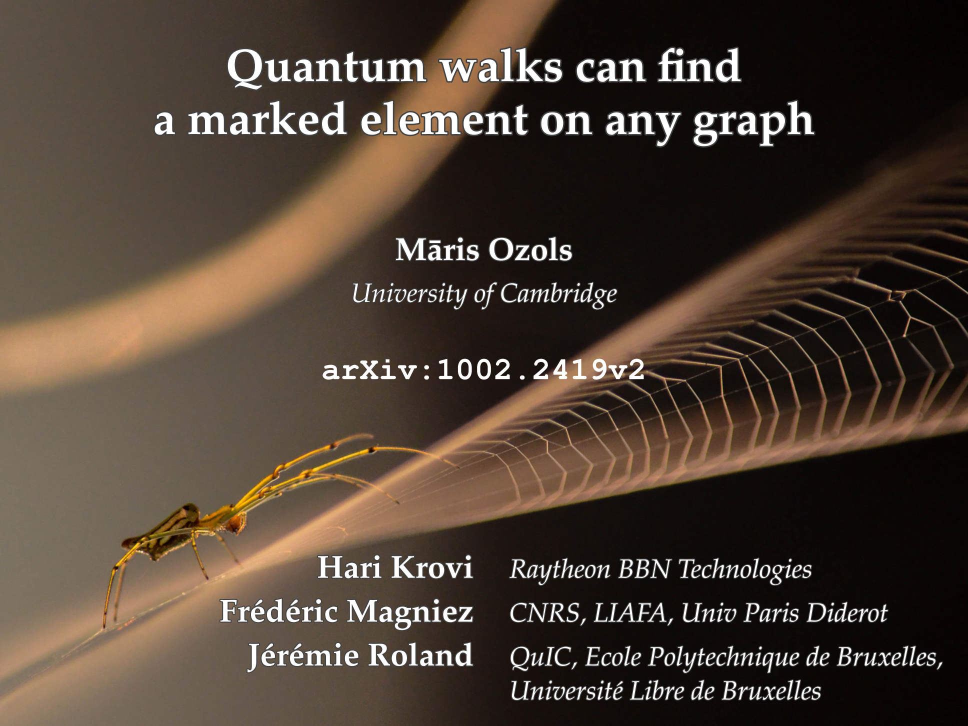


# Quantum walks can find a marked element on any graph


**Māris Ozols**

*University of Cambridge*



arXiv:1002.2419v2



**Hari Krovi** *Raytheon BBN Technologies*

**Frédéric Magniez** *CNRS, LIAFA, Univ Paris Diderot*

**Jérémie Roland** *QuIC, Ecole Polytechnique de Bruxelles, Université Libre de Bruxelles*

# Outline

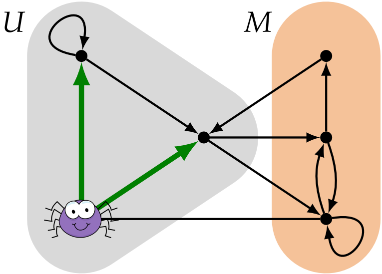
1. Problem and background
2. Main result
3. Classical intuition
4. Quantum algorithm
5. Hitting times

# Problem and background

# Problem: spatial search on a graph

## Setup

- Directed graph on $X = U \cup M$
- Unknown marked vertices $M$
- Edges representing legal moves



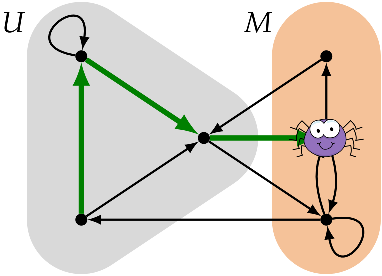
# Problem: spatial search on a graph

## Setup

- Directed graph on $X = U \cup M$
- Unknown marked vertices $M$
- Edges representing legal moves

## Goal

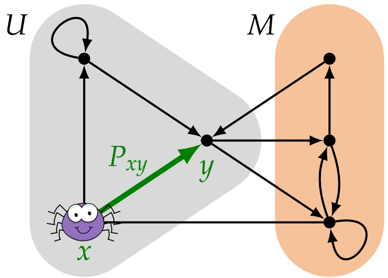
- Find any marked vertex
- Complexity = the number of steps



# Approach: random walk

## Setup

- Stochastic matrix $P = (P_{xy})$
- Restriction: $P_{xy} = 0$ if $(x, y)$ is *not* an edge



# Quantum walks

## Useful early applications

- Element distinctness [Amb04]
- Triangle finding [MSS05]
- Verification of matrix products [BŠ06]
- Testing group commutativity [MN07]

# Quantum walks

## Useful early applications

- Element distinctness [Amb04]
- Triangle finding [MSS05]
- Verification of matrix products [BŠ06]
- Testing group commutativity [MN07]

