

Quantum algorithms based on quantum walks

Jérémie Roland



Université Libre de Bruxelles



Quantum Information & Communication

- Preliminaries

- ▶ Classical random walks
- ▶ Three search algorithms based on random walks

- Search algorithms based on quantum walks

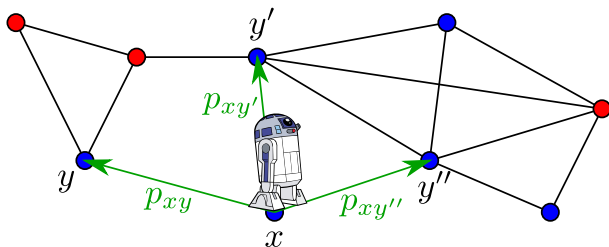
- ▶ From random to quantum walks [Szegedy'04]
- ▶ Grover's algorithm: Complete graph [Grover'95]
- ▶ Element Distinctness: Johnson graph [Ambainis'04]
- ▶ Generalized search algorithm via quantum walk [Magniez,Nayak,Roland,Santha'07]
- ▶ Quantum hitting time: Detecting vs finding [Szegedy'04],[Krovi,Magniez,Ozols,Roland'10]

- Other algorithms based on quantum walks

- ▶ Exponential speed-up: "Glued trees" [Childs,Cleve,Deotto,Farhi,Gutmann,Spielman'03]
- ▶ Scattering-based algorithms: Formula evaluation [Farhi,Goldstone,Gutmann'07]
- ▶ Universal quantum computation by quantum walks [Childs'09,Childs,Gossett,Webb'12]

- Conclusion

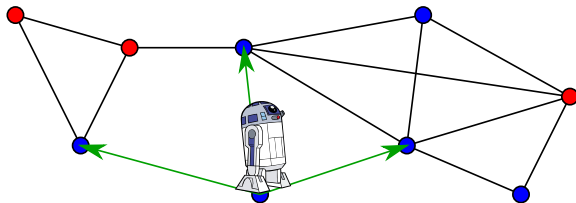
Random walk on a graph



Stochastic matrix $P = (p_{xy})$

- $p_{xy} \neq 0$ only if (x, y) is an edge
- Eigenvalues $1 = \lambda_0 > \lambda_1 \geq \dots \geq \lambda_{n-1} > -1$ (Assume P ergodic, symmetric)
- Stationary distribution: $\pi P = \pi$ (π uniform if P symmetric)
- Eigenvalue gap $\delta = 1 - \lambda_1$

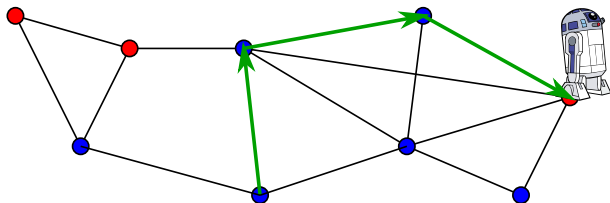
Mixing time



Definition: Mixing time

- Mixing time $MT(P)$: Number of steps necessary to approach π
- $MT(P) \leq \frac{1}{\delta}$, where $\delta = 1 - \lambda_1$ is the eigenvalue gap

Hitting time



Definition: Hitting time

- Let M be a set of marked vertices
- Assume we start from a random vertex $x \sim \pi$
- Hitting time $\text{HT}(P, M)$: Expected number of steps to reach $m \in M$
- $\text{HT}(P, M) \leq \frac{1}{\varepsilon \delta}$, where $\varepsilon = \frac{|M|}{|X|}$

Abstract search problem (classical)

The problem

Input:

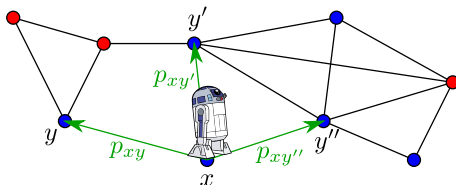
- a set of **elements** X
- with unknown subset of **marked** elements $M \subseteq X$
($\varepsilon = \frac{|M|}{|X|}$)

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



Three (classical) search algorithms

Naive algorithm

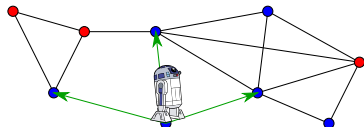
Repeat $(\frac{1}{\epsilon}) \times$

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

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

Idea: Use random walk!

