Quantum algorithms based on quantum walks

Jérémie Roland

Université Libre de Bruxelles

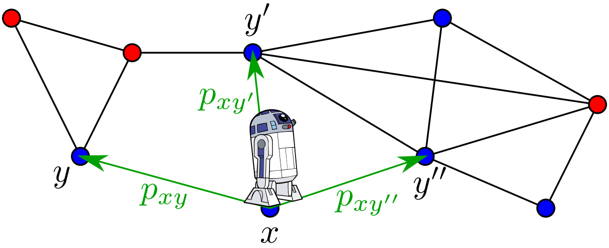
Quantum Information & Communication

Outline

- 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]
  - Nested quantum walks [Childs,Jeffery,Kothari,Magniez'13]
  - 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

Preliminaries

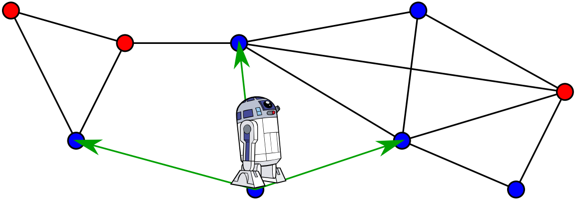
Random walk on a graph



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

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

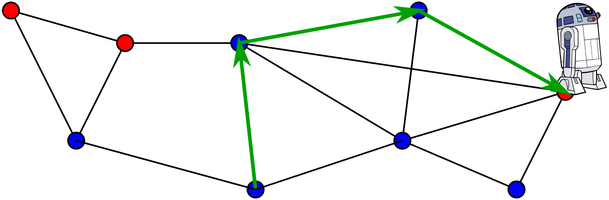
Mixing time



Definition: Mixing time

- Mixing time $\mathrm{MT}(P)$: Number of steps necessary to approach π
- $\mathrm{MT}(P) \leq \frac{1}{\delta}$, 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 $\mathrm{HT}(P, M)$: Expected number of steps to reach some $m \in M$
- $\mathrm{HT}(P, M) \leq \frac{1}{\varepsilon\delta}$, where $\varepsilon = \mathrm{Prob}_{x\sim\pi}[x \in M]$ ($\varepsilon = \frac{|M|}{|X|}$ if π uniform)

Abstract search problem (classical)

The problem

Input:

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

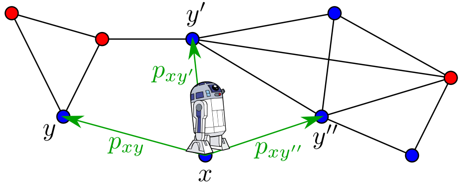
$(\varepsilon = \mathrm{Prob}_{x \sim \pi}[x \in M])$

Output:

- a marked element $x \in M$

Available procedures

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



Three (classical) search algorithms

Naive algorithm

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

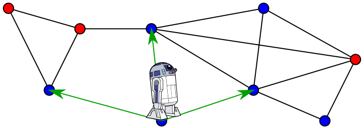
- Pick random $x \in X$ (S)
- Check whether $x \in M$ (C)

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

Idea: Use random walk!

$\mathrm{MT}\times$ random walk $\approx$ pick random x

(mixing time $MT \leq \frac{1}{\delta}$))



Random walk I

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

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

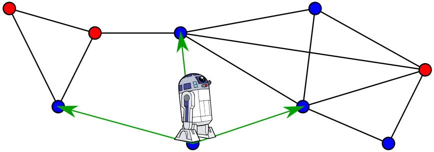
Random walk II

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

Cost: $\mathsf{S}+\mathrm{HT}(\mathsf{U}+\mathsf{C})$

Observations

- Naive algorithm
  - Samples a new independent vertex at each step
  - $\Rightarrow$ 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, $\lambda_i^T \to 0$ whenever $|\lambda_i| < 1$



Search via quantum walk

Abstract search problem (quantum)

Two related problems

Input:

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

Output:

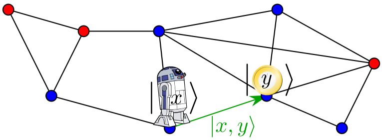
1. Find a marked element $x \in M$
2. Detect whether there is a marked element ($M = \emptyset$?)

