ELFS, TREES AND QUANTUM WALKS

Simon Apers
(CNRS & IRIF, Paris)

with **Stephen Piddock** (Royal Holloway, London)

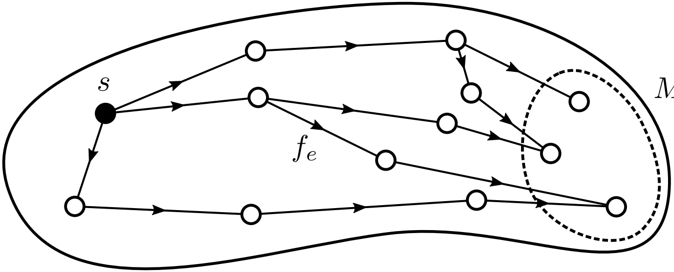
Sublinear algorithms, Bernoulli center (EPFL), December ’22

ELFS, TREES AND QUANTUM WALKS

simple graph $G = (V, E)$

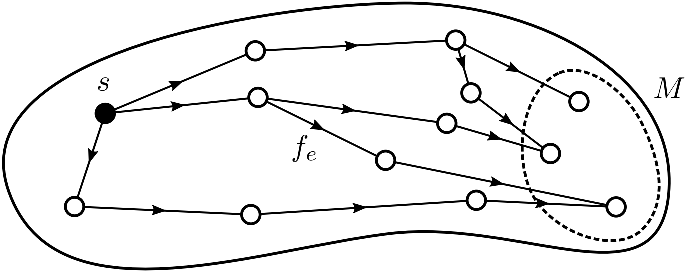
simple graph $G = (V, E)$

unit electric flow f from source s to sink M



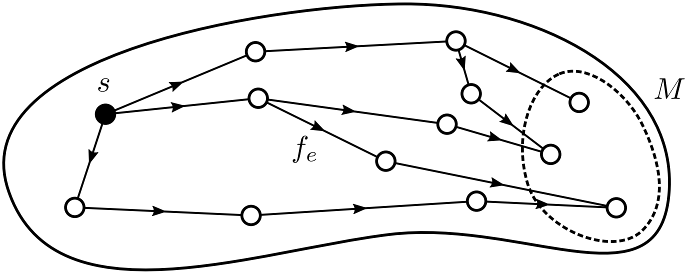
simple graph $G = (V, E)$

unit electric flow f from source s to sink M



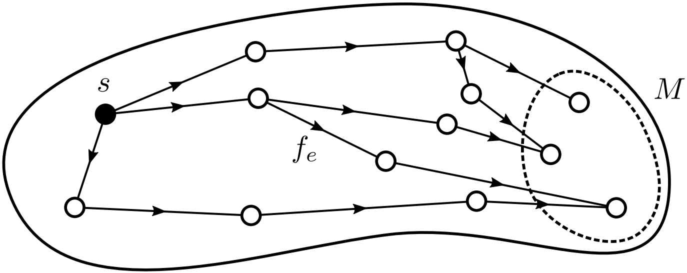
effective resistance $R = \sum_e f_e^2$

Electric flow sampling (elfs) process:



Electric flow sampling (elfs) process: repeat

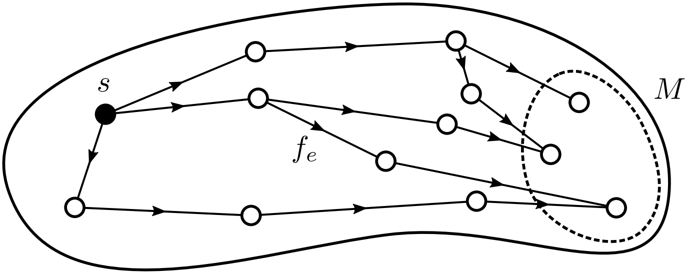
1. sample edge e with probability $p_e = f_e^2/R$



Electric flow sampling (elfs) process: repeat

1. sample edge e with probability $p_e = f_e^2/R$

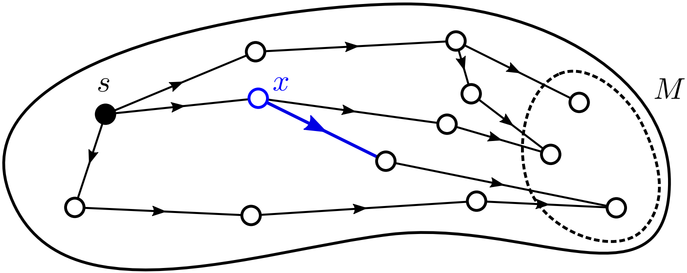
2. pick random endpoint x from e, set source $s = x$