MT \times random walk \approx pick random x
(mixing time $MT \leq \frac{1}{\delta}$)



Random walk I

- Pick random $x \in X$ (S)
- Repeat $(\frac{1}{\epsilon}) \times$
 - ▶ Check whether $x \in M$ (C)
 - ▶ Repeat $(\frac{1}{\delta}) \times$
 - ★ Random walk (U)

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

Random walk II

- Pick random $x \in X$ (S)
- Repeat HT \times (hitting time $HT \leq \frac{1}{\delta}$)
 - ▶ Check whether $x \in M$ (C)
 - ▶ Random walk (U)

Cost: $S + HT(U + C)$

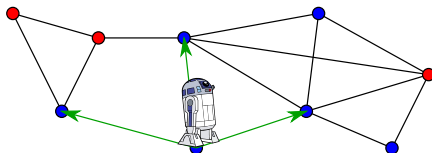
Observations

- Naive algorithm

- ▶ Samples a new independent vertex at each step
- ⇒ Equivalent to walk on complete graph with $U = S$
- ▶ $p_{xy} = \frac{1}{n}$ for all x, y
- ▶ Eigenvalues: $\lambda_0 = 1$ and $\lambda_i = 0$ for all $i \neq 0$

- Random walk I

- ▶ Repetitions of an ergodic walk approximates the walk on the complete graph
- ▶ Mathematically: For large T , $\lambda_i^T \rightarrow 0$ whenever $|\lambda_i| < 1$



Abstract search problem (quantum)

Two related problems

Input:

- a set of **elements** X
- with unknown subset of **marked** elements $M \subseteq X$

Output:

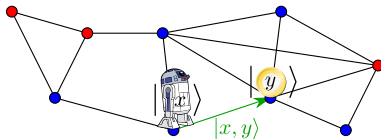
- 1 **Find** a marked element $x \in M$
- 2 **Detect** whether there is a marked element ($M = \emptyset?$)

Available procedures

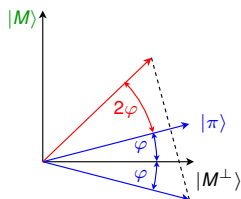
- **Setup** (cost **S**):
prepare $|\pi\rangle = \frac{1}{\sqrt{|X|}} \sum_x |x\rangle$
- **Check** (cost **C**):
reflection / marked elements

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

- **Update** (cost **U**):
apply **quantum walk** W



- 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 $|M^\perp\rangle$: $\text{ref}_{M^\perp} = -\text{ref}_M$ (C)
 - ▶ through $|\pi\rangle$: ref_π (S)



Grover's algorithm

- Prepare $|\pi\rangle$ (S)
- Repeat $T \times$
 - ▶ apply ref_{M^\perp} (C)
 - ▶ apply ref_π (S)

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

$$\begin{aligned} \sin \varphi &= \langle M | \pi \rangle \\ &= \sqrt{\frac{|M|}{|X|}} \\ &= \sqrt{\epsilon} \end{aligned}$$

Grover's algorithm: Observations

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

What if **S** is high?

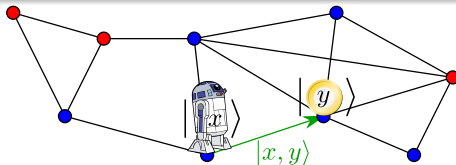
\implies Replace ref_{π} by some **quantum walk** W !

Random walk P on edges (x, y)

- Acts on two registers: position x and coin y
- Walk in two steps:
 - ▶ Flip the coin y over the neighbours of x
 - ▶ Swap x and y

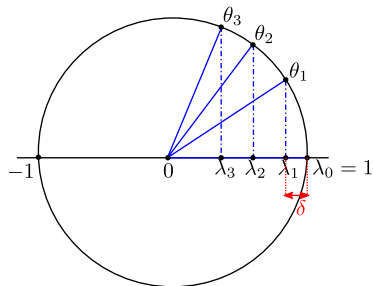
Quantum analogue $W(P)$

- Acts on two registers $|x\rangle|y\rangle$
- Walk in two steps:
 - ▶ reflection of $|y\rangle$ through $|p_x\rangle = \sum_{y'} \sqrt{p_{y'x}} |y'\rangle$
 - ▶ Swap the $|x\rangle$ and $|y\rangle$ registers



