Finding a marked node on any graph by continuous-time quantum walk

Shantanav Chakraborty, Leonardo Novo, Jérémie Roland

arXiv:1807.05957

QuIC, Université libre de Bruxelles

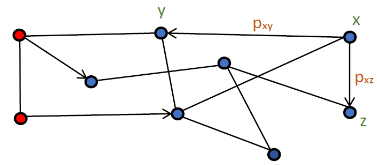
ICoQC Paris

November 26, 2018

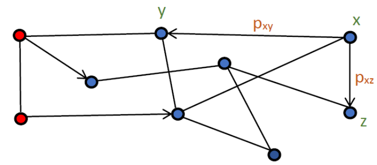
ULB UNIVERSITÉ LIBRE DE BRUXELLES

QuIC Quantum Information & Communication

Classical random walk on a graph

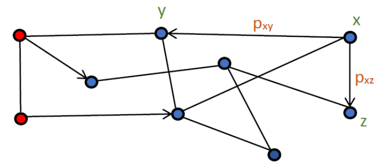


Classical random walk on a graph



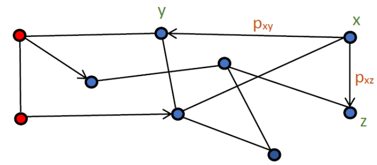
- Classical random walk on a discrete state space X, such that $|X| = n$.
- Described by a $n \times n$ stochastic matrix P such that its $(x, y)^{\mathrm{th}}$ entry is p_{xy}.
- If the row-vector v_0 is the initial state of the walker, after t-steps: $v_t = v_0 P^t$.

Classical random walk on a graph



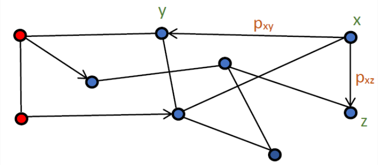
- Classical random walk on a discrete state space X, such that $|X| = n$.
- Described by a $n \times n$ stochastic matrix P such that its $(x, y)^{\mathrm{th}}$ entry is p_{xy}.
- If the row-vector v_0 is the initial state of the walker, after t-steps: $v_t = v_0 P^t$.
- Stationary state: row vector π such that $\pi = \pi P$.

Classical random walk on a graph



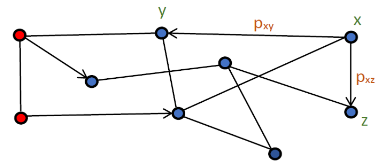
- Classical random walk on a discrete state space X, such that $|X| = n$.
- Described by a $n \times n$ stochastic matrix P such that its $(x, y)^{\mathrm{th}}$ entry is p_{xy}.
- If the row-vector v_0 is the initial state of the walker, after t-steps: $v_t = v_0 P^t$.
- Stationary state: row vector π such that $\pi = \pi P$.
- Assumptions: **P is ergodic**

Classical random walk on a graph



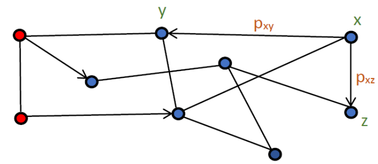
- Classical random walk on a discrete state space X, such that $|X| = n$.
- Described by a $n \times n$ stochastic matrix P such that its $(x, y)^{\mathrm{th}}$ entry is p_{xy}.
- If the row-vector v_0 is the initial state of the walker, after t-steps: $v_t = v_0 P^t$.
- Stationary state: row vector π such that $\pi = \pi P$.
- Assumptions: **P is ergodic**
  - Eigenvalues of P lie between -1 and 1.
  - π is unique.

Classical Hitting time



Set of marked nodes:
$M \subseteq X$.

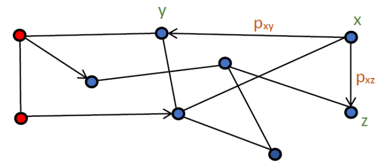
Classical Hitting time



Set of marked nodes: $M \subseteq X$.

Hitting time: Starting from some random node $x \sim \pi$, the expected number of steps to reach some node $\in M$.

Classical Hitting time



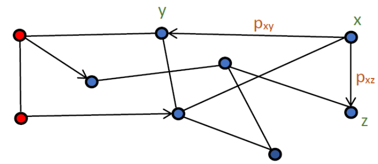
Set of marked nodes: $M \subseteq X$.

Hitting time: Starting from some random node $x \sim \pi$, the expected number of steps to reach some node $\in M$.

Spatial search (classical)

1. Sample $x \in X$ from π.
2. Check if $x \in M$.
3. If $x \in M$, output x
4. Otherwise update x according to P and go to step 2.

Classical Hitting time



Set of marked nodes: $M \subseteq X$.

Hitting time: Starting from some random node $x \sim \pi$, the expected number of steps to reach some node $\in M$.

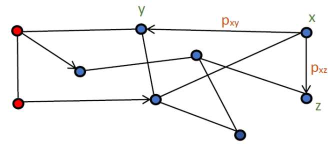
Spatial search (classical)

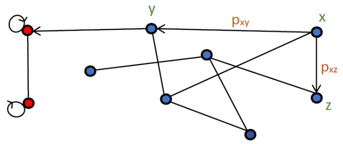
1. Sample $x \in X$ from π.
2. Check if $x \in M$.
3. If $x \in M$, output x
4. Otherwise update x according to P and go to step 2.

Hitting time of P with respect to M is the expected number of times step 4 is executed.

Classical Hitting time

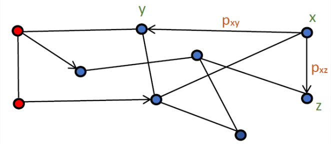
Spatial search stops when $x \in M \implies$ Walk on an absorbing Markov chain P'

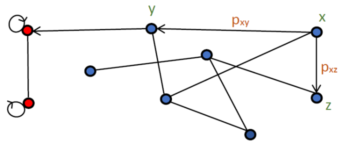




Classical Hitting time

Spatial search stops when $x \in M \implies$ Walk on an absorbing Markov chain P'

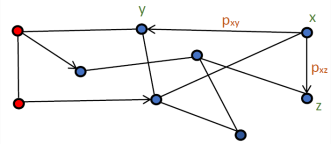


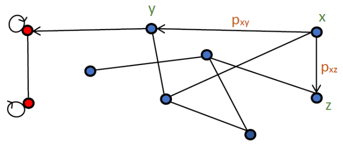


$HT(P, M) =$ Expected number of steps of P' to reach some $x \in M$.