Electric flow sampling (elfs) process: repeat

1. sample edge e with probability $p_e = f_e^2/R$

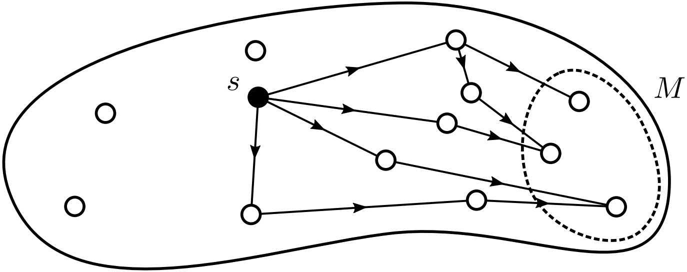
2. pick random endpoint x from e, set source $s = x$



Electric flow sampling (elfs) process: repeat

1. sample edge e with probability $p_e = f_e^2/R$

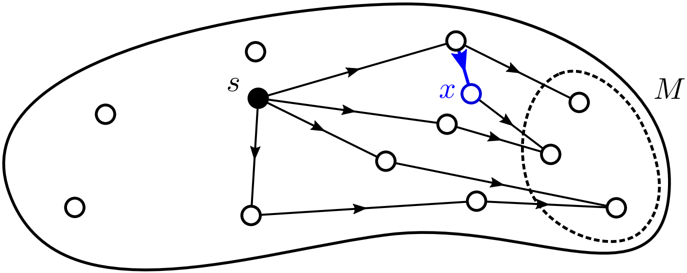
2. pick random endpoint x from e, set source $s = x$



Electric flow sampling (elfs) process: repeat

1. sample edge e with probability $p_e = f_e^2/R$

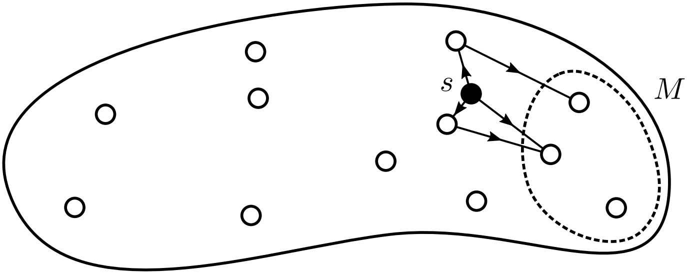
2. pick random endpoint x from e, set source $s = x$



Electric flow sampling (elfs) process: repeat

1. sample edge e with probability $p_e = f_e^2/R$

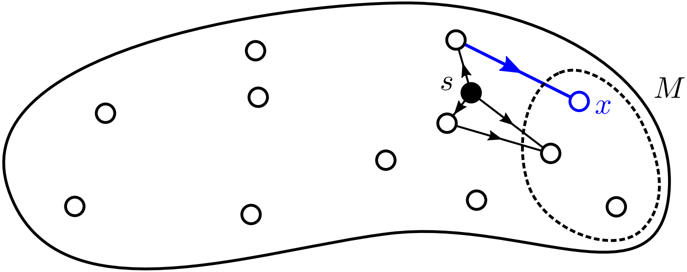
2. pick random endpoint x from e, set source $s = x$



Electric flow sampling (elfs) process: repeat

1. sample edge e with probability $p_e = f_e^2/R$

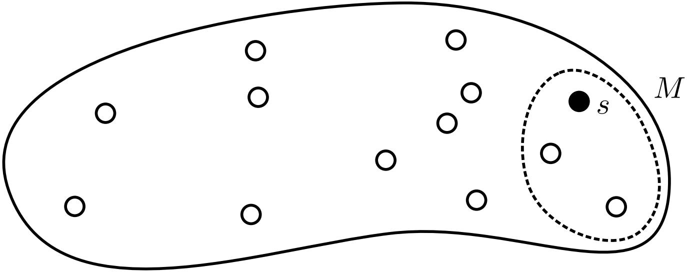
2. pick random endpoint x from e, set source $s = x$



Electric flow sampling (elfs) process: repeat

1. sample edge e with probability $p_e = f_e^2/R$

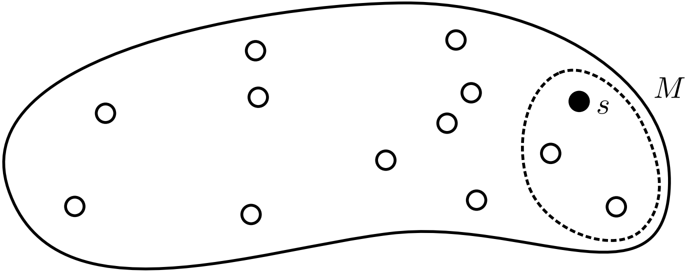
2. pick random endpoint x from e, set source $s = x$