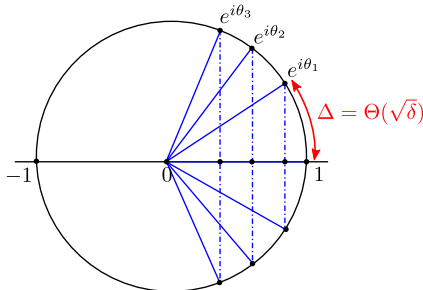
Random walk

- $P = (p_{xy})$
- 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

- $W(P) = \text{SWAP} \cdot \text{ref}_X$
- E-v: $e^{\pm i\theta_k}$
- **Stationary** state ($\theta_0 = 0$):
 $|\pi\rangle = \sum_x \sqrt{\pi_x} |x\rangle |p_x\rangle$
- phase gap: $\Delta = |\theta_1| = \Theta(\sqrt{\delta})$



Back to Grover's algorithm

Grover's algorithm

Prepare $|\pi\rangle$ (S)

Repeat $O(\frac{1}{\sqrt{\epsilon}}) \times$

- Reflection / marked ref_M (C)
- Reflection / uniform ref_π (S)

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

Use quantum walk?

Replace ref_π by

$\rightarrow T \times$ quantum walk???

$$(T = O(\frac{1}{\Delta}) = O(\frac{1}{\sqrt{\delta}}))$$

Tentative quantum walk

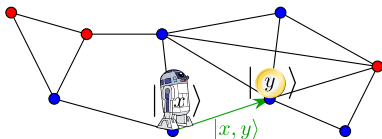
Prepare $|\pi\rangle$ (S)

Repeat $O(\frac{1}{\sqrt{\epsilon}}) \times$

- Reflection / marked ref_M (C)
- Repeat $O(\frac{1}{\sqrt{\delta}}) \times$
 - ▶ Quantum walk (U)

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

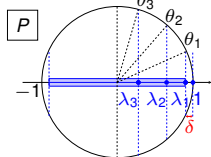
What if S is high???



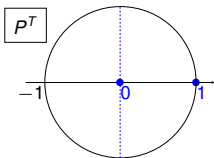
Use repetitions of quantum walk instead of reflection?

Random walk P : eigenvalues

- $\lambda_k = \cos \theta_k \quad |\lambda_k| \leq 1 - \delta$



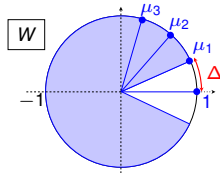
- $(\lambda_k)^T \approx 0 \quad T = O(\frac{1}{\delta})$



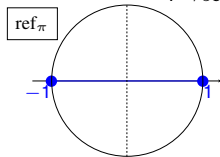
$P^T \approx$ walk on complete graph

Quantum walk W : eigenvalues

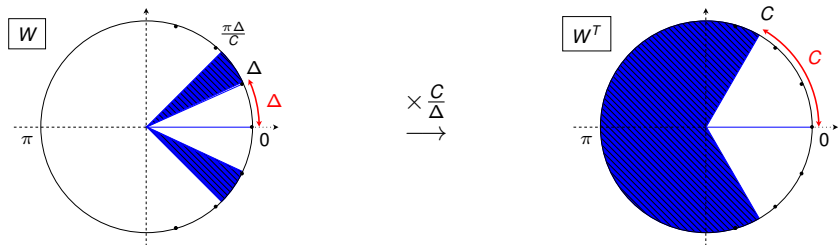
- $\mu_k = e^{2i\theta_k} \quad |1 - \mu_k| \leq \Delta$



- Problem: $(\mu_k)^T \xrightarrow{T \rightarrow \infty} -1$



W^T does not simulate ref_π !



- If W has eigenvalues $e^{i\theta_k}$, W^T has eigenvalues $e^{iT\theta_k}$
- Suppose that $\exists C \leq \pi$ such that $\forall k \neq 0$:

$$\Delta \leq \theta_k \leq \frac{\pi\Delta}{C} \quad \text{or} \quad -\frac{\pi\Delta}{C} \leq \theta_k \leq -\Delta$$

- If $T = \frac{C}{\Delta}$, then W^T has eigenvalue gap $\Delta' = C$

Idea

Replace ref_π by W^T in Grover's algorithm, with $T = O(\frac{1}{\sqrt{\delta}})$

Works under some **assumptions**:

- W^T must have constant gap $\Omega(C)$
- ⇒ OK for Johnson graphs (Element Distinctness)
- Unique solution

Properties:

- **Finds** a marked element
- Cost

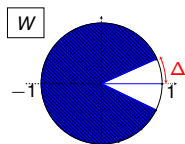
$$S + \frac{1}{\sqrt{\varepsilon}} \left(\frac{1}{\sqrt{\delta}} U + C \right)$$

What about other graphs?