Classical Hitting time

Spatial search stops when $x \in M \implies$ Walk on an absorbing Markov chain P'



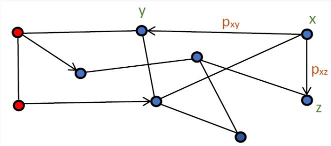


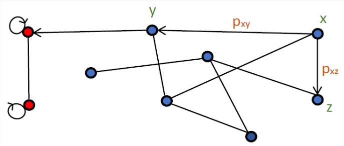
$HT(P, M) =$ Expected number of steps of P' to reach some $x \in M$.

Complexity of spatial search by **quantum walk**?

Classical Hitting time

Spatial search stops when $x \in M \implies$ Walk on an absorbing Markov chain P'





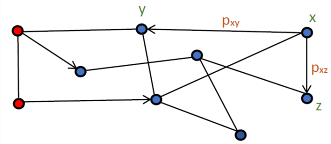
$HT(P, M) =$ Expected number of steps of P' to reach some $x \in M$.

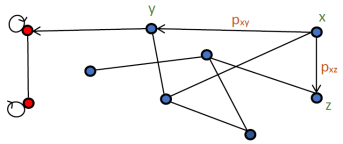
Complexity of spatial search by **quantum walk**?

Discrete time quantum walk (DTQW)?

Classical Hitting time

Spatial search stops when $x \in M \implies$ Walk on an absorbing Markov chain P'



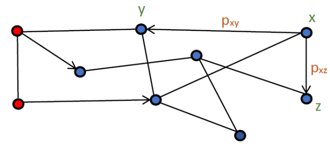


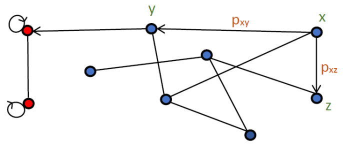
$HT(P, M) =$ Expected number of steps of P' to reach some $x \in M$.

> Complexity of spatial search by **quantum walk**?
>
> Discrete time quantum walk (DTQW)?
>
> Continuous-time quantum walk (CTQW)?

Framework

Let $U = X \backslash M$.



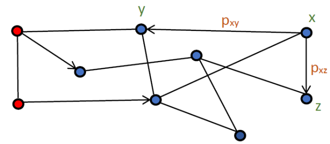


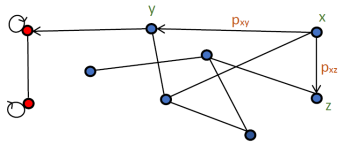
$\pi = (\pi_U \;\; \pi_M)$

$\pi' \propto (0 \;\; \pi_M)$ (P' is not ergodic)

Framework

Let $U = X \backslash M$.





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

$\pi' \propto (0 \;\; \pi_M)$ (P' is not ergodic)

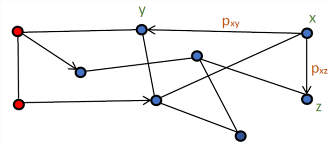
Additional assumption: P is reversible $\implies \pi_x p_{xy} = \pi_y p_{yx}, \;\; \forall (x, y) \in X$.

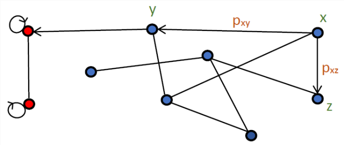
Interpolating Markov Chains

$P(s) = (1 - s)P + sP', \;\; s \in [0, 1]$

Framework

Let $U = X \backslash M$.





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

$\pi' \propto (0 \quad \pi_M)$ (P' is not ergodic)

Additional assumption: P is reversible $\implies \pi_x p_{xy} = \pi_y p_{yx}, \ \forall (x, y) \in X$.

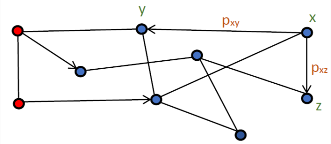
Interpolating Markov Chains

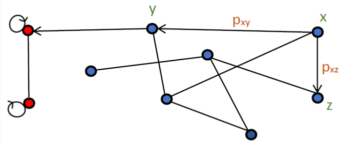
$P(s) = (1 - s)P + sP', \ s \in [0, 1]$

For $0 \leq s < 1$, stationary state: $\pi(s) \propto ((1 - s)\pi_U \quad \pi_M)$.

Framework

Let $U = X \backslash M$.





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

$\pi' \propto (0 \quad \pi_M)$ (P' is not ergodic)

Additional assumption: P is reversible $\implies \pi_x p_{xy} = \pi_y p_{yx}, \ \forall (x, y) \in X$.

Interpolating Markov Chains

$P(s) = (1 - s)P + sP', \ s \in [0, 1]$

For $0 \leq s < 1$, stationary state: $\pi(s) \propto ((1 - s)\pi_U \quad \pi_M)$.

If P is reversible $\implies$ $P(s)$ is reversible for $0 \leq s < 1$.

Interpolating Markov Chains

$P(s) = (1-s)P + sP',\ s \in [0,1]$

For $0 \leq s < 1$, stationary state: $\pi(s) \propto ((1-s)\pi_U \quad \pi_M)$.

If P is reversible $\implies$ $P(s)$ is reversible for $0 \leq s < 1$.

Interpolating Markov Chains

$P(s) = (1-s)P + sP', \; s \in [0,1]$

For $0 \leq s < 1$, stationary state: $\pi(s) \propto ((1-s)\pi_U \quad \pi_M)$.

If P is reversible $\implies$ $P(s)$ is reversible for $0 \leq s < 1$.

We will work with

- Lazy walk: $(I + P(s))/2$
  - Eigenvalues: $0 < \lambda_1(s) \leq \lambda_2(s) \leq \cdots \lambda_n(s) = 1$.

Interpolating Markov Chains

$P(s) = (1-s)P + sP',\ \ s \in [0,1]$

For $0 \leq s < 1$, stationary state: $\pi(s) \propto ((1-s)\pi_U \ \ \pi_M)$.

If P is reversible $\implies$ $P(s)$ is reversible for $0 \leq s < 1$.

We will work with

- Lazy walk: $(I + P(s))/2$
  - Eigenvalues: $0 < \lambda_1(s) \leq \lambda_2(s) \leq \cdots \lambda_n(s) = 1$.
- Discriminant matrix of $P(s)$: $D(P(s)) = \sqrt{P(s) \circ P(s)^T}$.
  - $D_{xy}(s) = \sqrt{p_{xy}(s) p_{yx}(s)}$.

Interpolating Markov Chains

$P(s) = (1-s)P + sP', \;\; s \in [0,1]$

For $0 \leq s < 1$, stationary state: $\pi(s) \propto ((1-s)\pi_U \quad \pi_M)$.

If P is reversible $\implies$ $P(s)$ is reversible for $0 \leq s < 1$.