Electric flow sampling (elfs) process: repeat

1. sample edge e with probability $p_e = f_e^2/R$

2. pick random endpoint x from e, set source $s = x$



process ends when step 2. picks sink vertex

Results

Results

- **elfs**

coupling with random walk

Results

- **elfs**

coupling with random walk

- **trees**

electric hitting time is $O(\log n)$

Results

- **elfs**

coupling with random walk

- **trees**

electric hitting time is $O(\log n)$

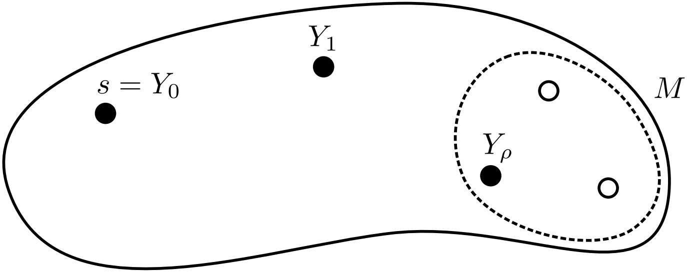
- **quantum walks**

quantum algorithm for elfs

elfs process describes Markov chain

$$\{Y_0 = s, Y_1, Y_2, \ldots, Y_\rho \in M\}$$

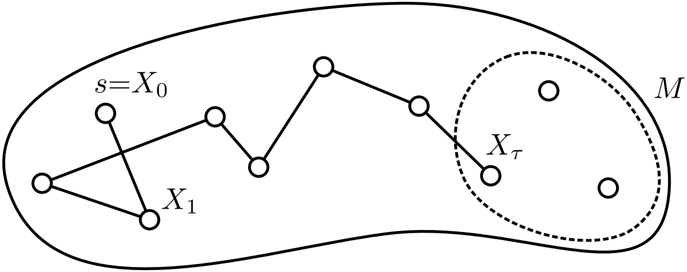
with ρ the “electric hitting time”



random walk

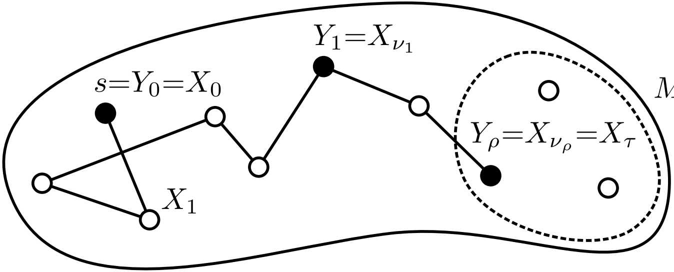
$$\{X_0 = s, X_1, X_2, \ldots, X_\tau \in M\}$$

with τ the random walk hitting time



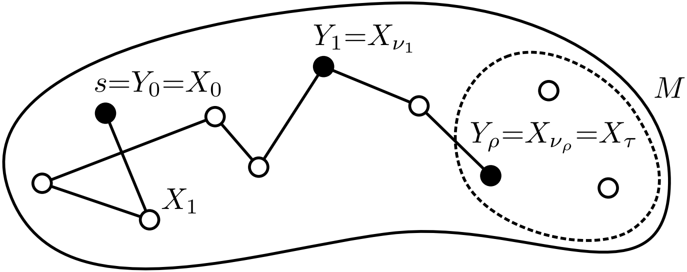
lemma: exists stopping times $0 < \nu_1 < \cdots < \nu_\rho = \tau$ such that

$$\{Y_0 = X_0, \quad Y_1 = X_{\nu_1}, \quad \ldots, \quad Y_\rho = X_{\nu_\rho} = X_\tau\}$$



lemma: exists stopping times $0 < \nu_1 < \cdots < \nu_\rho = \tau$ such that

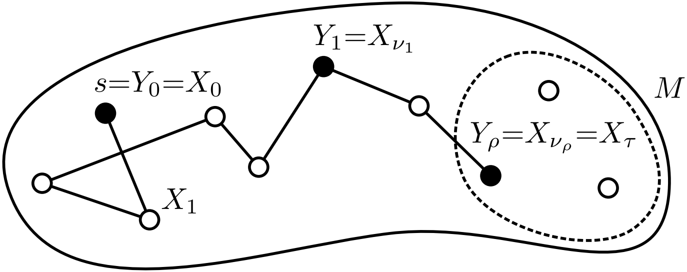
$$\{Y_0 = X_0,\ \ Y_1 = X_{\nu_1},\ \ \ldots,\ \ Y_\rho = X_{\nu_\rho} = X_\tau\}$$