Available procedures

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

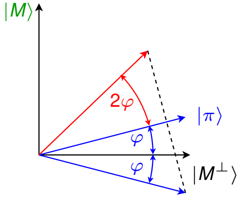
$$\mathrm{ref}_M : |x\rangle \mapsto \begin{cases} |x\rangle & \text{if } x \in M \\ -|x\rangle & \text{otherwise} \end{cases}$$

- Update (cost U):
  apply quantum walk W



Grover's algorithm [Grover'95]

- 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 $M^\perp$: $\mathrm{ref}_{M^\perp} = -\mathrm{ref}_M$ (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 $T\times$
  - apply $\mathrm{ref}_{M^\perp}$ (C)
  - apply ref_π (S)

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

Grover's algorithm: Observations

- 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!

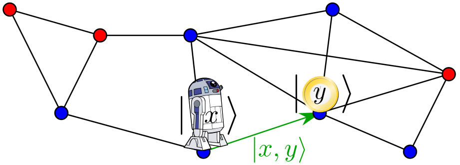
From random to quantum walks [Szegedy'04]

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:
  - Reflect $|y\rangle$ through $|p_x\rangle = \sum_{y'} \sqrt{p_{y'x}}|y'\rangle$
  - Swap the $|x\rangle$ and $|y\rangle$ registers



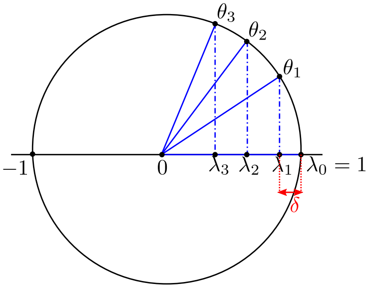
Spectral correspondance [Szegedy'04]

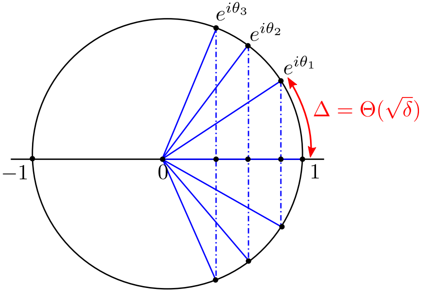
Random walk

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

- $W(P) = \text{SWAP} \cdot \mathrm{ref}_{\mathcal{X}}$
- 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

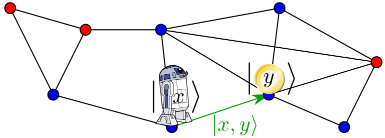
Prepare $|\pi\rangle$ (S)

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

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

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

What if S is high???



Use quantum walk?

Replace ref_π by

$\longrightarrow T\times$ quantum walk???

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

Tentative quantum walk

Prepare $|\pi\rangle$ (S)

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

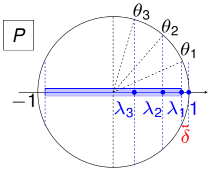
- Reflection / marked ref_M (C)
- Repeat $O(\frac{1}{\sqrt{\delta}})\times$
  - Quantum walk (U)

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

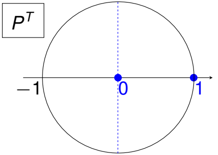
Use repetitions of quantum walk instead of reflection?

Random walk P : eigenvalues

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



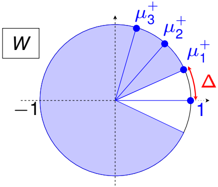
- $(\lambda_k)^T \approx 0 \qquad T = O(\frac{1}{\delta})$



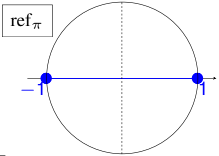
$P^T \approx$ walk on complete graph

Quantum walk W: eigenvalues

- $\mu_k^\pm = e^{\pm i\theta_k} \qquad |1 - \mu_k^\pm| \leq \Delta$

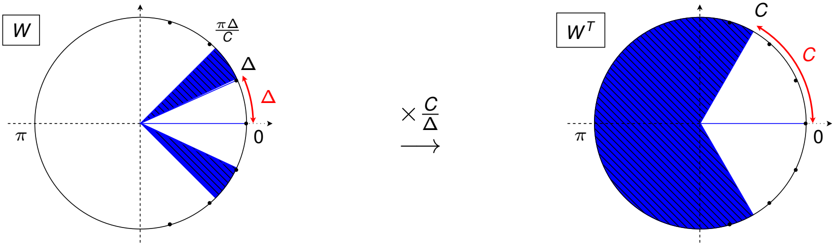


- Problem: $(\mu_k^\pm)^T \cancel{\xrightarrow[T\to\infty]{}} -1$



W^T does not simulate ref_π!

