$ $\left(\varepsilon = \frac{|M|}{|X|}\right)$

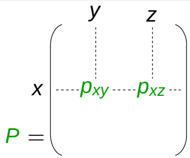
Output:

- a marked element $x \in M$

Available procedures

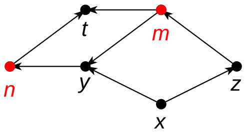
- Setup (cost S): pick a random $x \in X$
- Check (cost C): check whether $x \in M$
- Update (cost U): make a random walk P





- here: assume P ergodic, symmetric
- $\delta =$ e-v gap of P

Three (classical) search algorithms



Naive algorithm

Repeat $T_1\times$ $\left(T_1 = O(\frac{1}{\varepsilon})\right)$

- Pick random $x \in X$ (S)
- Check whether $x \in M$ (C)

Cost: $\frac{1}{\varepsilon}(\mathrm{S}+\mathrm{C})$

Idea: Use random walk!

$T_2\times$ random walk $\left(T_2 = O(\frac{1}{\delta})\right)$

$\approx$ pick random x

Random walk 1

- Pick random $x \in X$ (S)
- Repeat $T_1\times$
  - Check whether $x \in M$ (C)
  - Repeat $T_2\times$
    - Random walk (U)

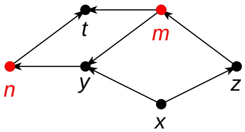
Cost: $\mathrm{S}+\frac{1}{\varepsilon}(\frac{1}{\delta}\mathrm{U}+\mathrm{C})$

Random walk 2

- Pick random $x \in X$ (S)
- Repeat $T_1 T_2\times$
  - Check whether $x \in M$ (C)
  - Random walk (U)

Cost: $\mathrm{S}+\frac{1}{\varepsilon\delta}(\mathrm{U}+\mathrm{C})$

Three (classical) search algorithms



Naive algorithm

Repeat $T_1 \times$ $\left(T_1 = O(\frac{1}{\varepsilon})\right)$

- Pick random $x \in X$ (S)
- Check whether $x \in M$ (C)

Cost: $\frac{1}{\varepsilon}(\mathrm{S} + \mathrm{C})$

Idea: Use random walk!

$T_2 \times$ random walk $\left(T_2 = O(\frac{1}{\delta})\right)$

$\approx$ pick random x

Random walk 1

- Pick random $x \in X$ (S)
- Repeat $T_1 \times$
  - Check whether $x \in M$ (C)
  - Repeat $T_2 \times$
    - Random walk (U)

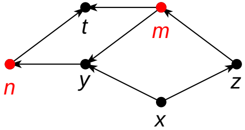
Cost: $\mathrm{S} + \frac{1}{\varepsilon}(\frac{1}{\delta}\mathrm{U} + \mathrm{C})$

Random walk 2

- Pick random $x \in X$ (S)
- Repeat $T_1 T_2 \times$
  - Check whether $x \in M$ (C)
  - Random walk (U)

Cost: $\mathrm{S} + \frac{1}{\varepsilon\delta}(\mathrm{U} + \mathrm{C})$

Three (classical) search algorithms



Naive algorithm

Repeat $T_1\times$ $\left(T_1 = O(\frac{1}{\varepsilon})\right)$

- Pick random $x \in X$ (S)
- Check whether $x \in M$ (C)

Cost: $\frac{1}{\varepsilon}(S + C)$

Idea: Use random walk!

$T_2\times$ random walk $\left(T_2 = O(\frac{1}{\delta})\right)$

$\approx$ pick random x

Random walk 1

- Pick random $x \in X$ (S)
- Repeat $T_1\times$
  - Check whether $x \in M$ (C)
  - Repeat $T_2\times$
    - Random walk (U)

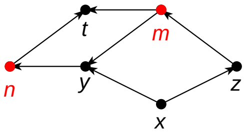
Cost: $S + \frac{1}{\varepsilon}(\frac{1}{\delta}U + C)$

Random walk 2

- Pick random $x \in X$ (S)
- Repeat $T_1 T_2\times$
  - Check whether $x \in M$ (C)
  - Random walk (U)

Cost: $S + \frac{1}{\varepsilon\delta}(U + C)$

Three (classical) search algorithms



Naive algorithm

Repeat $T_1\times$ $\left(T_1 = O(\frac{1}{\varepsilon})\right)$

- Pick random $x \in X$ (S)
- Check whether $x \in M$ (C)

Cost: $\frac{1}{\varepsilon}(\mathrm{S}+\mathrm{C})$

Idea: Use random walk!