i.e., exists coupling between random walk and elfs process

lemma: exists stopping times $0 < \nu_1 < \cdots < \nu_\rho = \tau$ such that

$$\{Y_0 = X_0,\ \ Y_1 = X_{\nu_1},\ \ \ldots,\ \ Y_\rho = X_{\nu_\rho} = X_\tau\}$$



corollary: random walk and elfs process same arrival distribution

$$\Pr(Y_\rho = x) = \Pr(X_\tau = x) = \sum_y f_{y,x}, \qquad x \in M$$

stopping rule ν from s: when visiting vertex x, stop with probability[1]

$$p_x = \frac{\sum_y (v_x - v_y)^2}{v_x d_x + \sum_y (v_x - v_y)^2}$$

[1] v_z = voltage at z in unit electric s-M flow

stopping rule ν from s: when visiting vertex x, stop with probability[1]

$$p_x = \frac{\sum_y (v_x - v_y)^2}{v_x d_x + \sum_y (v_x - v_y)^2}$$

e.g., $p_x = 1$ for $x \in M$

[1] v_z = voltage at z in unit electric s-M flow

stopping rule ν from s: when visiting vertex x, stop with probability[1]

$$p_x = \frac{\sum_y (v_x - v_y)^2}{v_x d_x + \sum_y (v_x - v_y)^2}$$

expected stopping time $\mathbb{E}[\nu]$**?**

[1] v_z = voltage at z in unit electric s-M flow

stopping rule ν from s: when visiting vertex x, stop with probability[1]

$$p_x = \frac{\sum_y (v_x - v_y)^2}{v_x d_x + \sum_y (v_x - v_y)^2}$$

lemma: $$\mathbb{E}[\nu] = \frac{1}{R} \sum_x v_x^2 d_x =: \mathrm{ET}_s$$

[1] v_z = voltage at z in unit electric s-M flow

stopping rule ν from s: when visiting vertex x, stop with probability[1]

$$p_x = \frac{\sum_y (v_x - v_y)^2}{v_x d_x + \sum_y (v_x - v_y)^2}$$

lemma: $$\mathbb{E}[\nu] = \frac{1}{R} \sum_x v_x^2 d_x =: \mathrm{ET}_s$$

using $v_x \leq v_s = R$:
$$\begin{aligned} 1 &\leq Rd_s \quad \left(= v_s d_s = 1/\Pr_s(\tau_M < \tau_s^+)\right) \\ &\leq \mathrm{ET}_s \\ &\leq \mathrm{HT}_s \quad \left(= \sum_x v_x d_x\right) \end{aligned}$$

[1] v_z = voltage at z in unit electric s-M flow

stopping rule ν from s: when visiting vertex x, stop with probability[1]

$$p_x = \frac{\sum_y (v_x - v_y)^2}{v_x d_x + \sum_y (v_x - v_y)^2}$$

lemma: $\mathbb{E}[\nu] = \frac{1}{R} \sum_x v_x^2 d_x =: \mathrm{ET}_s$

lemma': $\mathrm{ET}_s =$ "escape time" from s to M

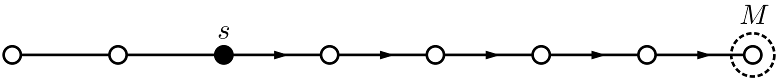
$$= \mathbb{E}\{1 + \max\{t \mid X_t = s\}\}$$

[1] v_z = voltage at z in unit electric s-M flow

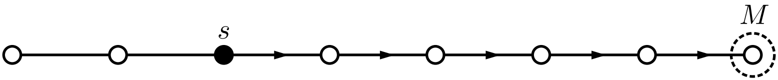
ELFS, **TREES** AND QUANTUM WALKS

? electric hitting time $\mathrm{EHT}_s = \mathbb{E}[\rho]$ **?**

? electric hitting time $\mathrm{EHT}_s = \mathbb{E}[\rho]$ **?**

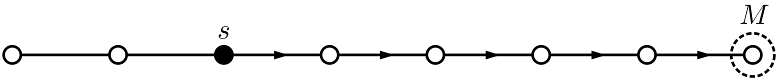


? electric hitting time $\mathrm{EHT}_s = \mathbb{E}[\rho]$?