We will work with

- Lazy walk: $(I + P(s))/2$
  - Eigenvalues: $0 < \lambda_1(s) \leq \lambda_2(s) \leq \cdots \lambda_n(s) = 1$.
- Discriminant matrix of $P(s)$: $D(P(s)) = \sqrt{P(s) \circ P(s)^T}$.
  - $D_{xy}(s) = \sqrt{p_{xy}(s) p_{yx}(s)}$.
  - Spectral decomposition: $D(P(s)) = \sum_{i=1}^{n} \lambda_i(s) \left| v_i(s) \right\rangle \left\langle v_i(s) \right|$.

Interpolating Markov Chains

$P(s) = (1-s)P + sP', \;\; s \in [0,1]$

For $0 \leq s < 1$, stationary state: $\pi(s) \propto ((1-s)\pi_U \;\; \pi_M)$.

If P is reversible $\implies$ $P(s)$ is reversible for $0 \leq s < 1$.

We will work with

- Lazy walk: $(I + P(s))/2$
  - Eigenvalues: $0 < \lambda_1(s) \leq \lambda_2(s) \leq \cdots \lambda_n(s) = 1$.
- Discriminant matrix of $P(s)$: $D(P(s)) = \sqrt{P(s) \circ P(s)^T}$.
  - $D_{xy}(s) = \sqrt{p_{xy}(s) p_{yx}(s)}$.
  - Spectral decomposition: $D(P(s)) = \sum_{i=1}^{n} \lambda_i(s) \left| v_i(s) \right\rangle \left\langle v_i(s) \right|$.
  - Same eigenvalues as $P(s)$.

Interpolating Markov Chains

$P(s) = (1-s)P + sP', \ s \in [0,1]$

For $0 \leq s < 1$, stationary state: $\pi(s) \propto ((1-s)\pi_U \ \ \pi_M)$.

If P is reversible $\implies$ $P(s)$ is reversible for $0 \leq s < 1$.

We will work with

- Lazy walk: $(I + P(s))/2$
  - Eigenvalues: $0 < \lambda_1(s) \leq \lambda_2(s) \leq \cdots \lambda_n(s) = 1$.
- Discriminant matrix of $P(s)$: $D(P(s)) = \sqrt{P(s) \circ P(s)^T}$.
  - $D_{xy}(s) = \sqrt{p_{xy}(s) p_{yx}(s)}$.
  - Spectral decomposition: $D(P(s)) = \sum_{i=1}^{n} \lambda_i(s) \left| v_i(s) \right\rangle \left\langle v_i(s) \right|$.
  - Same eigenvalues as $P(s)$.
  - $\left| v_n(s) \right\rangle = \sqrt{\pi(s)^T}$.

Interpolating Markov Chains

$P(s) = (1-s)P + sP', \;\; s \in [0,1]$

For $0 \leq s < 1$, stationary state: $\pi(s) \propto ((1-s)\pi_U \quad \pi_M)$.

If P is reversible $\implies$ $P(s)$ is reversible for $0 \leq s < 1$.

We will work with

- Lazy walk: $(I + P(s))/2$
  - Eigenvalues: $0 < \lambda_1(s) \leq \lambda_2(s) \leq \cdots \lambda_n(s) = 1$.
- Discriminant matrix of $P(s)$: $D(P(s)) = \sqrt{P(s) \circ P(s)^T}$.
  - $D_{xy}(s) = \sqrt{p_{xy}(s) p_{yx}(s)}$.
  - Spectral decomposition: $D(P(s)) = \sum_{i=1}^{n} \lambda_i(s) \left| v_i(s) \right\rangle \left\langle v_i(s) \right|$.
  - Same eigenvalues as $P(s)$.
  - $\left| v_n(s) \right\rangle = \sqrt{\pi(s)^T}$.
  - Spectral gap: $\Delta(s) = 1 - \lambda_{n-1}(s)$.

The known and the unknown

Complexity of spatial search by DTQW

For any *ergodic*, *reversible* Markov chain P with a set of M marked nodes: $\mathcal{O}\left(\sqrt{HT^{+}(P,M)}\right)$.
[Krovi, Magniez, Ozols, and Roland 2014]

$HT^{+}(P,M)$: *Extended hitting time*

The known and the unknown

Complexity of spatial search by DTQW

For any *ergodic*, *reversible* Markov chain P with a set of M marked nodes: $\mathcal{O}\left(\sqrt{HT^+(P,M)}\right)$.
[Krovi, Magniez, Ozols, and Roland 2014]

$HT^+(P,M)$: *Extended hitting time*

- For $|M| = 1$, $HT^+(P,M) = HT(P,M) \implies$ Quadratic speedup for unique marked node.

The known and the unknown

Complexity of spatial search by DTQW

For any *ergodic*, *reversible* Markov chain P with a set of M marked nodes: $\mathcal{O}\left(\sqrt{HT^{+}(P, M)}\right)$.

[Krovi, Magniez, Ozols, and Roland 2014]

$HT^{+}(P, M)$: *Extended hitting time*

- For $|M| = 1$, $HT^{+}(P, M) = HT(P, M) \implies$ Quadratic speedup for unique marked node.
- For $|M| > 1$, $HT^{+}(P, M) \geq HT(P, M)$.

The known and the unknown

Complexity of spatial search by DTQW

For any *ergodic*, *reversible* Markov chain P with a set of M marked nodes: $\mathcal{O}\left(\sqrt{HT^+(P,M)}\right)$.

[Krovi, Magniez, Ozols, and Roland 2014]

$HT^+(P,M)$: *Extended hitting time*

- For $|M| = 1$, $HT^+(P,M) = HT(P,M) \implies$ Quadratic speedup for unique marked node.
- For $|M| > 1$, $HT^+(P,M) \geq HT(P,M)$.

Complexity of spatial search by CTQW

No such general result known.

The algorithm by Childs and Goldstone has been applied to certain specific graphs such as d-dimensional lattices, hypercubes and others [Childs and Goldstone 2004 and several subsequent papers].

Main results

- For any ergodic, reversible Markov chain provide a spatial search algorithm by CTQW that has a running time of $\Theta(\sqrt{HT^+(P,M)})$.

Main results

- For any ergodic, reversible Markov chain provide a spatial search algorithm by CTQW that has a running time of $\Theta(\sqrt{HT^{+}(P,M)})$.
- State general conditions for the optimality of the Childs and Goldstone algorithm on any ergodic, reversible Markov chain.

Main results

- For any ergodic, reversible Markov chain provide a spatial search algorithm by CTQW that has a running time of $\Theta(\sqrt{HT^{+}(P,M)})$.
- State general conditions for the optimality of the Childs and Goldstone algorithm on any ergodic, reversible Markov chain.
- Compare the running time of our algorithm with the Childs and Goldstone algorithm.