$T_2\times$ random walk $\left(T_2 = O(\frac{1}{\delta})\right)$
$\approx$ pick random x

Random walk 1

- Pick random $x \in X$ (S)
- Repeat $T_1\times$
  - Check whether $x \in M$ (C)
  - Repeat $T_2\times$
    - Random walk (U)

Cost: $\mathrm{S}+\frac{1}{\varepsilon}(\frac{1}{\delta}\mathrm{U}+\mathrm{C})$

Random walk 2

- Pick random $x \in X$ (S)
- Repeat $T_1 T_2\times$
  - Check whether $x \in M$ (C)
  - Random walk (U)

Cost: $\mathrm{S}+\frac{1}{\varepsilon\delta}(\mathrm{U}+\mathrm{C})$

Quantum search problem

Two related problems

Input:

- a set of elements X
- a set of marked elements $M \subseteq X$

Output:

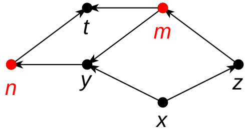
1. Find a marked element $x \in M$
2. Detect whether there is a marked element ($M = \emptyset$?)

Available procedures

- Setup (cost S): prepare $|\pi\rangle = \frac{1}{\sqrt{|X|}} \sum_x |x\rangle$
- Check (cost C): reflection / marked elements

$$\mathrm{ref}_M : |x\rangle \mapsto \begin{cases} |x\rangle & \text{if } x \in M \\ -|x\rangle & \text{otherwise} \end{cases}$$

- Update (cost U): apply quantum walk W



Quantum search problem

Two related problems

Input:

- a set of elements X
- a set of marked elements $M \subseteq X$

Output:

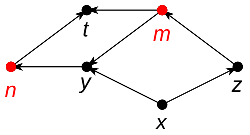
1. Find a marked element $x \in M$
2. Detect whether there is a marked element ($M = \emptyset$?)

Available procedures

- Setup (cost S): prepare $|\pi\rangle = \frac{1}{\sqrt{|X|}} \sum_x |x\rangle$
- Check (cost C): reflection / marked elements

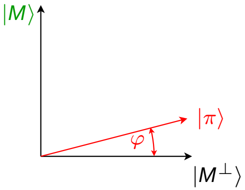
$$\mathrm{ref}_M : |x\rangle \mapsto \begin{cases} |x\rangle & \text{if } x \in M \\ -|x\rangle & \text{otherwise} \end{cases}$$

- Update (cost U): apply quantum walk W



Grover's algorithm

- We start with $|\pi\rangle = \frac{1}{\sqrt{|X|}} \sum_{x \in X} |x\rangle$
- Goal: prepare $|M\rangle = \frac{1}{\sqrt{|M|}} \sum_{x \in M} |x\rangle$
- We use 2 reflections:
  - through $|M^\perp\rangle$: $\mathrm{ref}_{M^\perp} = -\mathrm{ref}_M$ (C)
  - through $|\pi\rangle$: ref_π (S)



Grover's algorithm

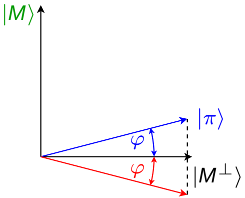
- Prepare $|\pi\rangle$ (S)
- Repeat $T_1 \times$
  - apply $\mathrm{ref}_{M^\perp}$ (C)
  - apply ref_π (S)

Cost: $T_1(\mathrm{S} + \mathrm{C})$

$$\sin\varphi = \langle M|\pi\rangle = \sqrt{\frac{|M|}{|X|}} = \sqrt{\varepsilon}$$

Grover's algorithm

- We start with $|\pi\rangle = \frac{1}{\sqrt{|X|}} \sum_{x \in X} |x\rangle$
- Goal: prepare $|M\rangle = \frac{1}{\sqrt{|M|}} \sum_{x \in M} |x\rangle$
- We use 2 reflections:
  - through $|M^\perp\rangle$: $\mathrm{ref}_{M^\perp} = -\mathrm{ref}_M$ (C)
  - through $|\pi\rangle$: ref_π (S)



$$\sin\varphi = \langle M|\pi\rangle = \sqrt{\frac{|M|}{|X|}} = \sqrt{\varepsilon}$$