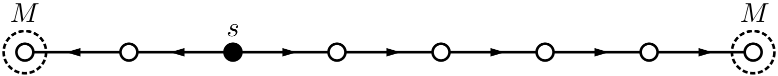


$$\mathrm{EHT}_s \sim \log n$$

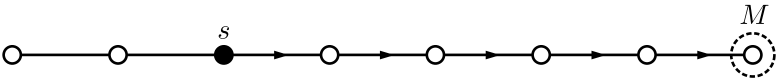
? electric hitting time $\mathrm{EHT}_s = \mathbb{E}[\rho]$?



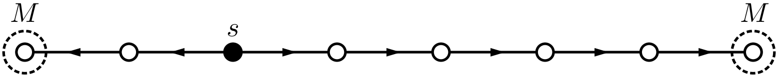
$$\mathrm{EHT}_s \sim \log n$$



? electric hitting time $\mathrm{EHT}_s = \mathbb{E}[\rho]$ **?**

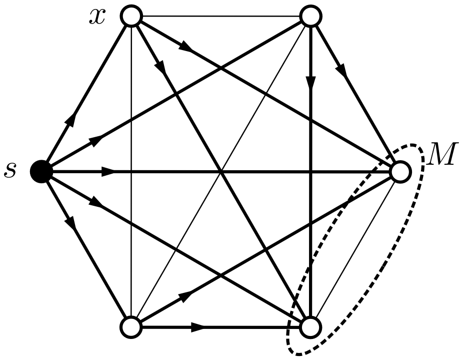


$$\mathrm{EHT}_s \sim \log n$$

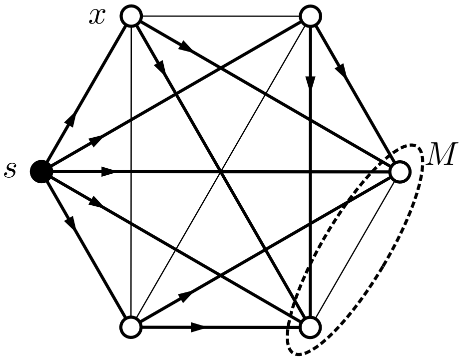


$$\mathrm{EHT}_s \sim ?$$

? electric hitting time $\mathrm{EHT}_s = \mathbb{E}[\rho]$ **?**



? electric hitting time $\mathrm{EHT}_s = \mathbb{E}[\rho]$ **?**



$$\mathrm{EHT}_s \sim \min\{|M|, n/|M|\}$$

theorem: on trees, $\mathrm{EHT}_s \in O(\log n)$

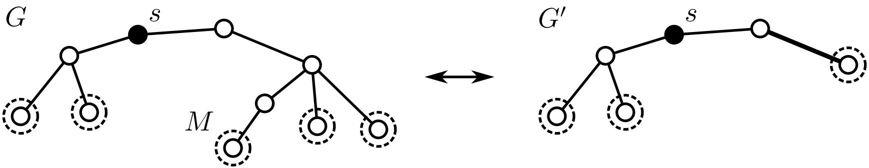
theorem: on trees, $\mathrm{EHT}_s \in O(\log n)$

proof via divide-and-conquer

theorem: on trees, $\mathrm{EHT}_s \in O(\log n)$

proof via divide-and-conquer

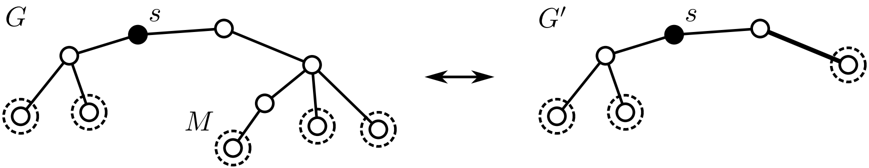
- divide via Schur complement for elfs process on trees



theorem: on trees, $\mathrm{EHT}_s \in O(\log n)$

proof via divide-and-conquer

- divide via Schur complement for elfs process on trees



- conquer via bottoms-up:

steps below vertex = f(# steps below children)