Continuous-time quantum walk on a graph

- Evolve a time-independent Hamiltonian, H_G, encoding the connectivity of the underlying graph according to the Schrödinger equation.

Continuous-time quantum walk on a graph

- Evolve a time-independent Hamiltonian, H_G, encoding the connectivity of the underlying graph according to the Schrödinger equation.
- Initial state: $|\psi_0\rangle$

Continuous-time quantum walk on a graph

- Evolve a time-independent Hamiltonian, H_G, encoding the connectivity of the underlying graph according to the Schrödinger equation.
- Initial state: $|\psi_0\rangle$
- Probability of the walker being in state $|\psi_f\rangle$, after time t

$$p(t) = |\langle\psi_f| e^{-iH_G t} |\psi_0\rangle|^2.$$

Continuous-time quantum walk on a graph

- Evolve a time-independent Hamiltonian, H_G, encoding the connectivity of the underlying graph according to the Schrödinger equation.
- Initial state: $|\psi_0\rangle$
- Probability of the walker being in state $|\psi_f\rangle$, after time t

$$p(t) = |\langle\psi_f| e^{-iH_G t} |\psi_0\rangle|^2.$$

Task

Convert a graph (or a Markov chain) to a Hamiltonian.
We will use the formalism of Somma and Ortiz [Somma and Ortiz 2010].
Also used to develop an adiabatic version of quantum spatial search [Krovi, Ozols and Roland 2010].

Search Hamiltonian

Interpolating Markov Chains

$P(s) = (1-s)P + sP', \; s \in [0,1)$

Discriminant matrix: $D_{xy}(s) = \sqrt{p_{xy}(s)p_{yx}(s)}$.

Same eigenvalues as $P(s)$. Spectral gap: $\Delta(s) = 1 - \lambda_{n-1}(s)$.

$|v_n(s)\rangle = \sqrt{\pi(s)^T}$.

$\mathcal{H}$: n-dimensional Hilbert space whose basis states are labelled by the vertices of P.

Consider a unitary $V(s) \in \mathcal{H} \times \mathcal{H}$:

$$V(s)\,|x,0\rangle = \sum_{y \in X} \sqrt{p_{xy}(s)}\,|x,y\rangle\,.$$

Search Hamiltonian

Interpolating Markov Chains

$P(s) = (1-s)P + sP', \; s \in [0,1)$

Discriminant matrix: $D_{xy}(s) = \sqrt{p_{xy}(s)p_{yx}(s)}$.

Same eigenvalues as $P(s)$. Spectral gap: $\Delta(s) = 1 - \lambda_{n-1}(s)$.

$|v_n(s)\rangle = \sqrt{\pi(s)^T}$.

$\mathcal{H}$: n-dimensional Hilbert space whose basis states are labelled by the vertices of P.

Consider a unitary $V(s) \in \mathcal{H} \times \mathcal{H}$:

$$V(s)\,|x,0\rangle = \sum_{y\in X} \sqrt{p_{xy}(s)}\,|x,y\rangle\,.$$

If $S\,|x,y\rangle = |y,x\rangle$, then observe that

$$\langle x,0|V^\dagger(s)SV(s)|y,0\rangle = \sqrt{p_{xy}(s)p_{yx}(s)} = D_{xy}(s).$$

Search Hamiltonian

Interpolating Markov Chains

$P(s) = (1-s)P + sP', \ s \in [0,1)$

Discriminant matrix: $D_{xy}(s) = \sqrt{p_{xy}(s)p_{yx}(s)}$.

Same eigenvalues as $P(s)$. Spectral gap: $\Delta(s) = 1 - \lambda_{n-1}(s)$.

$|v_n(s)\rangle = \sqrt{\pi(s)^T}$.

$\mathcal{H}$: n-dimensional Hilbert space whose basis states are labelled by the vertices of P.

Consider a unitary $V(s) \in \mathcal{H} \times \mathcal{H}$:

$$V(s) |x,0\rangle = \sum_{y \in X} \sqrt{p_{xy}(s)} |x,y\rangle .$$

If $S|x,y\rangle = |y,x\rangle$, then observe that

$$\langle x,0| V^\dagger(s) S V(s) |y,0\rangle = \sqrt{p_{xy}(s)p_{yx}(s)} = D_{xy}(s).$$

If $\Pi_0 = I \otimes |0\rangle\langle 0|$,

$$H(s) = i[V^\dagger(s) S V(s), \Pi_0].$$

Spectrum of $H(s)$

- $|v_n(s), 0\rangle$ is an eigenstate of $H(s)$ with eigenvalue 0.
- Spectral gap of $H(s)$: $\sqrt{1-\lambda_{n-1}^2(s)} = \Theta(\sqrt{\Delta(s)})$.

Spectrum of $H(s)$

- $|v_n(s), 0\rangle$ is an eigenstate of $H(s)$ with eigenvalue 0.
- Spectral gap of $H(s)$: $\sqrt{1-\lambda_{n-1}^2(s)} = \Theta(\sqrt{\Delta(s)})$.

For $s = s^* = 1 - p_M/(1-p_M)$, $|v_n(s^*)\rangle = \frac{|U\rangle + |M\rangle}{\sqrt{2}}$.

$$p_M = \sum_{x \in M} \pi_x$$

$$|U\rangle = \frac{1}{\sqrt{1-p_M}} \sum_{x \notin M} \sqrt{\pi_x}\, |x\rangle$$

$$|M\rangle = \frac{1}{\sqrt{p_M}} \sum_{x \in M} \sqrt{\pi_x}\, |x\rangle$$

Algorithm

1. Prepare the state $|v_n(0)\rangle |0\rangle$.
2. For $s^* = 1 - p_M/(1 - p_M)$ and $T = \Theta(\sqrt{HT^+(P, M)})$, evolve according to $H(s^*)$ for a time chosen uniformly at random between $[0, T]$.
3. Measure in the basis spanned by the state space, in the first register.

Algorithm

1. Prepare the state $|v_n(0)\rangle |0\rangle$.
2. For $s^* = 1 - p_M/(1 - p_M)$ and $T = \Theta(\sqrt{HT^+(P, M)})$, evolve according to $H(s^*)$ for a time chosen uniformly at random between $[0, T]$.
3. Measure in the basis spanned by the state space, in the first register.

Proof idea: Quantum phase randomization [Boixo, Knill and Somma 2009]

- Decoherence in the eigenbasis of $H(s^*)$ kills coherence terms between $|v_n(s^*)\rangle$ and its orthogonal eigenstates.

Algorithm

1. Prepare the state $|v_n(0)\rangle\,|0\rangle$.
2. For $s^* = 1 - p_M/(1-p_M)$ and $T = \Theta(\sqrt{HT^+(P,M)})$, evolve according to $H(s^*)$ for a time chosen uniformly at random between $[0, T]$.
3. Measure in the basis spanned by the state space, in the first register.