## Random walk $\to$ quantum walk

- A general "quantization" technique [Sze04a]
- Walk on a complete graph $\to$ Grover's algorithm [Gro96]
- **Goal:** a quadratic quantum speedup for finding a marked vertex compared to any random walk

# Finding with quadratic speedup

## Previous results

- Quadratic speedup for detecting if marked vertices are present [Sze04a]
- Can find, but no quadratic speedup in general [MNRS07]
- Quadratic speedup for state-transitive Markov chains with a unique marked vertex [Tul08, MNRS12]

# Finding with quadratic speedup

## Previous results

- Quadratic speedup for detecting if marked vertices are present [Sze04a]
- Can find, but no quadratic speedup in general [MNRS07]
- Quadratic speedup for state-transitive Markov chains with a unique marked vertex [Tul08, MNRS12]

## Our contribution

- Quadratic speedup for any Markov chain with a unique marked vertex [KOR10, KMOR10]
- Note: Markov chain has to be ergodic and reversible

# Main result

# The main result

**Theorem**

Let $P$ be a reversible, ergodic Markov chain on a set $X$, and $M \subseteq X$ be a set of marked elements. Then a quantum algorithm can find a marked element in $O(\sqrt{\mathrm{HT}^+})$ steps where $\mathrm{HT}^+$ is the "extended" hitting time of $P$

# The main result

**Theorem**

Let $P$ be a reversible, ergodic Markov chain on a set $X$, and $M \subseteq X$ be a set of marked elements. Then a quantum algorithm can find a marked element in $O(\sqrt{\mathrm{HT}^+})$ steps where $\mathrm{HT}^+$ is the "extended" hitting time of $P$

**Note**

- For any $M$, $\mathrm{HT}^+ \geq \mathrm{HT} =$ the hitting time of $P$
- If $|M| = 1$, $\mathrm{HT}^+ = \mathrm{HT}$

# The main ~~result~~ question

### Theorem

Let $P$ be a reversible, ergodic Markov chain on a set $X$, and $M \subseteq X$ be a set of marked elements. Then a quantum algorithm can find a marked element in $O(\sqrt{\mathrm{HT}^+})$ steps where $\mathrm{HT}^+$ is the "extended" hitting time of $P$

### Note

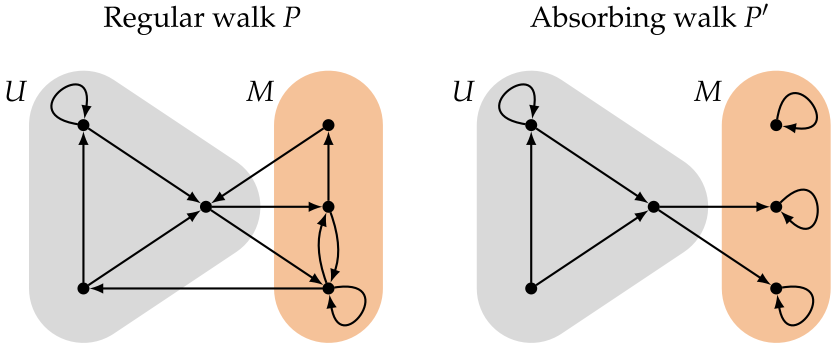
- For any $M$, $\mathrm{HT}^+ \geq \mathrm{HT} =$ the hitting time of $P$
- If $|M| = 1$, $\mathrm{HT}^+ = \mathrm{HT}$

### Question

Can we find in $O(\sqrt{\mathrm{HT}})$ steps for *any* $M$?