ELFS, TREES AND **QUANTUM WALKS**

quantum walks

quantum walks

=

algorithmic tool in quantum computing

(element distinctness, search, Monte Carlo)

quantum walks

=

algorithmic tool in quantum computing
(element distinctness, search, Monte Carlo)

original motivation for elfs:
quantum walks can create quantum flow state

$$|f\rangle = \frac{1}{\sqrt{R}} \sum_e f_e \, |e\rangle$$

quantum walks

=

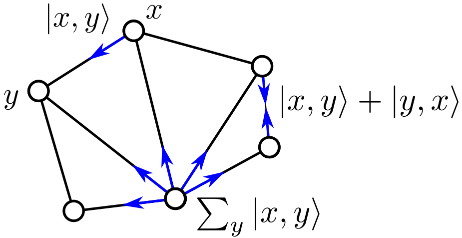
algorithmic tool in quantum computing
(element distinctness, search, Monte Carlo)

original motivation for elfs:
quantum walks can create quantum flow state

$$|f\rangle = \frac{1}{\sqrt{R}} \sum_e f_e \, |e\rangle$$

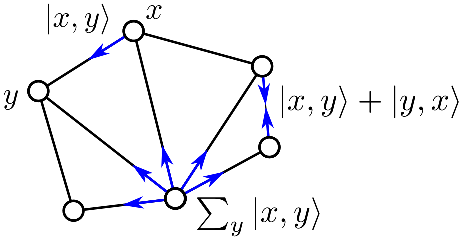
measuring $|f\rangle$ yields electric flow sample let $\mathrm{Ref}_{+} =$ reflection around $\mathrm{span}_{x,y}\{|x,y\rangle + |y,x\rangle\}$

$\mathrm{Ref}_{*} =$ reflection around $\mathrm{span}_{x}\Big\{\sum_{y\sim x} |x,y\rangle\Big\}$



let $\mathrm{Ref}_+ =$ reflection around $\mathrm{span}_{x,y}\{|x,y\rangle + |y,x\rangle\}$

$\mathrm{Ref}_* =$ reflection around $\mathrm{span}_x\left\{\sum_{y\sim x}|x,y\rangle\right\}$



quantum walk operator

$$W = \mathrm{Ref}_+\mathrm{Ref}_*$$

quantum walk operator

$$W = \mathrm{Ref}_{+}\mathrm{Ref}_{*}$$

quantum walk operator

$$W = \mathrm{Ref}_{+}\mathrm{Ref}_{*}$$

invariant subspaces:

$$\Pi_{+} \cap \Pi_{*} = \sum_{x,y} |x, y\rangle$$

$$\Pi_{-} \cap (\Pi_{*})^{\perp} = \{|h\rangle \mid h \text{ closed flow}\}$$

quantum walk operator

$$W = \mathrm{Ref}_{+}\mathrm{Ref}_{*}$$

invariant subspaces:

$$\Pi_{+} \cap \Pi_{*} = \sum_{x,y} |x, y\rangle$$

$$\Pi_{-} \cap (\Pi_{*})^{\perp} = \{|h\rangle \mid h \text{ closed flow}\}$$

! electric flow state $|f\rangle$
orthogonal to all closed flows

quantum walk algorithm

(for s-t flow)

1. start from $|s, t\rangle$

2. apply "quantum phase estimation"
to project orthogonal to invariant subspace

quantum walk algorithm

(for s-t flow)

1. start from $|s,t\rangle$

2. apply "quantum phase estimation"
to project orthogonal to invariant subspace

lemma:
can create $|f\rangle$ in $O(\sqrt{\mathrm{ET}_s})$ quantum walk steps

quantum walk algorithm

(for s-t flow)

1. start from $|s,t\rangle$

2. apply "quantum phase estimation"
to project orthogonal to invariant subspace

lemma:
can create $|f\rangle$ in $O(\sqrt{\mathrm{ET}_s})$ quantum walk steps

= quantum version of Laplacian solving

quantum walk algorithm

(for s-t flow)

1. start from $|s, t\rangle$

2. apply "quantum phase estimation"
to project orthogonal to invariant subspace

lemma:
can create $|f\rangle$ in $O(\sqrt{\mathrm{ET}_s})$ quantum walk steps

= quantum version of Laplacian solving

conjecture:
can sample from f in $O(\mathrm{ET}_s)$ random walk steps

quantum walk algorithm

(for s-t flow)

1. start from $|s, t\rangle$

2. apply "quantum phase estimation"
to project orthogonal to invariant subspace

lemma:
can create $|f\rangle$ in $O(\sqrt{\mathrm{ET}_s})$ quantum walk steps

= quantum version of Laplacian solving

conjecture:
can sample from f in $O(\mathrm{ET}_s)$ random walk steps

(ignoring errors in remainder)

quantum walk algorithm for elfs process:

quantum walk algorithm for elfs process:
repeatedly generate $|f_{Y_i,M}\rangle$ and sample

quantum walk algorithm for elfs process:
repeatedly generate $|f_{Y_i,M}\rangle$ and sample