Proof idea: Quantum phase randomization [Boixo, Knill and Somma 2009]

- Decoherence in the eigenbasis of $H(s^*)$ kills coherence terms between $|v_n(s^*)\rangle$ and its orthogonal eigenstates.
- We obtain a mixed state between $|v_n(s^*)\rangle$ and the rest.

Algorithm

1. Prepare the state $|v_n(0)\rangle\,|0\rangle$.
2. For $s^* = 1 - p_M/(1-p_M)$ and $T = \Theta(\sqrt{HT^+(P,M)})$, evolve according to $H(s^*)$ for a time chosen uniformly at random between $[0, T]$.
3. Measure in the basis spanned by the state space, in the first register.

Proof idea: Quantum phase randomization [Boixo, Knill and Somma 2009]

- Decoherence in the eigenbasis of $H(s^*)$ kills coherence terms between $|v_n(s^*)\rangle$ and its orthogonal eigenstates.
- We obtain a mixed state between $|v_n(s^*)\rangle$ and the rest.
- Expected running time: T

Spatial search by CTQW (Childs and Goldstone 2004)

Spatial search by CTQW (Childs and Goldstone 2004)

Hilbert space spanned by the nodes of a graph G $\{|1\rangle, \ldots, |n\rangle\}$

Oracle Hamiltonian: $H_{oracle} = |w\rangle\langle w|$.

Dynamics is driven by the Hamiltonian $H_1 = A$, where A is the adjacency matrix of the graph.

Spatial search by CTQW (Childs and Goldstone 2004)

Hilbert space spanned by the nodes of a graph G $\{|1\rangle, \ldots, |n\rangle\}$

Oracle Hamiltonian: $H_{oracle} = |w\rangle \langle w|$.

Dynamics is driven by the Hamiltonian $H_1 = A$, where A is the adjacency matrix of the graph.

Search Hamiltonian: $H = H_{oracle} + rH_1$ [r is a real number which we are free to choose.]

Task: Find $|w\rangle$.

Given a graph G, algorithm involves the following steps:

1. Prepare the initial state to be the equal superposition of all nodes of G. Thus $|s\rangle = 1/\sqrt{n}\sum_i |i\rangle$.

Spatial search by CTQW (Childs and Goldstone 2004)

Hilbert space spanned by the nodes of a graph G $\{|1\rangle, \ldots, |n\rangle\}$

Oracle Hamiltonian: $H_{oracle} = |w\rangle\langle w|$.

Dynamics is driven by the Hamiltonian $H_1 = A$, where A is the adjacency matrix of the graph.

Search Hamiltonian: $H = H_{oracle} + rH_1$ [r is a real number which we are free to choose.]

Task: Find $|w\rangle$.

Given a graph G, algorithm involves the following steps:

1. Prepare the initial state to be the equal superposition of all nodes of G. Thus $|s\rangle = 1/\sqrt{n}\sum_i |i\rangle$.
2. Measure after $T = \mathcal{O}(\sqrt{n})$ in the basis of the nodes.

Spatial search by CTQW (Childs and Goldstone 2004)

Hilbert space spanned by the nodes of a graph G $\{|1\rangle, \ldots, |n\rangle\}$
Oracle Hamiltonian: $H_{oracle} = |w\rangle\langle w|$.

Dynamics is driven by the Hamiltonian $H_1 = A$, where A is the adjacency matrix of the graph.

Search Hamiltonian: $H = H_{oracle} + rH_1$ [r is a real number which we are free to choose.]

Task: Find $|w\rangle$.

Given a graph G, algorithm involves the following steps:

1. Prepare the initial state to be the equal superposition of all nodes of G. Thus $|s\rangle = 1/\sqrt{n}\sum_i |i\rangle$.
2. Measure after $T = \mathcal{O}(\sqrt{n})$ in the basis of the nodes. If $P_w(T) = |\langle w| e^{-iAT} |s\rangle|^2 = \Theta(1) \implies$ **Optimal for G!**

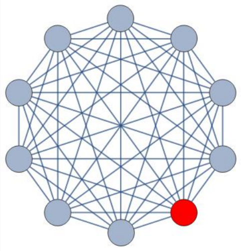
Complete graph:

$$A_{ij} = 1, i \neq j$$

Same as analog Grover

$$\begin{aligned} H &= -|w\rangle\langle w| - |s\rangle\langle s| \\ &= -|w\rangle\langle w| - \frac{1}{n}A \end{aligned}$$

Optimal! $T = O(\sqrt{n})$



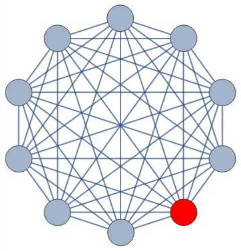
Complete graph:

$$A_{ij} = 1, i \neq j$$

Same as analog Grover

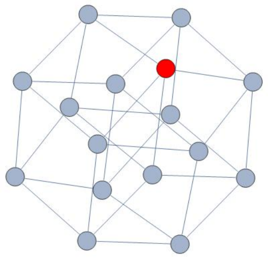
$$H = -|w\rangle\langle w| - |s\rangle\langle s|$$
$$= -|w\rangle\langle w| - \frac{1}{n}A$$

Optimal! $T = O(\sqrt{n})$



Hypercube:

Optimal! $T = O(\sqrt{n})$



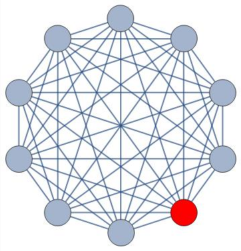
Complete graph:

$$A_{ij} = 1, i \neq j$$

Same as analog Grover

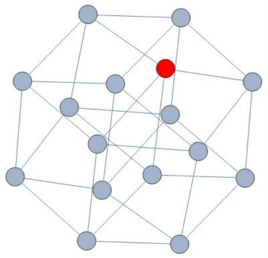
$$H = -\left|w\right\rangle \left\langle w\right| - \left|s\right\rangle \left\langle s\right|$$
$$= -\left|w\right\rangle \left\langle w\right| - \frac{1}{n} A$$

Optimal! $T = O(\sqrt{n})$



Hypercube:

Optimal! $T = O(\sqrt{n})$



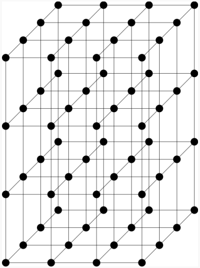
Square Lattices:

$d \leq 3$
No significant speed-up

$d = 4$

$$T = O(\sqrt{n} \log n)$$

$d > 4$
Optimal! $T = O(\sqrt{n})$



Sufficient condition for the optimality of $\mathcal{CG}$ algorithm

Let H_G be a Hamiltonian with eigenvalues

$$\lambda_n = 1 > \lambda_{n-1} = 1 - \Delta \geq \ldots \geq \lambda_1 \geq 0$$

$(\Delta > 0)$ such that $H_G \left|v_i\right\rangle = \lambda_i \left|v_i\right\rangle$.