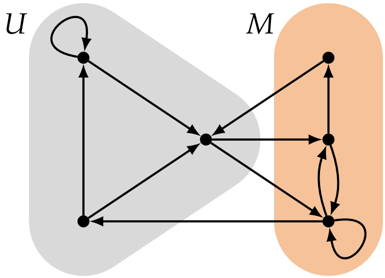
# Classical intuition

# Regular vs absorbing walk



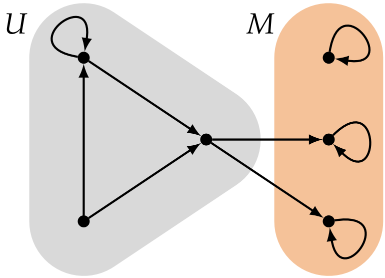
# Regular vs absorbing walk

Regular walk $P$




$$P = \begin{pmatrix} P_{UU} & P_{UM} \\ P_{MU} & P_{MM} \end{pmatrix}$$

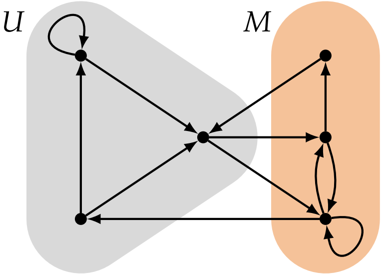
Absorbing walk $P'$




$$P' = \begin{pmatrix} P_{UU} & P_{UM} \\ 0 & I \end{pmatrix}$$

# Regular vs absorbing walk

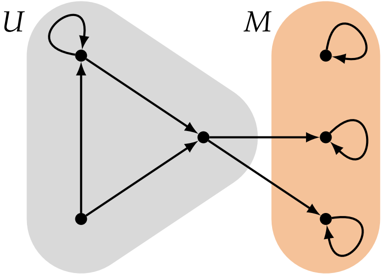
Regular walk $P$




$$P = \begin{pmatrix} P_{UU} & P_{UM} \\ P_{MU} & P_{MM} \end{pmatrix}$$

$$\pi = (\pi_U, \pi_M)$$

Absorbing walk $P'$



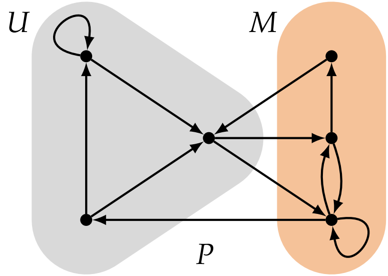

$$P' = \begin{pmatrix} P_{UU} & P_{UM} \\ 0 & I \end{pmatrix}$$

$$\pi' \propto (0, \pi_M)$$

# Classical search via random walk

First…

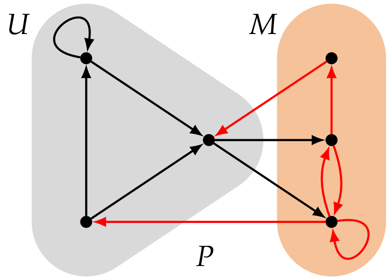
Forget $P$, we will use $P'$ instead!



# Classical search via random walk

First…

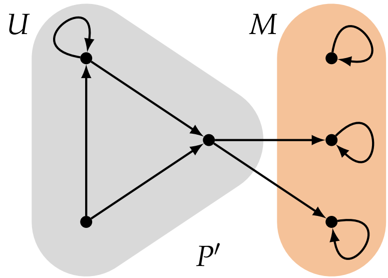
Forget $P$, we will use $P'$ instead!



# Classical search via random walk

First. . .

Forget $P$, we will use $P'$ instead!



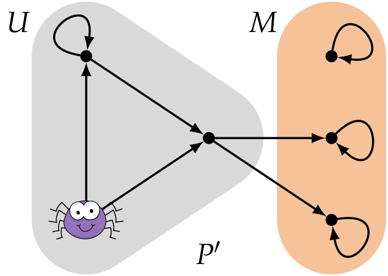
# Classical search via random walk

**First…**

Forget $P$, we will use $P'$ instead!

**Algorithm (BasicWalk)**

1. Pick a random $x \sim \pi$
2. Check if $x \in M$.
   If yes, output $x$ and exit
3. Update $x$ according to
   $P'$ and go back to step 2



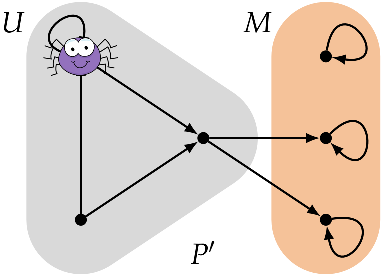
# Classical search via random walk

### First...

Forget $P$, we will use $P'$ instead!

### Algorithm (BasicWalk)

1. Pick a random $x \sim \pi$
2. Check if $x \in M$.
   If yes, output $x$ and exit
3. Update $x$ according to $P'$ and go back to step 2



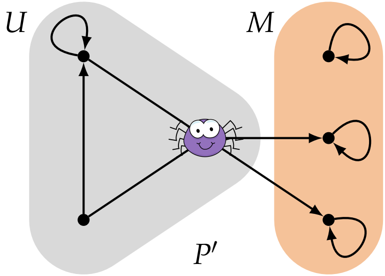
# Classical search via random walk

**First…**

Forget $P$, we will use $P'$ instead!

**Algorithm (BasicWalk)**

1. Pick a random $x \sim \pi$
2. Check if $x \in M$.
   If yes, output $x$ and exit
3. Update $x$ according to
   $P'$ and go back to step 2



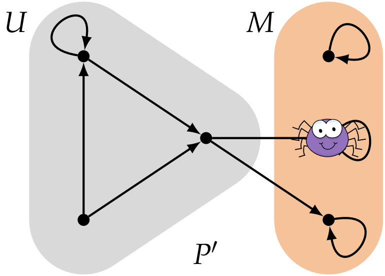
# Classical search via random walk

**First...**

Forget $P$, we will use $P'$ instead!