cost $\sim \sum_{i=0}^{\rho-1} \sqrt{\mathrm{ET}_{Y_i}}$

quantum walk algorithm for elfs process:
repeatedly generate $|f_{Y_i,M}\rangle$ and sample

cost $\sim \sum_{i=0}^{\rho-1} \sqrt{\mathrm{ET}_{Y_i}}$
$(\leq \sum_{i=0}^{\rho-1} \mathrm{ET}_{Y_i} = \mathrm{HT}_s)$

quantum walk algorithm for elfs process:
repeatedly generate $|f_{Y_i,M}\rangle$ and sample

cost $\sim \sum_{i=0}^{\rho-1} \sqrt{\mathrm{ET}_{Y_i}}$
$(\leq \sum_{i=0}^{\rho-1} \mathrm{ET}_{Y_i} = \mathrm{HT}_s)$

returns element from sink M
according to random walk arrival distribution

quantum walk algorithm for elfs process:
repeatedly generate $|f_{Y_i,M}\rangle$ and sample

cost $\sim \sum_{i=0}^{\rho-1} \sqrt{\mathrm{ET}_{Y_i}}$
$(\leq \sum_{i=0}^{\rho-1} \mathrm{ET}_{Y_i} = \mathrm{HT}_s)$

returns element from sink M
according to random walk arrival distribution

many applications (to explore)
(sampling RSTs, semi-supervised learning, linear system solving)

general framing:

(quantum) algorithms for **L**x=b

general framing:

(quantum) algorithms for **L**x=b

quantum/probabilistic solution	**explicit solution**

general framing:
(quantum) algorithms for **L**x=b

quantum/probabilistic solution

quantum in

$\mathrm{poly}(\kappa)$

via HHL / quantum walks

explicit solution

general framing:

(quantum) algorithms for **L**x=b

quantum/probabilistic solution

quantum in

$\mathrm{poly}(\kappa)$

via HHL / quantum walks

classical in

$\mathrm{poly}'(\kappa)$

via random walks / Monte Carlo

explicit solution

general framing:
(quantum) algorithms for **L**x=b

quantum/probabilistic solution

quantum in
$\mathrm{poly}(\kappa)$
via HHL / quantum walks

classical in
$\mathrm{poly}'(\kappa)$
via random walks / Monte Carlo

explicit solution

quantum in
$\widetilde{O}(\sqrt{mn}/\varepsilon)$
via Grover
[Apers-de Wolf, FOCS'20]

general framing:

(quantum) algorithms for **L**x=b

quantum/probabilistic solution

quantum in

$\mathrm{poly}(\kappa)$

via HHL / quantum walks

classical in

$\mathrm{poly}'(\kappa)$

via random walks / Monte Carlo

explicit solution

quantum in

$\widetilde{O}(\sqrt{mn}/\varepsilon)$

via Grover

[Apers-de Wolf, FOCS'20]

classical in

$\widetilde{O}(m)$

[Spielman-Teng, STOC'04]

general framing:
(quantum) algorithms for **L**x=b

quantum/probabilistic solution	**explicit solution**
quantum in $\mathrm{poly}(\kappa)$ via HHL / quantum walks	quantum in $\widetilde{O}(\sqrt{mn}/\varepsilon)$ via Grover [Apers-de Wolf, FOCS'20]
classical in $\mathrm{poly}'(\kappa)$ via random walks / Monte Carlo	classical in $\widetilde{O}(m)$ [Spielman-Teng, STOC'04]

$\rightarrow$ **polynomial speedups**

general framing:
(quantum) algorithms for **A**x=b

quantum/probabilistic solution	**explicit solution**

general framing:

(quantum) algorithms for **A**x=b

quantum/probabilistic solution	**explicit solution**
quantum in $\mathrm{poly}(\kappa)$ via HHL	

general framing:
(quantum) algorithms for **A**x=b

quantum/probabilistic solution

quantum in

$\mathrm{poly}(\kappa)$

via HHL

classical?
no in general (BQP-complete)
yes for low rank (Tang '19)

explicit solution

general framing:
(quantum) algorithms for **A**x=b

quantum/probabilistic solution

quantum in
$\mathrm{poly}(\kappa)$
via HHL

classical?
no in general (BQP-complete)
yes for low rank (Tang '19)

explicit solution

quantum?

general framing:
(quantum) algorithms for **A**x=b

quantum/probabilistic solution

quantum in
$\mathrm{poly}(\kappa)$
via HHL

classical?
no in general (BQP-complete)
yes for low rank (Tang '19)

explicit solution

quantum?

classical in
n^{ω}
via matrix inversion

general framing:
(quantum) algorithms for **A**x=b

quantum/probabilistic solution

quantum in
$\mathrm{poly}(\kappa)$
via HHL

classical?
no in general (BQP-complete)
yes for low rank (Tang ’19)

$\rightarrow$ **exponential speedups?**

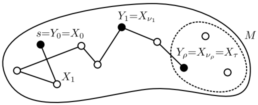
explicit solution

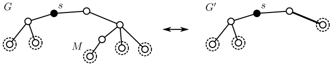
quantum?