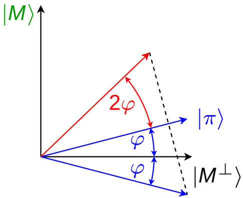
Grover's algorithm

- Prepare $|\pi\rangle$ (S)
- Repeat $T_1 \times$
  - apply $\mathrm{ref}_{M^\perp}$ (C)
  - apply ref_π (S)

Cost: $T_1(\mathrm{S} + \mathrm{C})$

Grover's algorithm

- We start with $|\pi\rangle = \frac{1}{\sqrt{|X|}} \sum_{x \in X} |x\rangle$
- Goal: prepare $|M\rangle = \frac{1}{\sqrt{|M|}} \sum_{x \in M} |x\rangle$
- We use 2 reflections:
  - through $|M^\perp\rangle$: $\mathrm{ref}_{M^\perp} = -\mathrm{ref}_M$ (C)
  - through $|\pi\rangle$: ref_π (S)



$$\sin\varphi = \langle M|\pi\rangle = \sqrt{\frac{|M|}{|X|}} = \sqrt{\varepsilon}$$

Grover's algorithm

- Prepare $|\pi\rangle$ (S)
- Repeat $T_1 \times$
  - apply $\mathrm{ref}_{M^\perp}$ (C)
  - apply ref_π (S)

Cost: $\frac{1}{\sqrt{\varepsilon}}(S + C)$

Grover's algorithm: Comments

- Quantum analogue of the *naive* algorithm "pick and check".
- $\frac{1}{\sqrt{\varepsilon}}(\mathsf{S}+\mathsf{C})$ vs $\frac{1}{\varepsilon}(\mathsf{S}+\mathsf{C}) \Longrightarrow$ Grover's quadratic speed-up

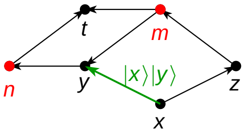
What if S is high?

$\Longrightarrow$ Replace ref_π by some *quantum walk* W!

From random to quantum walks [Szegedy'04]

Quantum walk $W(P)$:

- State space: Pairs of neighbours $|x\rangle|y\rangle \Longrightarrow$ Walk on edges (x, y)
- Two steps
  - Diffusion of y over the neighbours of x
  - Diffusion of x over the neighbours of y



We use diffusions *à la Grover*, i.e., reflections

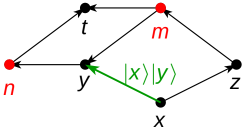
- Superposition over neighbours of x: $|p_x\rangle = \sum_y \sqrt{p_{yx}}|y\rangle$
- $\mathrm{ref}_{\mathcal{X}}$: reflection through subspace $\mathcal{X} = \{|x\rangle|p_x\rangle : x \in X\}$
- Similarly for $\mathcal{Y}$

We define the quantum walk W as

$$W = \mathrm{ref}_{\mathcal{Y}} \cdot \mathrm{ref}_{\mathcal{X}}$$

Quantum walk $W(P)$:

- State space: Pairs of neighbours $|x\rangle|y\rangle \implies$ Walk on edges (x, y)
- Two steps
  - Diffusion of y over the neighbours of x
  - Diffusion of x over the neighbours of y



We use diffusions *à la Grover*, i.e., reflections

- Superposition over neighbours of x: $|p_x\rangle = \sum_y \sqrt{p_{yx}}|y\rangle$
- $\mathrm{ref}_{\mathcal{X}}$: reflection through subspace $\mathcal{X} = \{|x\rangle|p_x\rangle : x \in X\}$
- Similarly for $\mathcal{Y}$

We define the quantum walk W as

$$W = \mathrm{ref}_{\mathcal{Y}} \cdot \mathrm{ref}_{\mathcal{X}}$$

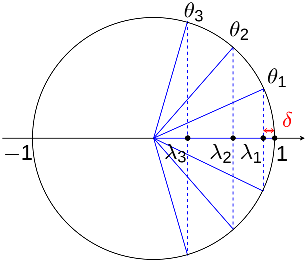
From random to quantum walks [Szegedy'04]