**Algorithm (BasicWalk)**

1. Pick a random $x \sim \pi$
2. Check if $x \in M$.
   If yes, output $x$ and exit
3. Update $x$ according to $P'$ and go back to step 2

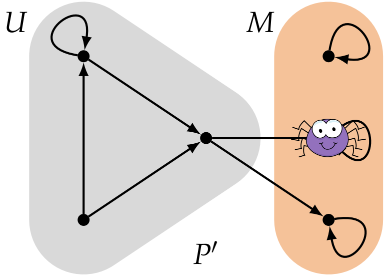


# Classical search via random walk

## First…

Forget $P$, we will use $P'$ instead!

## Algorithm (BasicWalk)




1. Pick a random $x \sim \pi$
2. Check if $x \in M$.
   If yes, output $x$ and exit
3. Update $x$ according to
   $P'$ and go back to step 2

## Hitting time

HT = the expected # of steps of $P'$ to reach any $x \in M$

# “Adiabatic” classical search

## Semi-absorbing walk

- $P(s) := (1-s)P + sP'$ for $s \in [0,1]$
- Stationary distribution: $\pi(s) \propto \big((1-s)\pi_U, \pi_M\big)$

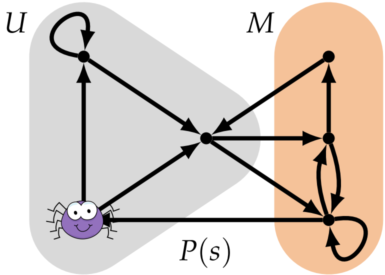
# “Adiabatic” classical search

## Semi-absorbing walk

- $P(s) := (1-s)P + sP'$ for $s \in [0,1]$
- Stationary distribution: $\pi(s) \propto \big((1-s)\pi_U, \pi_M\big)$

## Algorithm

1. Pick a random $x \sim \pi$
2. Apply $P(s)$ several times while changing $s$ from 0 to 1
3. Check if $x$ is marked



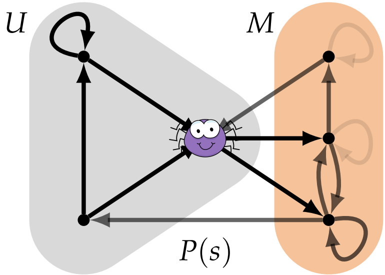
# "Adiabatic" classical search

## Semi-absorbing walk

- $P(s) := (1-s)P + sP'$ for $s \in [0,1]$
- Stationary distribution: $\pi(s) \propto \big((1-s)\pi_U, \pi_M\big)$

## Algorithm

1. Pick a random $x \sim \pi$
2. Apply $P(s)$ several times while changing $s$ from 0 to 1
3. Check if $x$ is marked



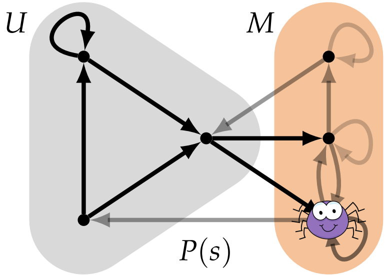
# “Adiabatic” classical search

## Semi-absorbing walk

- $P(s) := (1-s)P + sP'$ for $s \in [0,1]$
- Stationary distribution: $\pi(s) \propto \big((1-s)\pi_U, \pi_M\big)$

## Algorithm

1. Pick a random $x \sim \pi$
2. Apply $P(s)$ several times while changing $s$ from 0 to 1
3. Check if $x$ is marked



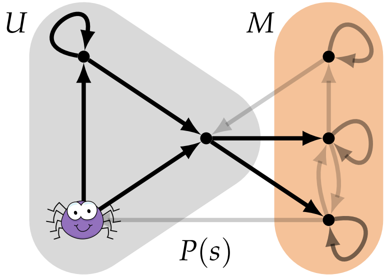
# "Adiabatic" classical search

## Semi-absorbing walk

- $P(s) := (1-s)P + sP'$ for $s \in [0,1]$
- Stationary distribution: $\pi(s) \propto \big((1-s)\pi_U, \pi_M\big)$

## Algorithm

1. Pick a random $x \sim \pi$
2. Apply $P(s)$ several times while changing $s$ from 0 to 1
3. Check if $x$ is marked



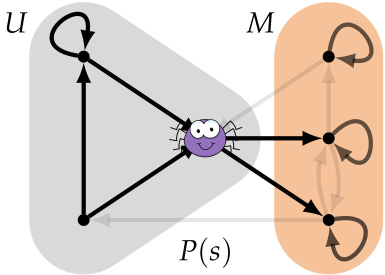
# "Adiabatic" classical search

## Semi-absorbing walk

- $P(s) := (1-s)P + sP'$ for $s \in [0,1]$
- Stationary distribution: $\pi(s) \propto \big((1-s)\pi_U, \pi_M\big)$