Ambainis' quantum walk [Ambainis'04]



- 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 $\Delta' = C$

Quantum walk for Element Distinctness [Ambainis'04]

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 $\Omega(C)$

$\Rightarrow$ OK for Johnson graphs (Element Distinctness)

- Unique solution

Properties:

- Finds a marked element
- Cost

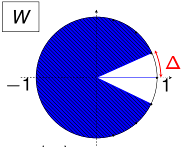
$$\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathsf{U} + \mathsf{C}\right)$$

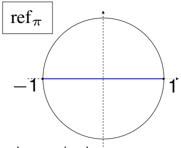
What about other graphs?

Indirect simulation of the reflection

Idea

Using W, simulate ref_π to use Grover's algorithm





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

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

We need a procedure to discriminate between eigenstates

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

We use quantum phase estimation! [Kitaev'95, Cleve *et al*'98]

Using phase estimation [Magniez,Santha,Roland,Nayak'07]

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

- Search algorithm via quantum walk
- from any ergodic Markov chain
- Total cost:
$$\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathsf{U} + \mathsf{C}\right)$$
- finds marked elements
- No assumption on the number of marked elements

Nested quantum walks [Childs,Jeffery,Kothari,Magniez'13]

- Suppose we have a main quantum walk with cost

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

  - where the checking is based on another quantum walk with cost

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

  - Leading to the total cost

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

- By a clever method, this can be reduced to

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

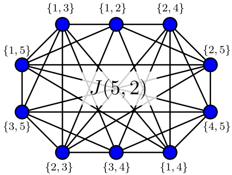
  - Idea: embed the initial state of the inner walk in the data associated to vertices of the outer walk
  - Similar (but more involved) method for nested updates
  - For more details: see Stacey Jeffery's PhD defense on Wednesday

Random walk algorithm type I vs type II

Random walk I

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

Example: Johnson graph $J(n, k)$

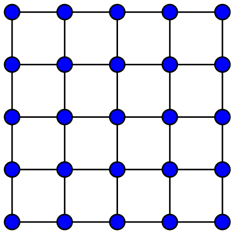


- Definition
  - Vertices: subsets $S \subset [n]$ of size $|S| = k$
  - Edge (S, S') iff $|S \cap S'| = k - 1$
- $\mathrm{HT} = \frac{1}{\varepsilon\delta}$
  - Type I better if C expensive ($\mathsf{C} = \omega(U)$)

Random walk II

Cost: $\mathsf{S} + \mathrm{HT}(\mathsf{U} + \mathsf{C})$

Example: 2D grid (torus)



- $\frac{1}{\varepsilon\delta} = \Theta(n^2)$
- $\mathrm{HT} = \Theta(n \log n)$
  - Type II better if C cheap ($\mathsf{C} = O(\mathsf{U})$)

Szegedy's quantum walk [Szegedy'04]

Recall: Random walk II

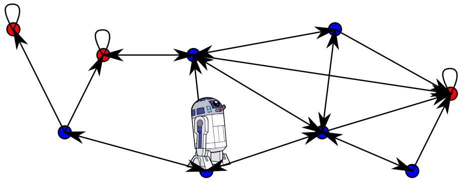
- Pick random $x \in X$ (S)
- Repeat $\mathrm{HT}\times$
  - Check whether $x \in M$ (C)
  - Random move (U)

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

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!