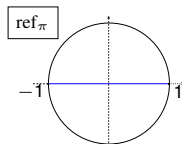
Indirect simulation of the reflection

Idea

Using W , simulate ref_π to use Grover's algorithm



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



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

We need a procedure to discriminate between eigenstates

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

We use quantum phase estimation!

[Kitaev'95, Cleve *et al*'98]

Discriminating between phases 0 and $\geq \Delta$ has a cost $O(1/\Delta) = O(1/\sqrt{\delta})$
 \implies We obtain

- Search algorithm via quantum walk
- from **any ergodic Markov chain**
- Total cost:

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

- **finds** marked elements
- No assumption on the number of marked elements

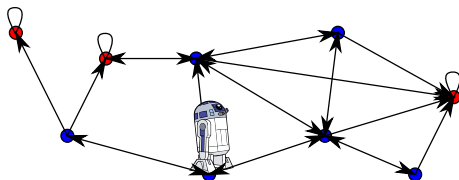
Recall: Random walk II

- Pick random $x \in X$ (S)
- Repeat HT \times
 - ▶ Check whether $x \in M$ (C)
 - ▶ Random move (U)

Cost: S + HT(U + C)

When $x \in M \rightarrow$ **STOP**
 \Rightarrow Equivalent to **random walk P'**

$$P'_{xy} = \begin{cases} P_{xy} & \text{if } x \notin M, \\ \delta_{xy} & \text{if } x \in M. \end{cases}$$



IDEA: Let us build the quantum walk from P' instead of P !

Absorbing walk P'

- \sqrt{HT} iterations of $W(P')$ make $|\pi\rangle$ deviate by angle $\Omega(1)$
 - ▶ Can be used to **detect** marked items
 - ▶ Cost

[Szegedy'04]

$$S + \sqrt{HT}(U + C)$$

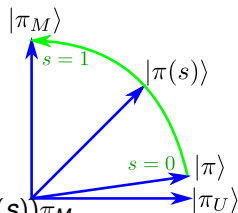
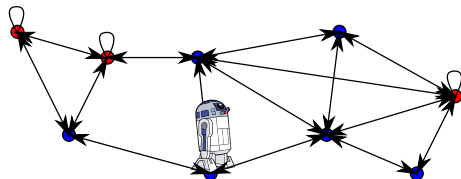
- But: state may remain far from marked elements
 - ▶ Does not **find** marked elements
 - ▶ Can be fixed for state-transitive P
 - ▶ Difficult analysis, less intuition [Tulsi'08][Magniez,Nayak,Richter,Santha'09]

Other approach: interpolation between P and P'

- Finds marked elements for any reversible P
- Good intuition, simpler analysis

Interpolation between P and P'

- $P(s) = (1 - s)P + sP'$
 - ▶ Unmarked vertices: apply P
 - ▶ Marked vertices: apply P with probability $1 - s$, otherwise self-loop



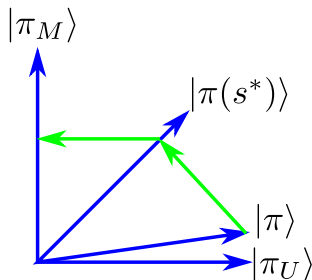
- Stationary distribution $\pi(s) = (\cos^2 \phi(s))\pi_U + (\sin^2 \phi(s))\pi_M$
 - ▶ where $\phi(s) = \arcsin \sqrt{\frac{\epsilon}{1-s(1-\epsilon)}}$
 - ▶ Similarly, $|\pi(s)\rangle = \cos \phi(s)|\pi_U\rangle + \sin \phi(s)|\pi_M\rangle$
 - ▶ Rotates from $|\pi\rangle = \sqrt{1-\epsilon}|\pi_U\rangle + \sqrt{\epsilon}|\pi_M\rangle$ to $|\pi_M\rangle$
- Reminiscent of adiabatic quantum computing
 - ▶ Indeed, we can also design an adiabatic algorithm [Krovi, Ozols, R.'10, PRA]
 - ▶ Note: Interpolation at the **classical** level

General idea

- Using quantum phase estimation [Kitaev'95][Cleve, Ekert, Macchiavello, Mosca'98]
 - ▶ We can measure in the eigenbasis of $W(P(s^*))$
 - ▶ At a cost \sqrt{HT}
- $W(P(s^*))$ has unique 1-eigenvector $|\pi(s^*)\rangle$
 - ▶ Measuring phase 0 projects onto $|\pi(s^*)\rangle$