## Algorithm

1. Pick a random $x \sim \pi$
2. Apply $P(s)$ several times while changing $s$ from 0 to 1
3. Check if $x$ is marked



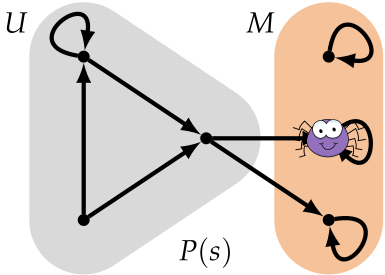
# “Adiabatic” classical search

## Semi-absorbing walk

- $P(s) := (1-s)P + sP'$ for $s \in [0,1]$
- Stationary distribution: $\pi(s) \propto \big((1-s)\pi_U, \pi_M\big)$

## Algorithm

1. Pick a random $x \sim \pi$
2. Apply $P(s)$ several times while changing $s$ from 0 to 1
3. Check if $x$ is marked



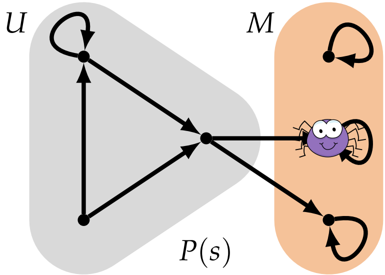
# "Adiabatic" classical search

## Semi-absorbing walk

- $P(s) := (1-s)P + sP'$ for $s \in [0,1]$
- Stationary distribution: $\pi(s) \propto \big((1-s)\pi_U, \pi_M\big)$

## Algorithm

1. Pick a random $x \sim \pi$
2. Apply $P(s)$ several times while changing $s$ from 0 to 1
3. Check if $x$ is marked




## Key observation

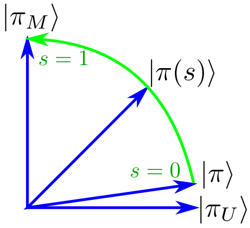
$\pi(s)$ changes continuously from $\pi$ to $\pi'$ as $s$ ranges from 0 to 1

# Quantum algorithm

# Adiabatic version [KOR10]

## Construction

- Use the method of Somma and Ortiz [SO10] to convert $P(s)$ into a Hamiltonian $H(s)$




## Algorithm

1. Prepare $|\pi\rangle$, the quantum state corresponding to $\pi$
2. Evolve by $H(s)$ while interpolating $s$ from 0 to 1

## Theorem

Let $P$ be an ergodic and reversible Markov chain and assume the adiabaticity requirement holds $H(s)$. Then the adiabatic search algorithm finds a marked vertex with probability at least $1 - \varepsilon^2$ in time $T = \frac{\pi}{2\varepsilon}\sqrt{\mathrm{HT}^+}$

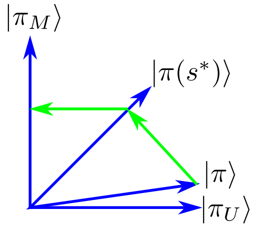
# Circuit version [KMOR10]

## Construction

- Use Szegedy's method [Sze04a] to define a unitary $W(P(s))$
- $W(P(s))$ has a unique 1-eigenvector $|\pi(s)\rangle$
- Use phase estimation to measure in the eigenbasis of $W(P(s))$

## Algorithm

1. Prepare $|\pi\rangle$
2. Project onto $|\pi(s^*)\rangle = \frac{|\pi_U\rangle + |\pi_M\rangle}{\sqrt{2}}$
3. Measure current vertex




## Theorem

If the values of $p_M$ and $\mathrm{HT}^+$ are known, then a quantum algorithm can find a marked vertex in $O\big(\sqrt{\mathrm{HT}^+}\big)$ steps

# Spatial search on $G = (V, E)$

- State space: vertex register $\times$ workspace

$$\underset{V}{\text{vertex register}} \times \underset{V \cup \{\bar{0}\}}{\text{workspace}}$$

# Spatial search on $G = (V, E)$

- State space: vertex register $\times$ workspace
  $$\underset{V}{\text{vertex register}} \times \underset{V \cup \{\bar{0}\}}{\text{workspace}}$$
- Locality-respecting move:
  $$\textsc{Shift}\,|x, y\rangle := \begin{cases} |y, x\rangle & \text{if } (x, y) \in E \\ |x, y\rangle & \text{otherwise} \end{cases}$$

# Spatial search on $G = (V, E)$

- State space: vertex register $\times$ workspace
  $V$ $\qquad$ $V \cup \{\bar{0}\}$
- Locality-respecting move:

$$\textsc{Shift}\,|x, y\rangle := \begin{cases} |y, x\rangle & \text{if } (x, y) \in E \\ |x, y\rangle & \text{otherwise} \end{cases}$$

- Workspace update:

$$V(P)|x\rangle|\bar{0}\rangle := |x\rangle \sum_{y \in X} \sqrt{P_{xy}}|y\rangle$$