Szegedy's quantum walk [Szegedy'04]

Absorbing walk P'

- $\sqrt{\mathrm{HT}}$ iterations of $W(P')$ make $|\pi\rangle$ deviate by angle $\Omega(1)$
  - Can be used to detect marked items
  - Cost [Szegedy'04]

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

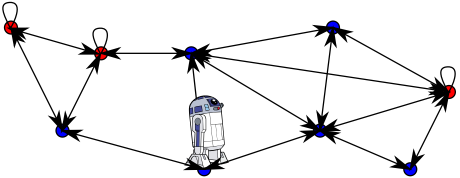
- But: state may remain far from marked elements
  - Does not find marked elements
  - Can be fixed for state-transitive P
  - Difficult analysis, less intuitive [Tulsi'08][Magniez,Nayak,Richter,Santha'09]

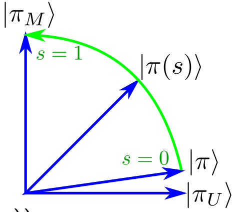
Other approach: interpolation between P and P'

- 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

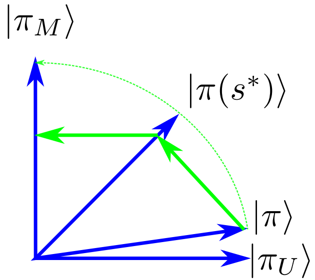
Quantum hitting time algorithm [Krovi,Magniez,Ozols,Roland'10]

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 $\sqrt{\mathrm{HT}^+(s)}(\mathsf{U}+\mathsf{C})$ ($\mathrm{HT}^+(s)$: *interpolated* hitting time)
- $W(P(s))$ 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. $1/2$



Interpolated hitting time $\mathrm{HT}^+(s)$

A few definitions:

- $D(s)$: discriminant of $P(s)$

$$D_{xy}(s) := \sqrt{p_{xy}(s) \cdot p_{yx}(s)}$$

  - $\lambda_k(s)$: eigenvalues of $D(s)$ (same as eigenvalues of $P(s)$)
  - $|v_k(s)\rangle$: eigenvectors of $D(s)$

- $\mathrm{HT}^+(s)$: Interpolated hitting time of $P(s)$

$$\mathrm{HT}^+(s) := \sum_{k=1}^{n-1} \frac{|\langle v_k(s)|\pi_U\rangle|^2}{1-\lambda_k(s)}$$

- $\mathrm{HT}(P, M)$: Usual hitting time of P for set M

$$\mathrm{HT}(P, M) := \sum_{k=|M|}^{n-1} \frac{|\langle v_k(1)|\pi_U\rangle|^2}{1-\lambda_k(1)}$$

  - $\lambda_k(1) = 1$ for $k < |M|$
  - When $|M| > 1$, eigenvalue 1 is $|M|\times$ degenerate ($|M|$ stationary vertices)

Extended hitting time

One more definition:

- $\mathrm{HT}^+(P, M)$: Extended hitting time of P for set M

$$\mathrm{HT}^+(P, M) := \lim_{s \to 1} \mathrm{HT}^+(s)$$

  - When $|M| = 1$, we simply have $\mathrm{HT}^+(P, M) = \mathrm{HT}(P, M)$

- Recall: Quantum walk algorithm II has cost

$$\mathsf{S} + \sqrt{\mathrm{HT}^+(s^*)}(\mathsf{U} + \mathsf{C})$$

  - where s^* is such that $\cos\phi(s^*) = \sin\phi(s^*) = 1/\sqrt{2}$

- We prove that

$$\mathrm{HT}^+(s) = \sin^4\phi(s)\mathrm{HT}^+(P, M)$$

- Hence, quantum walk algorithm II has cost

$$\mathsf{S} + \sqrt{\mathrm{HT}^+(P, M)}(\mathsf{U} + \mathsf{C})$$

Discussion

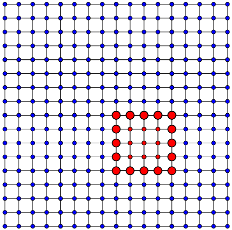
Quantum walk algorithm II has cost:

$$\mathsf{S} + \sqrt{\mathrm{HT}^{+}(P, M)}(\mathsf{U} + \mathsf{C})$$

- When $|M| = 1$ (unique marked vertex)
  - $\mathrm{HT}^{+}(P, M) = \mathrm{HT}(P, M)$
  - Full quadratic speed-up over classical random walk algorithm II
- When $|M| > 1$ (multiple marked vertices)
  - We *may* have $\mathrm{HT}^{+}(P, M) > \mathrm{HT}(P, M)$
  - ... and therefore no full quadratic speed-up

What happens when $|M| > 1$?