Sufficient condition for the optimality of $\mathcal{CG}$ algorithm

Let H_G be a Hamiltonian with eigenvalues

$$\lambda_n = 1 > \lambda_{n-1} = 1 - \Delta \geq \ldots \geq \lambda_1 \geq 0$$

($\Delta > 0$) such that $H_G |v_i\rangle = \lambda_i |v_i\rangle$.

Search Hamiltonian: $H_{\text{search}} = |w\rangle\langle w| + rH_G$.

Let $|w\rangle = \sum_i a_i |v_i\rangle$ and $|\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$.

Sufficient condition for the optimality of $\mathcal{CG}$ algorithm

Let H_G be a Hamiltonian with eigenvalues

$$\lambda_n = 1 > \lambda_{n-1} = 1 - \Delta \geq \ldots \geq \lambda_1 \geq 0$$

($\Delta > 0$) such that $H_G |v_i\rangle = \lambda_i |v_i\rangle$.

Search Hamiltonian: $H_{\text{search}} = |w\rangle\langle w| + rH_G$.

Let $|w\rangle = \sum_i a_i |v_i\rangle$ and $|\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$.

$$\text{Optimal } r = \sum_{i\neq n} \frac{|a_i|^2}{1-\lambda_i} \text{ and Max. amplitude } \nu = \frac{\sum_{i\neq n} \frac{|a_i|^2}{1-\lambda_i}}{\sqrt{\sum_{i\neq n} \frac{|a_i|^2}{(1-\lambda_i)^2}}}.$$

Sufficient condition for the optimality of $\mathcal{CG}$ algorithm

Let H_G be a Hamiltonian with eigenvalues

$$\lambda_n = 1 > \lambda_{n-1} = 1 - \Delta \geq \ldots \geq \lambda_1 \geq 0$$

$(\Delta > 0)$ such that $H_G |v_i\rangle = \lambda_i |v_i\rangle$.

Search Hamiltonian: $H_{\mathrm{search}} = |w\rangle\langle w| + rH_G$.

Let $|w\rangle = \sum_i a_i |v_i\rangle$ and $|\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$.

$$\text{Optimal } r = \sum_{i\neq n} \frac{|a_i|^2}{1-\lambda_i} \text{ and Max. amplitude } \nu = \frac{\sum_{i\neq n} \frac{|a_i|^2}{1-\lambda_i}}{\sqrt{\sum_{i\neq n} \frac{|a_i|^2}{(1-\lambda_i)^2}}}.$$

Restriction: $\sqrt{\epsilon} \ll r\Delta/\nu$, **Initial state:** $|v_n\rangle$, **Running time**: $T = \Theta\left(\frac{1}{\sqrt{\epsilon}\nu}\right)$.

Final state: $|f\rangle$ such that $|\langle w|f\rangle| \approx \nu$.

Sufficient condition for the optimality of $\mathcal{CG}$ algorithm

Let H_G be a Hamiltonian with eigenvalues

$$\lambda_n = 1 > \lambda_{n-1} = 1 - \Delta \geq \ldots \geq \lambda_1 \geq 0$$

$(\Delta > 0)$ such that $H_G \left|v_i\right\rangle = \lambda_i \left|v_i\right\rangle$.

Search Hamiltonian: $H_{\mathrm{search}} = \left|w\right\rangle\left\langle w\right| + rH_G$.

Let $\left|w\right\rangle = \sum_i a_i \left|v_i\right\rangle$ and $|\left\langle w|v_n\right\rangle| = |a_n| = \sqrt{\epsilon}$.

$$\text{Optimal } r = \sum_{i \neq n} \frac{|a_i|^2}{1-\lambda_i} \text{ and Max. amplitude } \nu = \frac{\sum_{i\neq n} \frac{|a_i|^2}{1-\lambda_i}}{\sqrt{\sum_{i\neq n} \frac{|a_i|^2}{(1-\lambda_i)^2}}}.$$

Restriction: $\sqrt{\epsilon} \ll r\Delta/\nu$, **Initial state:** $\left|v_n\right\rangle$, **Running time**: $T = \Theta\left(\frac{1}{\sqrt{\epsilon}\nu}\right)$.

Final state: $\left|f\right\rangle$ such that $|\left\langle w|f\right\rangle| \approx \nu$.

Apply $\mathcal{O}(1/\nu)$ rounds of amplitude amplification: $T_{search} = \mathcal{O}\left(\frac{1}{\nu^2\sqrt{\epsilon}}\right)$.

$$|\langle w|f\rangle| \approx \nu, \ |\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$$

$$T_{search} = \mathcal{O}\left(\frac{1}{\nu^2\sqrt{\epsilon}}\right).$$

$$|\langle w|f\rangle| \approx \nu, \; |\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$$

$$T_{search} = \mathcal{O}\left(\frac{1}{\nu^2\sqrt{\epsilon}}\right).$$

Whenever $\nu = \Theta(1)$, $T_{search} = \mathcal{O}(1/\sqrt{\epsilon})$.

$|\langle w|f\rangle| \approx \nu, \ |\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$

$$T_{search} = \mathcal{O}\left(\frac{1}{\nu^2\sqrt{\epsilon}}\right).$$

Whenever $\nu = \Theta(1), \ T_{search} = \mathcal{O}(1/\sqrt{\epsilon})$.

- When $\Delta = \Theta(1), \ \nu = \Theta(1)$
  - H_G is the adjacency matrix of a regular graph ($\epsilon = 1/n$) with $\Delta = \Theta(1)$ [C, Novo, Ambainis and Omar 2016].

$|\langle w|f\rangle| \approx \nu, \; |\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$

$$T_{search} = \mathcal{O}\left(\frac{1}{\nu^2\sqrt{\epsilon}}\right).$$

Whenever $\nu = \Theta(1)$, $T_{search} = \mathcal{O}(1/\sqrt{\epsilon})$.

- When $\Delta = \Theta(1)$, $\nu = \Theta(1)$
  - H_G is the adjacency matrix of a regular graph ($\epsilon = 1/n$) with $\Delta = \Theta(1)$ [C, Novo, Ambainis and Omar 2016]. **Optimal!**

$| \langle w|f \rangle | \approx \nu, \; | \langle w|v_n \rangle | = |a_n| = \sqrt{\epsilon}$

$$T_{search} = \mathcal{O}\left(\frac{1}{\nu^2\sqrt{\epsilon}}\right).$$

Whenever $\nu = \Theta(1), \; T_{search} = \mathcal{O}(1/\sqrt{\epsilon})$.