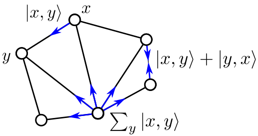
classical in
n^{ω}
via matrix inversion

$\rightarrow$ **no speedups?**

Summary

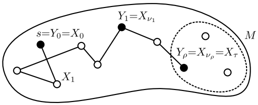

$s{=}Y_0{=}X_0$
$Y_1{=}X_{\nu_1}$
$Y_\rho{=}X_{\nu_\rho}{=}X_\tau$
X_1
M

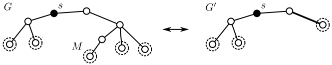

G
s
M
G'
s

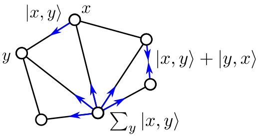

$|x, y\rangle$
x
y
$|x, y\rangle + |y, x\rangle$
$\sum_y |x, y\rangle$

Summary

- elfs process, random walk coupling

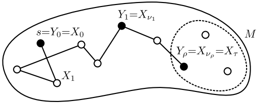


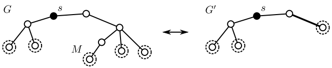


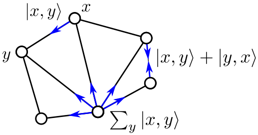


Summary

- elfs process, random walk coupling
- electric hitting time, $O(\log n)$ on trees

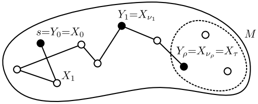


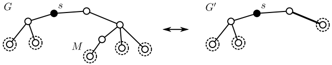


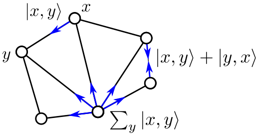


Summary

- elfs process, random walk coupling
- electric hitting time, $O(\log n)$ on trees
- quantum walk algorithms

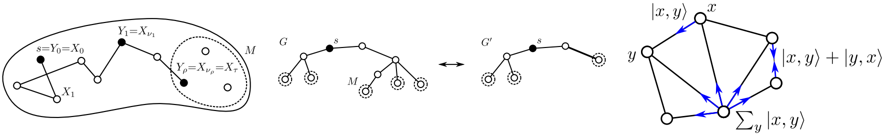






Summary

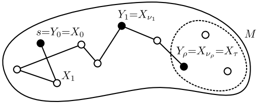
- elfs process, random walk coupling
- electric hitting time, $O(\log n)$ on trees
- quantum walk algorithms

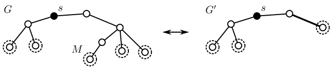


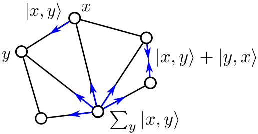
Open questions

Summary

- elfs process, random walk coupling
- electric hitting time, $O(\log n)$ on trees
- quantum walk algorithms







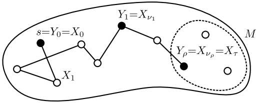
Open questions

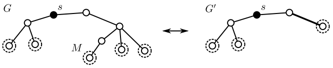
- electric flow sampling with random walks?

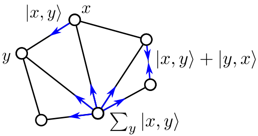
Elfs, trees and quantum walks

Summary

- elfs process, random walk coupling
- electric hitting time, $O(\log n)$ on trees
- quantum walk algorithms





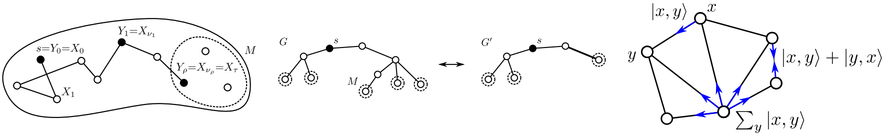


Open questions

- electric flow sampling with random walks?
- explore quantum applications

Summary

- elfs process, random walk coupling
- electric hitting time, $O(\log n)$ on trees
- quantum walk algorithms



Open questions

- electric flow sampling with random walks?
- explore quantum applications

Thank you!