# Spatial search on $G = (V, E)$

- State space: vertex register $\times$ workspace

  $V$ $\qquad$ $V \cup \{\bar{0}\}$

- Locality-respecting move:

$$\textsc{Shift}\,|x,y\rangle := \begin{cases} |y,x\rangle & \text{if } (x,y) \in E \\ |x,y\rangle & \text{otherwise} \end{cases}$$

- Workspace update:

$$V(P)|x\rangle|\bar{0}\rangle := |x\rangle \sum_{y \in X} \sqrt{P_{xy}}|y\rangle$$

- Szegedy's walk operator:

$$W(P) := \text{ref}_1 \cdot \text{ref}_2$$

$$\text{ref}_1 := V(P)^{\dagger}\,\textsc{Shift}\,V(P)$$

$$\text{ref}_2 := I \otimes (2|\bar{0}\rangle\langle\bar{0}| - I)$$

# Hitting times

# Operational definition

## Algorithm (BasicWalk)

1. Pick a random $x \sim \pi$
2. Check if $x \in M$. If yes, output $x$ and exit
3. Update $x$ according to $P'$ and go back to step 2

## Hitting time

HT = the expected # of steps of $P'$ to reach any $x \in M$

# Operational definition

## Algorithm (BasicWalk)

1. Pick a random $x \sim \pi$
2. Check if $x \in M$. If yes, output $x$ and exit
3. Update $x$ according to $P'$ and go back to step 2

## Hitting time

HT = the expected # of steps of $P'$ to reach any $x \in M$

$$\mathrm{HT} := \sum_{l=1}^{\infty} l \cdot \Pr[\text{need } \textit{exactly } l \text{ steps}]$$

# Operational definition

## Algorithm (BasicWalk)

1. Pick a random $x \sim \pi$
2. Check if $x \in M$. If yes, output $x$ and exit
3. Update $x$ according to $P'$ and go back to step 2

## Hitting time

HT = the expected # of steps of $P'$ to reach any $x \in M$

$$\begin{aligned} \mathrm{HT} &:= \sum_{l=1}^{\infty} l \cdot \Pr[\text{need } \textit{exactly } l \text{ steps}] \\ &= \sum_{t=0}^{\infty} \Pr[\text{need } \textit{more} \text{ than } t \text{ steps}] \end{aligned}$$

# Operational definition

## Algorithm (BasicWalk)

1. Pick a random $x \sim \pi$
2. Check if $x \in M$. If yes, output $x$ and exit
3. Update $x$ according to $P'$ and go back to step 2

## Hitting time

HT = the expected # of steps of $P'$ to reach any $x \in M$

$$\begin{aligned} \mathrm{HT} &:= \sum_{l=1}^{\infty} l \cdot \Pr[\text{need } \textit{exactly } l \text{ steps}] \\ &= \sum_{t=0}^{\infty} \Pr[\text{need } \textit{more} \text{ than } t \text{ steps}] \\ &= \sum_{t=0}^{\infty} \langle U | D(1)^t | U \rangle \end{aligned}$$

# Ingredients

## Umarked superposition

$$|U\rangle := \frac{1}{\sqrt{p_U}} \sum_{x \in U} \sqrt{\pi_x} |x\rangle$$

# Ingredients

## Umarked superposition

$$|U\rangle := \frac{1}{\sqrt{p_U}} \sum_{x \in U} \sqrt{\pi_x} |x\rangle$$

## Discriminant matrix

- $P(s)$ is not symmetric. Instead, consider

$$D(s) := \sqrt{P(s) \circ P(s)^{\mathsf{T}}}$$

- $P(s)$ and $D(s)$ are similar:

$$D(s) = \operatorname{diag}\big(\sqrt{\pi(s)}\big)\, P(s)\, \operatorname{diag}\big(\sqrt{\pi(s)}\big)^{-1}$$

# Eigenvalues and eigenvectors

- Spectral decomposition:

$$D(s) = \sum_{k=1}^{n} \lambda_k(s) |v_k(s)\rangle \langle v_k(s)|$$

$$0 \leq \lambda_1(s) \leq \lambda_2(s) \leq \cdots \leq \lambda_n(s) = 1$$

# Eigenvalues and eigenvectors

- Spectral decomposition:

$$D(s) = \sum_{k=1}^{n} \lambda_k(s) |v_k(s)\rangle \langle v_k(s)|$$

$$0 \leq \lambda_1(s) \leq \lambda_2(s) \leq \cdots \leq \lambda_n(s) = 1$$

- Multiplicity of eigenvalue 1:

$$\begin{array}{rll} 1 & \text{when} & s \in [0,1) \\ m & \text{when} & s = 1 \end{array}$$