Random walk

- $P = (p_{xy})$
- E-v: $\lambda_k = \cos\theta_k$
- Stationary dist. $(\cos\theta_0 = 1)$:

$$\pi = (\pi_x)$$

- E-v gap: $\delta = 1 - |\cos\theta_1|$



Quantum walk

- $W = \mathrm{ref}_{\mathcal{Y}} \cdot \mathrm{ref}_{\mathcal{X}}$
- E-v (on $\mathcal{X} \oplus \mathcal{Y}$): $e^{\pm 2i\theta_k}$
- Stationary state $(\theta_0 = 0)$:

$$|\pi\rangle = \sum_x \sqrt{\pi_x} |x\rangle |p_x\rangle$$

- phase gap: $\Delta = |2\theta_1|$

From random to quantum walks [Szegedy'04]

Random walk

- $P = (p_{xy})$
- E-v: $\lambda_k = \cos\theta_k$
- Stationary dist. ($\cos\theta_0 = 1$):

$$\pi = (\pi_x)$$

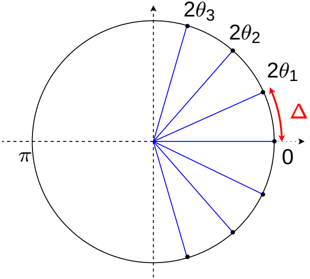
- E-v gap: $\delta = 1 - |\cos\theta_1|$

Quantum walk

- $W = \mathrm{ref}_{\mathcal{Y}} \cdot \mathrm{ref}_{\mathcal{X}}$
- E-v (on $\mathcal{X} \oplus \mathcal{Y}$): $e^{\pm 2i\theta_k}$
- Stationary state ($\theta_0 = 0$):

$$|\pi\rangle = \sum_x \sqrt{\pi_x}|x\rangle|p_x\rangle$$

- phase gap: $\Delta = |2\theta_1|$



$$\Delta = O(\sqrt{\delta})$$
$$\Downarrow$$

quantum phase gap
$= O\left(\sqrt{\text{classical e-v gap}}\right)$

Quantum walk for element distinctness [Ambainis'04]

IDEA:

- Replace ref_π by W^{T_2} in Grover's algorithm
- with $T_2 = O(\frac{1}{\Delta}) = O(\frac{1}{\sqrt{\delta}})$

Works under some assumptions:

- Johnson graphs (element distinctness)
- Unique solution (classical reduction to this case)

Properties:

- Finds a marked element
- Cost

$$S + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}U + C\right)$$

Notable application:

- Triangle Finding [Magniez, Santha, Szegedy'05]

Quantum walk for element distinctness [Ambainis'04]

IDEA:

- Replace ref_π by W^{T_2} in Grover's algorithm
- with $T_2 = O(\frac{1}{\Delta}) = O(\frac{1}{\sqrt{\delta}})$

Works under some assumptions:

- Johnson graphs (element distinctness)
- Unique solution (classical reduction to this case)

Properties:

- Finds a marked element
- Cost

$$\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathsf{U} + \mathsf{C}\right)$$

Notable application:

- Triangle Finding [Magniez, Santha, Szegedy'05]

- Quantum analogue of the marked random walk: Check if $x \in M$
  - If so: Stay in x
  - Otherwise: Apply random walk P
- Works for any symmetric, ergodic Markov chain
- Cost

$$S + \frac{1}{\sqrt{\varepsilon\delta}}(U + C)$$

- Detects marked items, but does not find them!
- Notable applications
  - Matrix Product Verification [Buhrman, Špalek'06]
  - Group Commutativity [Magniez, Nayak'05]

Szegedy's algorithm [Szegedy'04]

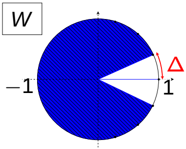
- Quantum analogue of the marked random walk:
  Check if $x \in M$
  - If so: Stay in x
  - Otherwise: Apply random walk P
- Works for any symmetric, ergodic Markov chain
- Cost

$$\mathsf{S} + \frac{1}{\sqrt{\varepsilon\delta}}(\mathsf{U} + \mathsf{C})$$

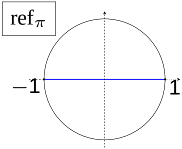
- Detects marked items, but does not find them!
- Notable applications
  - Matrix Product Verification [Buhrman, Špalek'06]
  - Group Commutativity [Magniez, Nayak'05]

