Applications of the Quantum st-Connectivity Algorithm

June 3, 2019

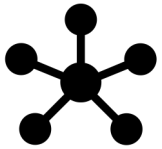
University of Maryland

Kai DeLorenzo[1], Shelby Kimmel[1], and **R. Teal Witter**[1]

[1]Middlebury College

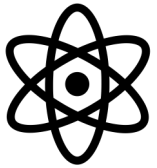
Layers of Abstraction

Our Algorithms



st-Connectivity
[Belovs, Reichardt, ’12]

Span Programs
[Reichardt, ’09, ‘11]



Quantum Gates

Recipe

Step 1: Encode a question into a graph structure.

Step 2: Analyze worst case effective resistance/capacitance.

Step 2b: Look for characteristics of original graph hidden in graph reduction.

Big Question

st-connectivity reduces quantum algorithm design to a simple classical algorithm

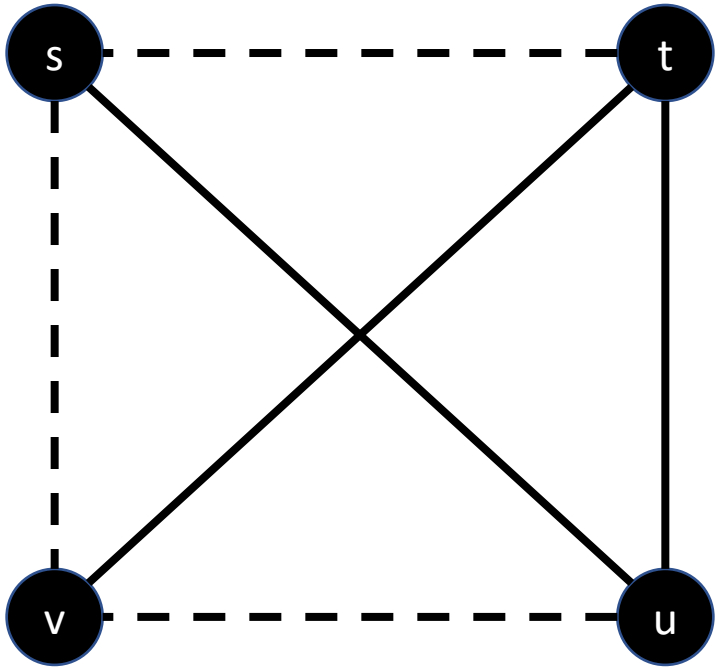
st-connectivity feels ‘natural’ in the following ways:

- it’s optimal for a wide range of problems (e.g. Boolean formula evaluation, total connectivity) and
- it’s easy to analyze using effective resistance/capacitance

Does the st-connectivity approach give intuition for optimal underlying quantum algorithms?

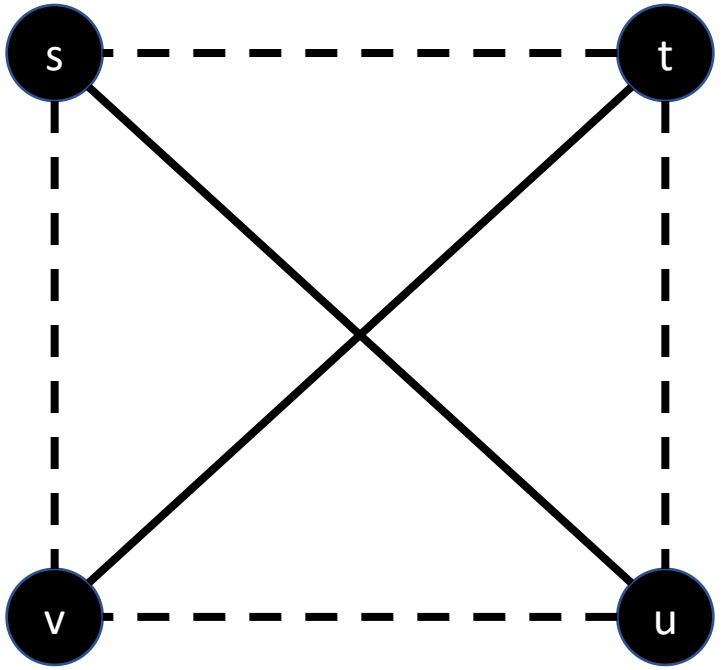
st-Connectivity

Is there a path between s and t?