# Eigenvalues and eigenvectors

- Spectral decomposition:

$$D(s) = \sum_{k=1}^{n} \lambda_k(s) |v_k(s)\rangle\langle v_k(s)|$$

$$0 \le \lambda_1(s) \le \lambda_2(s) \le \cdots \le \lambda_n(s) = 1$$

- Multiplicity of eigenvalue 1:

$$\begin{array}{ccc} 1 & \text{when} & s \in [0,1) \\ m & \text{when} & s = 1 \end{array}$$

- Overlap with $|U\rangle$:

$$\langle v_k(1)|U\rangle = 0 \quad \text{when} \quad k > n - m$$

# Operational definition (continued)

## Hitting time

HT = the expected # of steps of $P'$ to reach any $x \in M$

$$
\begin{aligned}
\mathrm{HT} &:= \sum_{l=1}^{\infty} l \cdot \Pr[\text{need } \textit{exactly } l \text{ steps}] \\
&= \sum_{t=0}^{\infty} \Pr[\text{need } \textit{more} \text{ than } t \text{ steps}] \\
&= \sum_{t=0}^{\infty} \langle U | D(1)^t | U \rangle
\end{aligned}
$$

# Operational definition (continued)

## Hitting time

HT = the expected # of steps of $P'$ to reach any $x \in M$

$$\begin{aligned}
\mathrm{HT} := \sum_{l=1}^{\infty} l \cdot \Pr[\text{need } \textit{exactly } l \text{ steps}] \\
= \sum_{t=0}^{\infty} \Pr[\text{need } \textit{more} \text{ than } t \text{ steps}] \\
= \sum_{t=0}^{\infty} \langle U | D(1)^t | U \rangle \\
= \sum_{t=0}^{\infty} \sum_{k=1}^{n} \lambda_k(1)^t \left| \langle v_k(1) | U \rangle \right|^2
\end{aligned}$$

# Operational definition (continued)

## Hitting time

HT = the expected # of steps of $P'$ to reach any $x \in M$

$$
\begin{aligned}
\mathrm{HT} &:= \sum_{l=1}^{\infty} l \cdot \Pr[\text{need } \textit{exactly } l \text{ steps}] \\
&= \sum_{t=0}^{\infty} \Pr[\text{need } \textit{more} \text{ than } t \text{ steps}] \\
&= \sum_{t=0}^{\infty} \langle U | D(1)^t | U \rangle \\
&= \sum_{t=0}^{\infty} \sum_{k=1}^{n} \lambda_k(1)^t \left| \langle v_k(1) | U \rangle \right|^2 \\
&= \sum_{k=1}^{n-m} \frac{\left| \langle v_k(1) | U \rangle \right|^2}{1 - \lambda_k(1)}
\end{aligned}
$$

# Hitting times

- Hitting time:

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

- Interpolated HT:

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

- Extended HT:

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

# Hitting times

- Hitting time:

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

- Interpolated HT:

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

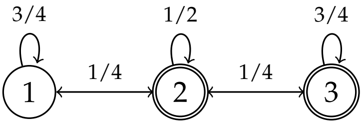
- Extended HT:

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

## Question

$$\text{What is} \quad \lim_{s\to 1} \frac{|\langle v_k(s)|U\rangle|^2}{1-\lambda_k(s)} \quad \text{for} \quad k > n-m?$$

# Example




$$P = \frac{1}{4}\begin{pmatrix} 3 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 3 \end{pmatrix} \qquad M = \{2, 3\}$$

$$\mathrm{HT} = 4 \qquad \mathrm{HT}(s) = \frac{20}{(3-s)^2} \qquad \mathrm{HT}^+ = 5$$

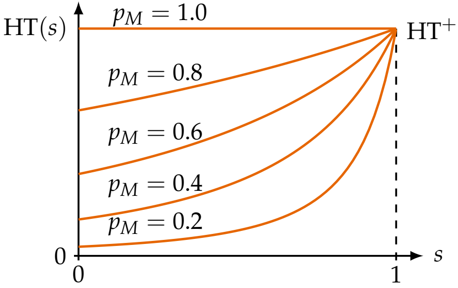
# Main technical result

- Differential equation for HT$(s)$:

$$\frac{d}{ds}\,\mathrm{HT}(s) = \frac{2p_U}{1 - sp_U}\,\mathrm{HT}(s)$$

- Solution:

$$\mathrm{HT}(s) = \left(\frac{1 - p_U}{1 - sp_U}\right)^2 \mathrm{HT}^+$$



# Explicit formulas

$$\mathrm{HT} = \langle \tilde{U} | (I - D_{UU})^{-1} | \tilde{U} \rangle$$
$$\mathrm{HT}^{+} = \langle \tilde{U} | (I - D_{UU} - S)^{-1} | \tilde{U} \rangle$$

where

$$S := D_{UM} \left[ (I - D_{MM})^{-1} - \frac{(I - D_{MM})^{-1} | \tilde{M} \rangle \langle \tilde{M} | (I - D_{MM})^{-1}}{\langle \tilde{M} | (I - D_{MM})^{-1} | \tilde{M} \rangle} \right] D_{MU}$$

$$D := \begin{pmatrix} D_{UU} & D_{UM} \\ D_{MU} & D_{MM} \end{pmatrix} \qquad |\tilde{U}\rangle := \sqrt{\frac{\pi_U}{p_U}} \qquad |\tilde{M}\rangle := \sqrt{\frac{\pi_M}{p_M}}$$