Our idea

IDEA: Using W, simulate ref_π to use Grover's algorithm



- $W|\pi\rangle = |\pi\rangle$
- $W|\psi_k\rangle = e^{i\theta_k}|\psi_k\rangle$



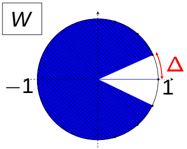
- $\mathrm{ref}_\pi|\pi\rangle = |\pi\rangle$
- $\mathrm{ref}_\pi|\psi_k\rangle = -|\psi_k\rangle$

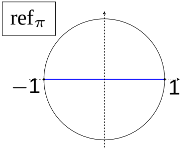
We need a procedure to discriminate between eigenstates

- $|\psi_k\rangle$ with $|\theta_k| \geq \Delta$
- $|\pi\rangle$ with $\theta_0 = 0$.

Our idea

IDEA: Using W, simulate ref_π to use Grover's algorithm





- $W|\pi\rangle = |\pi\rangle$
- $W|\psi_k\rangle = e^{i\theta_k}|\psi_k\rangle$

- $\mathrm{ref}_\pi|\pi\rangle = |\pi\rangle$
- $\mathrm{ref}_\pi|\psi_k\rangle = -|\psi_k\rangle$

We need a procedure to discriminate between eigenstates

- $|\psi_k\rangle$ with $|\theta_k| \geq \Delta$
- $|\pi\rangle$ with $\theta_0 = 0$.

Quantum phase estimation [Kitaev'95, Cleve *et al*'98]

- Let W be a unitary operator such that $W|\psi_k\rangle = e^{i\theta_k}|\psi_k\rangle$
- Quantum phase estimation $E(W)$ is a quantum circuit such that

$$E(W) : |\psi_k\rangle|0\rangle \mapsto |\psi_k\rangle \left[|\tilde{\theta}_k\rangle + |\mathrm{error}\rangle\right]$$

  where $\tilde{\theta}_k$ is an approximation of θ_k
- $|\tilde{\theta}_k - \theta_k| < \Delta$ requires $O(\frac{1}{\Delta})$ calls to W

We may now simulate ref_π:

- do phase estimation $E(W)$
- flip sign if $\tilde{\theta}_k \neq 0$
- undo phase estimation $E(W)$

Quantum phase estimation [Kitaev'95, Cleve *et al*'98]

- Let W be a unitary operator such that $W|\psi_k\rangle = e^{i\theta_k}|\psi_k\rangle$
- Quantum phase estimation $E(W)$ is a quantum circuit such that

$$E(W) : |\psi_k\rangle|0\rangle \mapsto |\psi_k\rangle \left[|\tilde{\theta}_k\rangle + |\mathrm{error}\rangle\right]$$

  where $\tilde{\theta}_k$ is an approximation of θ_k
- $|\tilde{\theta}_k - \theta_k| < \Delta$ requires $O(\frac{1}{\Delta})$ calls to W

We may now simulate ref_π:

- do phase estimation $E(W)$
- flip sign if $\tilde{\theta}_k \neq 0$
- undo phase estimation $E(W)$

Our quantum walk algorithm

Using this reflection in Grover's algorithm, we obtain

- Search algorithm via quantum walk
- from any irreducible Markov chain
- Cost:

$$\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathsf{U} + \mathsf{C}\right)$$

- finds marked elements
- $|\mathrm{error}\rangle$ due to imperfect phase estimation handled with a recursive search algorithm *à la* [Høyer, Mosca, de Wolf'04]

Applications

- Polylog factor improvement for Triangle Finding [MSS'05]
- Unified framework for this and
  - Element Distinctness [A'04]
  - Matrix Product Verification [BŠ'06]
  - Group Commutativity [MN'05]
- Better algorithms for applications in which checking cost is higher than update cost
- New application: Semigroup Problem [DT'07]
- Open: Spatial search [AA'05,AKR'05,CG'04,S'04]