- When $\Delta = \Theta(1), \; \nu = \Theta(1)$
  - H_G is the adjacency matrix of a regular graph ($\epsilon = 1/n$) with $\Delta = \Theta(1)$ [C, Novo, Ambainis and Omar 2016]. **Optimal!**
- ν can be $\Theta(1)$ even when $\Delta \neq \Theta(1)$
  - For d-dimensional lattices with $d > 4$, $\Delta \sim n^{-2/d}$ but $\nu = \Theta(1)$.

$$|\langle w|f\rangle| \approx \nu, \ |\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$$

$$T_{search} = \mathcal{O}\left(\frac{1}{\nu^2\sqrt{\epsilon}}\right).$$

Whenever $\nu = \Theta(1)$, $T_{search} = \mathcal{O}(1/\sqrt{\epsilon})$.

- When $\Delta = \Theta(1)$, $\nu = \Theta(1)$
  - H_G is the adjacency matrix of a regular graph ($\epsilon = 1/n$) with $\Delta = \Theta(1)$ [C, Novo, Ambainis and Omar 2016]. **Optimal!**
- ν can be $\Theta(1)$ even when $\Delta \neq \Theta(1)$
  - For d-dimensional lattices with $d > 4$, $\Delta \sim n^{-2/d}$ but $\nu = \Theta(1)$. **Optimal!**
  - At $d = 4$, $\nu = \Theta(1/\sqrt{\log n}) \implies T_{search} = \mathcal{O}(\sqrt{n}\log n)$.

$|\langle w|f\rangle| \approx \nu,\ |\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$

$$T_{search} = \mathcal{O}\left(\frac{1}{\nu^2\sqrt{\epsilon}}\right).$$

Whenever $\nu = \Theta(1)$, $T_{search} = \mathcal{O}(1/\sqrt{\epsilon})$.

- When $\Delta = \Theta(1)$, $\nu = \Theta(1)$
  - H_G is the adjacency matrix of a regular graph ($\epsilon = 1/n$) with $\Delta = \Theta(1)$ [C, Novo, Ambainis and Omar 2016]. **Optimal!**
- ν can be $\Theta(1)$ even when $\Delta \neq \Theta(1)$
  - For d-dimensional lattices with $d > 4$, $\Delta \sim n^{-2/d}$ but $\nu = \Theta(1)$. **Optimal!**
  - At $d = 4$, $\nu = \Theta(1/\sqrt{\log n}) \implies T_{search} = \mathcal{O}(\sqrt{n}\log n)$.
- Algorithm is optimal for any symmetric P having $\nu = \Theta(1)$.

Drawbacks of the $\mathcal{CG}$ algorithm

$r = \sum_{i \neq n} \frac{|\langle w|v_i\rangle|^2}{1-\lambda_i}, \; |\langle w|f\rangle| \approx \nu, \; |\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$

$$T_{search} = \mathcal{O}\left(\frac{1}{\nu^2\sqrt{\epsilon}}\right).$$

- Predicting r requires knowledge of solution unless P is also state-transitive.

Drawbacks of the $\mathcal{CG}$ algorithm

$r = \sum_{i \neq n} \frac{|\langle w|v_i\rangle|^2}{1-\lambda_i}, \ |\langle w|f\rangle| \approx \nu, \ |\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$

$$T_{search} = \mathcal{O}\left(\frac{1}{\nu^2\sqrt{\epsilon}}\right).$$

- Predicting r requires knowledge of solution unless P is also state-transitive.
- The condition $\sqrt{\epsilon} \ll r\Delta/\nu$ needs to be satisfied.

Drawbacks of the $\mathcal{CG}$ algorithm

$r = \sum_{i \neq n} \frac{|\langle w|v_i \rangle|^2}{1-\lambda_i}, \ |\langle w|f\rangle| \approx \nu, \ |\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$

$$T_{search} = \mathcal{O}\left(\frac{1}{\nu^2\sqrt{\epsilon}}\right).$$

- Predicting r requires knowledge of solution unless P is also state-transitive.
- The condition $\sqrt{\epsilon} \ll r\Delta/\nu$ needs to be satisfied.
    - Does not hold for lattices of dimension less than four.

Drawbacks of the $\mathcal{CG}$ algorithm

$$r = \sum_{i \neq n} \frac{|\langle w|v_i\rangle|^2}{1-\lambda_i}, \ |\langle w|f\rangle| \approx \nu, \ |\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$$

$$T_{search} = \mathcal{O}\left(\frac{1}{\nu^2\sqrt{\epsilon}}\right).$$

- Predicting r requires knowledge of solution unless P is also state-transitive.
- The condition $\sqrt{\epsilon} \ll r\Delta/\nu$ needs to be satisfied.
  - Does not hold for lattices of dimension less than four.
- We show that $\Omega(\sqrt{\Delta}) < \nu < 1$.
  - $HT(P, w) \leq \frac{1}{\Delta\epsilon}$.

Drawbacks of the $\mathcal{CG}$ algorithm

$r = \sum_{i \neq n} \frac{|\langle w|v_i\rangle|^2}{1-\lambda_i}, \ |\langle w|f\rangle| \approx \nu, \ |\langle w|v_n\rangle| = |a_n| = \sqrt{\epsilon}$

$$T_{search} = \mathcal{O}\left(\frac{1}{\nu^2\sqrt{\epsilon}}\right).$$

- Predicting r requires knowledge of solution unless P is also state-transitive.
- The condition $\sqrt{\epsilon} \ll r\Delta/\nu$ needs to be satisfied.
  - Does not hold for lattices of dimension less than four.
- We show that $\Omega(\sqrt{\Delta}) < \nu < 1$.
  - $HT(P, w) \leq \frac{1}{\Delta\epsilon}$. So when $\nu = \sqrt{\Delta} \implies$ Sub-optimal!
  - T_{search} is sub-optimal even for state-transitive P!

Example: Movement of rook on a rectangular chessboard

$\mathcal{CG}$ algorithm on any ergodic, reversible Markov chain

Given any ergodic, reversible P, use the Somma-Ortiz Hamiltonian, i.e. $H_1 = i[V^\dagger S V, \Pi_0]$.

$\mathcal{CG}$ algorithm on any ergodic, reversible Markov chain

Given any ergodic, reversible P, use the Somma-Ortiz Hamiltonian, i.e. $H_1 = i[V^\dagger S V, \Pi_0]$.

One possibility: $|w,0\rangle\langle w,0| + rH_1$.

$\mathcal{CG}$ algorithm on any ergodic, reversible Markov chain

Given any ergodic, reversible P, use the Somma-Ortiz Hamiltonian, i.e. $H_1 = i[V^\dagger S V, \Pi_0]$.