# Explicit formulas

$$\mathrm{HT} = \langle \tilde{U} | (I - D_{UU})^{-1} | \tilde{U} \rangle$$

$$\mathrm{HT}^{+} = \langle \tilde{U} | (I - D_{UU} - S)^{-1} | \tilde{U} \rangle$$

where

$$S := D_{UM} \left[ (I - D_{MM})^{-1} - \frac{(I - D_{MM})^{-1} |\tilde{M}\rangle \langle \tilde{M}| (I - D_{MM})^{-1}}{\langle \tilde{M} | (I - D_{MM})^{-1} | \tilde{M} \rangle} \right] D_{MU}$$

$$D := \begin{pmatrix} D_{UU} & D_{UM} \\ D_{MU} & D_{MM} \end{pmatrix} \qquad |\tilde{U}\rangle := \sqrt{\frac{\pi_U}{p_U}} \qquad |\tilde{M}\rangle := \sqrt{\frac{\pi_M}{p_M}}$$

## Important points

- HT depends only on transitions between *unmarked* states whereas $\mathrm{HT}^{+}$ does not!

# Explicit formulas

$$\mathrm{HT} = \langle \tilde{U} | (I - D_{UU})^{-1} | \tilde{U} \rangle$$
$$\mathrm{HT}^{+} = \langle \tilde{U} | (I - D_{UU} - S)^{-1} | \tilde{U} \rangle$$

where

$$S := D_{UM} \left[ (I - D_{MM})^{-1} - \frac{(I - D_{MM})^{-1} |\tilde{M}\rangle \langle \tilde{M}| (I - D_{MM})^{-1}}{\langle \tilde{M} | (I - D_{MM})^{-1} | \tilde{M} \rangle} \right] D_{MU}$$

$$D := \begin{pmatrix} D_{UU} & D_{UM} \\ D_{MU} & D_{MM} \end{pmatrix} \qquad |\tilde{U}\rangle := \sqrt{\frac{\pi_U}{p_U}} \qquad |\tilde{M}\rangle := \sqrt{\frac{\pi_M}{p_M}}$$

## Important points

- HT depends only on transitions between *unmarked* states whereas $\mathrm{HT}^{+}$ does not!
- We end up sampling a specific distribution over marked states—this might be harder than merely finding one!

# Conclusions

## Results

- Quadratic quantum speed-up of HT for reversible, ergodic Markov chains with 1 marked state
- For multiple marked states, quadratic speedup over $HT^+$

# Conclusions

## Results

- Quadratic quantum speed-up of HT for reversible, ergodic Markov chains with 1 marked state
- For multiple marked states, quadratic speedup over $\mathrm{HT}^+$

## Questions

- What is the meaning of $\mathrm{HT}(s)$ and $\mathrm{HT}^+$?
- Can we get $\sqrt{\mathrm{HT}}$ for multiple marked elements?

# Conclusions

## Results

- Quadratic quantum speed-up of HT for reversible, ergodic Markov chains with 1 marked state
- For multiple marked states, quadratic speedup over $\mathrm{HT}^+$

## Questions

- What is the meaning of $\mathrm{HT}(s)$ and $\mathrm{HT}^+$?
- Can we get $\sqrt{\mathrm{HT}}$ for multiple marked elements?

## Related talks

| | | | |
|---|---|---|---|
| Mon | 16:30 | Ambanis | Quantum search and extended hitting time |
| Tue | 16:30 | Montanaro | Quantum speedup of backtracking |
| Wed | 10:15 | Belovs | Quantum walks and electric networks |

# Conclusions

## Results

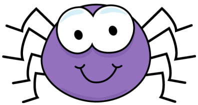
- Quadratic quantum speed-up of HT for reversible, ergodic Markov chains with 1 marked state
- For multiple marked states, quadratic speedup over $\mathrm{HT}^+$

## Questions

- What is the meaning of $\mathrm{HT}(s)$ and $\mathrm{HT}^+$?
- Can we get $\sqrt{\mathrm{HT}}$ for multiple marked elements?

## Related talks

| | | | |
|---|---|---|---|
| Mon | 16:30 | Ambanis | Quantum search and extended hitting time |
| Tue | 16:30 | Montanaro | Quantum speedup of backtracking |
| Wed | 10:15 | Belovs | Quantum walks and electric networks |



Thank you!