- Observation: Output of quantum vs random walk algorithms II
  - Quantum: samples marked vertices according to π_M
  - Classical: returns a marked vertex (but not necessarily fair sampling)
- The quantum algorithm solves a harder problem!
- Example:



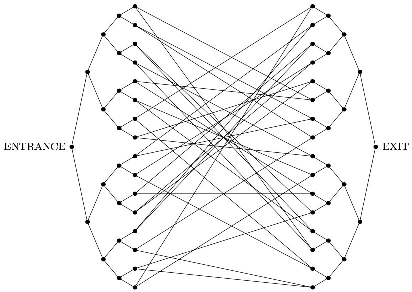
  - The classical random walk algorithm favors vertices on the border
  - This is precisely the kind of situation where $\mathrm{HT}^+(P, M) > \mathrm{HT}(P, M)$
- Note: It is possible to modify the random walk algorithm so that
  - it samples from π_M as well
  - in time $\mathsf{S} + \mathrm{HT}^+(P, M)(\mathsf{U} + \mathsf{C})$

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
  - Search for a triangle in a graph with n vertices
  - G: Johnson graph (with nesting)
  - Classical: n^2
  - Quantum: $n^{1.3}$ [Magniez,Santha,Szegedy'05]
    $n^{1.296}$ [Belovs'12]
    $n^{1.286}$ [Lee,Magniez,Santha'13]
- Others
  - Matrix Multiplication Testing [Buhrman,Špalek'06]
  - Commutativity testing [Magniez,Nayak'05]

Other algorithms based on quantum walks

"Glued trees" [Childs,Cleve,Deotto,Farhi,Gutmann,Spielman'03]

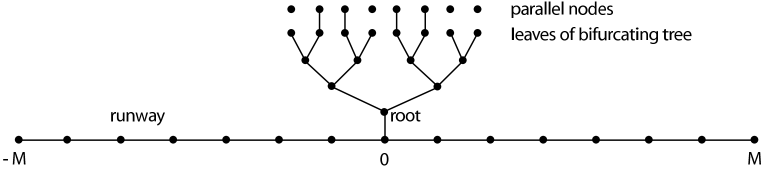


- Goal: Given the position of "ENTRANCE", find "EXIT"
- Classical random walk takes exponential time (gets lost in the middle)
- Quantum walk only takes linear time

$\Rightarrow$ Exponential speedup!

- Intuition: In "column-space", quantum walk reduces to walk on the line with defect in the middle.

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

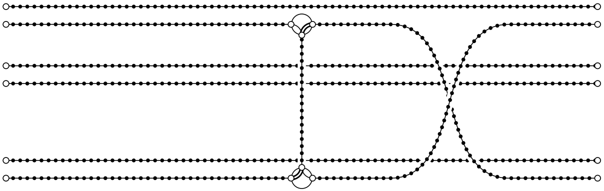


Scattering-based algorithm

- Idea: Quantum walk on a tree representing the formula
- Input bits $(x_0, x_1, \ldots, x_n)$ 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

Conclusion

Search via quantum walk

- Random walks find an element in time
  - $\mathsf{S} + \frac{1}{\epsilon}(\frac{1}{\delta}\mathsf{U} + \mathsf{C})$
  - $\mathsf{S} + \mathrm{HT}(\mathsf{U} + \mathsf{C})$, where $\mathrm{HT} \leq \frac{1}{\varepsilon\delta}$
- Quantum walks find (sample) an element in time
  - $\mathsf{S} + \frac{1}{\sqrt{\epsilon}}(\frac{1}{\sqrt{\delta}}\mathsf{U} + \mathsf{C})$
  - $\mathsf{S} + \sqrt{\mathrm{HT}^+}(\mathsf{U} + \mathsf{C})$, where $\mathrm{HT} \leq \mathrm{HT}^+ \leq \frac{1}{\varepsilon\delta}$ $(\mathrm{HT} = \mathrm{HT}^+$ if $|M| = 1)$

Other quantum walk algorithms

- "Glued trees": exponential speed-up
- Scattering based algorithms:
  - formula evaluation
  - universal quantum computation
- And more:
  - Quantum query algorithms based on span programs and learning graphs
  - Quantum Metropolis algorithm, etc.

Thank you!