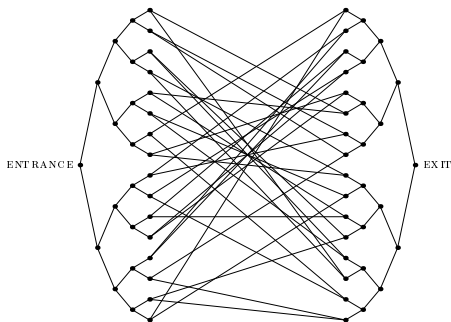
Algorithm

- Prepare $|\pi\rangle$
- Project onto $|\pi(s^*)\rangle = \frac{1}{\sqrt{2}} (|\pi_U\rangle + |\pi_M\rangle)$
 - ▶ succeeds with prob. $\approx 1/2$
- Measure current vertex
 - ▶ marked with prob. $1/2$



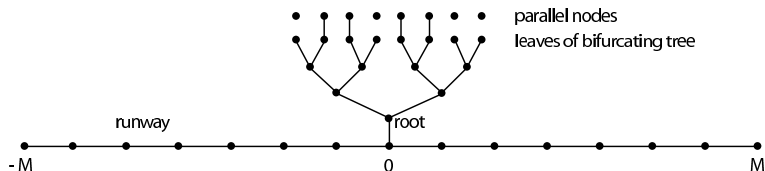
Algorithmic applications

- Grover Search [Grover'95]
 - ▶ Search for a 1 in an n -bit string
 - ▶ G : complete graph
 - ▶ Classical: n Quantum: \sqrt{n}
- Element Distinctness [Ambainis'04]
 - ▶ Search for equal elements in a set of n elements
 - ▶ G : Johnson graph
 - ▶ Classical: n Quantum: $n^{2/3}$
- Triangle Finding [Magniez, Santha, Szegedy'05]
 - ▶ Search for a triangle in a graph with n vertices
 - ▶ G : Johnson graph
 - ▶ Classical: n^2 Quantum: $n^{1.3}$
- Others
 - ▶ Matrix Multiplication Testing [Buhrman, Špalek'06]
 - ▶ Commutativity testing [Magniez, Nayak'05]



- **Goal:** Given the position of “ENTRANCE”, find “EXIT”
 - Classical random walk takes **exponential** time (gets lost in the middle)
 - Quantum walk only takes linear time
- ⇒ **Exponential speedup!**
- Intuition: In “column-space”, quantum walk reduces to walk on the line with defect in the middle.

- **Goal:** Evaluate a Boolean formula $f(x)$

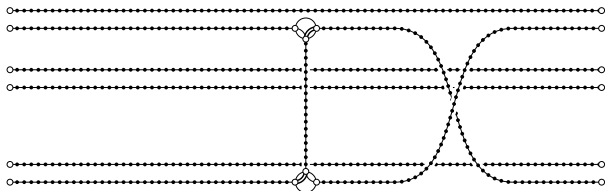


Scattering-based algorithm

- **Idea:** Quantum walk on a tree representing the formula
- Input bits (x_0, x_1, \dots, x_n) correspond to the leaves
- Send incoming wave-packet from the left
- Wave-packet transmitted $\Rightarrow f(x) = 0$
- Wave-packet reflected $\Rightarrow f(x) = 1$

Quantum walks are universal [Childs'09],[Childs,Gossett,Webb'12]

- Any quantum circuit can be implemented as a quantum walk
- Replace any gate in the circuit by a gadget, for example:



- Send incoming wave-packets from the left
- Output of the algorithm given by locations of outgoing wave-packets

Conclusion

Search via quantum walk

- Random walks find an element in time
 - ▶ $S + \frac{1}{\epsilon}(\frac{1}{\delta}U + C)$
 - ▶ $S + HT(U + C)$, where $HT \leq \frac{1}{\epsilon\delta}$
- Quantum walks find an element in time
 - ▶ $S + \frac{1}{\sqrt{\epsilon}}(\frac{1}{\sqrt{\delta}}U + C)$
 - ▶ $S + \sqrt{HT}(U + C)$, where $\sqrt{HT} \leq \frac{1}{\sqrt{\epsilon\delta}}$

Other quantum walk algorithms

- “Glued trees”: exponential speed-up
- Scattering based algorithms:
 - ▶ formula evaluation
 - ▶ universal quantum computation
- And more:
 - ▶ Quantum simulated annealing, Quantum Metropolis algorithm, etc.

Thank you!