One possibility: $|w,0\rangle \langle w,0| + rH_1$. Does not work as critical value of $r = 0$.

$\mathcal{CG}$ algorithm on any ergodic, reversible Markov chain

Given any ergodic, reversible P, use the Somma-Ortiz Hamiltonian, i.e. $H_1 = i[V^\dagger S V, \Pi_0]$.

One possibility: $|w,0\rangle\langle w,0| + rH_1$. Does not work as critical value of $r = 0$.

Alternatively: Use a different oracle Hamiltonian [Foulger et al. 2014, Childs and Ge 2014]

$$H_{oracle} = -|w,0\rangle\langle w,0| H_1 - H_1 |w,0\rangle\langle w,0|.$$

$\mathcal{CG}$ algorithm on any ergodic, reversible Markov chain

Given any ergodic, reversible P, use the Somma-Ortiz Hamiltonian, i.e. $H_1 = i[V^\dagger S V, \Pi_0]$.

One possibility: $|w,0\rangle \langle w,0| + rH_1$. Does not work as critical value of $r = 0$.

Alternatively: Use a different oracle Hamiltonian [Foulger et al. 2014, Childs and Ge 2014]

$$H_{oracle} = -|w,0\rangle \langle w,0| H_1 - H_1 |w,0\rangle \langle w,0| .$$

Example: If H_1 is the adjacency matrix of the graph, H_{oracle} removes the edges connected to the marked node.

$\mathcal{CG}$ algorithm on any ergodic, reversible Markov chain

Given any ergodic, reversible P, use the Somma-Ortiz Hamiltonian, i.e. $H_1 = i[V^\dagger S V, \Pi_0]$.

One possibility: $|w,0\rangle\langle w,0| + rH_1$. Does not work as critical value of $r = 0$.

Alternatively: Use a different oracle Hamiltonian [Foulger et al. 2014, Childs and Ge 2014]

$$H_{oracle} = -|w,0\rangle\langle w,0| H_1 - H_1 |w,0\rangle\langle w,0|.$$

Example: If H_1 is the adjacency matrix of the graph, H_{oracle} removes the edges connected to the marked node.

Use the search Hamiltonian

$$H_{search} = -|w,0\rangle\langle w,0| H_1 - H_1 |w,0\rangle\langle w,0| + H_1$$

to rotate from $|v_n, 0\rangle$ to the state

$$|\widetilde{w}\rangle = \frac{H_1 |w\rangle}{||H_1 |w\rangle ||}.$$

- Amplitude amplification results in the state $|\widetilde{w}\rangle$ and so

$$T_{search} = \mathcal{O}\left(\mu^2/\sqrt{\epsilon}\right).$$

- Amplitude amplification results in the state $|\widetilde{w}\rangle$ and so

$$T_{search} = \mathcal{O}\left(\mu^2/\sqrt{\epsilon}\right).$$

- Also $e^{-it'H_2}|\widetilde{w}\rangle = |w, 0\rangle$, where $t' = \mathcal{O}(\mu)$. So $\mathcal{O}(\mu)$ additional queries to the oracle are needed to obtain $|w\rangle$.

- Amplitude amplification results in the state $|\widetilde{w}\rangle$ and so

$$T_{search} = \mathcal{O}\left(\mu^2/\sqrt{\epsilon}\right).$$

- Also $e^{-it' H_{oracle}} |\widetilde{w}\rangle = |w, 0\rangle$, where $t' = \mathcal{O}(\mu)$. So $\mathcal{O}(\mu)$ additional queries to the oracle are needed to obtain $|w\rangle$.

- Amplitude amplification results in the state $|\widetilde{w}\rangle$ and so

$$T_{search} = \mathcal{O}\left(\mu^2/\sqrt{\epsilon}\right).$$

- Also $e^{-it' H_{oracle}} |\widetilde{w}\rangle = |w, 0\rangle$, where $t' = \mathcal{O}(\mu)$. So $\mathcal{O}(\mu)$ additional queries to the oracle are needed to obtain $|w\rangle$.

Improvement over the $\mathcal{CG}$ algorithm

- Whenever $\mathcal{CG}$ algorithm is optimal, so is $\mathcal{CG}'$.

- Amplitude amplification results in the state $|\widetilde{w}\rangle$ and so
$$T_{search} = \mathcal{O}\left(\mu^2/\sqrt{\epsilon}\right).$$
- Also $e^{-it' H_{oracle}} |\widetilde{w}\rangle = |w, 0\rangle$, where $t' = \mathcal{O}(\mu)$. So $\mathcal{O}(\mu)$ additional queries to the oracle are needed to obtain $|w\rangle$.

Improvement over the $\mathcal{CG}$ algorithm

- Whenever $\mathcal{CG}$ algorithm is optimal, so is $\mathcal{CG}'$.
- For $2d$-lattices, $\mathcal{CG}'$ algorithm has a running time of $T_{search} = \mathcal{O}(\sqrt{n} \log n)$.

- Amplitude amplification results in the state $|\widetilde{w}\rangle$ and so

$$T_{search} = \mathcal{O}\left(\mu^2/\sqrt{\epsilon}\right).$$

- Also $e^{-it' H_{oracle}} |\widetilde{w}\rangle = |w, 0\rangle$, where $t' = \mathcal{O}(\mu)$. So $\mathcal{O}(\mu)$ additional queries to the oracle are needed to obtain $|w\rangle$.

Improvement over the $\mathcal{CG}$ algorithm

- Whenever $\mathcal{CG}$ algorithm is optimal, so is $\mathcal{CG}'$.
- For 2d-lattices, $\mathcal{CG}'$ algorithm has a running time of $T_{search} = \mathcal{O}(\sqrt{n} \log n)$. **Our algorithm:** $\mathcal{O}(\sqrt{n \log n})$.

Conclusions

- Spatial search algorithm by CTQW that runs in $\Theta(\sqrt{HT^+(P,M)})$ time on any ergodic, reversible Markov chain.
- Provided general conditions for the optimality of the spatial search algorithm by Childs and Goldstone
  - Applicable to only state-transitive Markov chains.
  - Sub-optimal.
- Improved the Childs and Goldstone algorithm to be applicable to any ergodic, reversible Markov chain.

Open questions:

Extended hitting time vs hitting time

Open questions:

Extended hitting time vs hitting time

Future work:

Open questions:

Extended hitting time vs hitting time

Future work:

Quantum algorithms for preparing the stationary state of an ergodic, reversible Markov chain.

Thank you for your attention!

For more details see:

S. Chakraborty, L. Novo and J. Roland, Finding a marked node on any graph by continuous time quantum walk, arXiv:1807.05957 (2018).