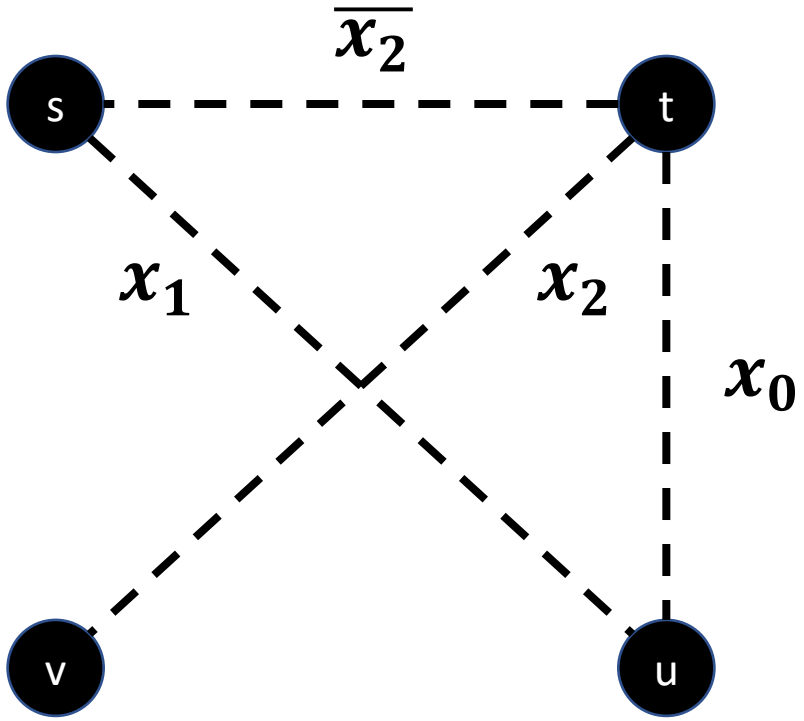
st-Connectivity

Is there a path between s and t?



st-Connectivity Query Model

Input:
- graph skeleton
- hidden bit string

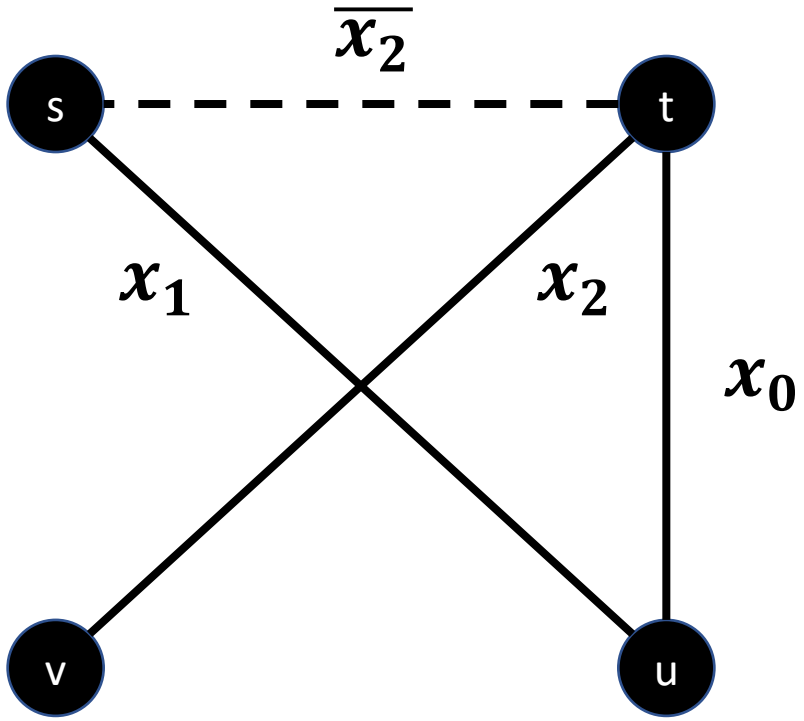


x_0 x_1 x_2

st-Connectivity Query Model

Input:
- graph skeleton
- hidden bit string

Output:
connected



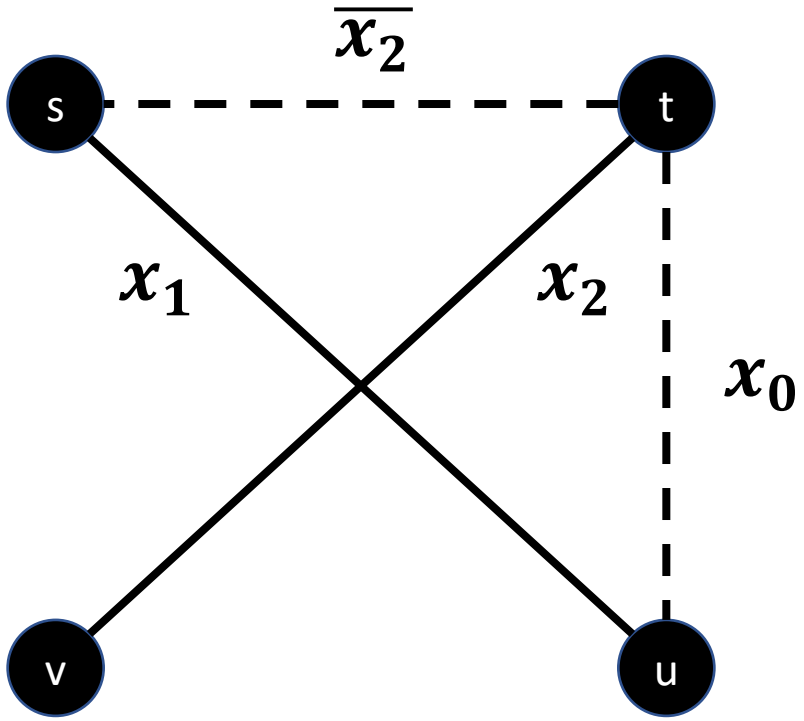
x_0	x_1	x_2
1	1	1

st-Connectivity Query Model

Input:
- graph skeleton
- hidden bit string

Output:
not connected

Goal:
minimize queries
to bits



x_0	x_1	x_2
0	1	1

st-Connectivity Complexity

Space: $O(\log(\#edges\ in\ skeleton))$

[Belovs, Reichardt, ’12] [Jeffery, Kimmel, ’17]

Effective Resistance

[Belovs, Reichardt, ’12]

Query: $O\left(\sqrt{\max_{connected\ G} R_{s,t}(G)} \sqrt{\max_{not\ connected\ G} C_{s,t}(G)}\right)$

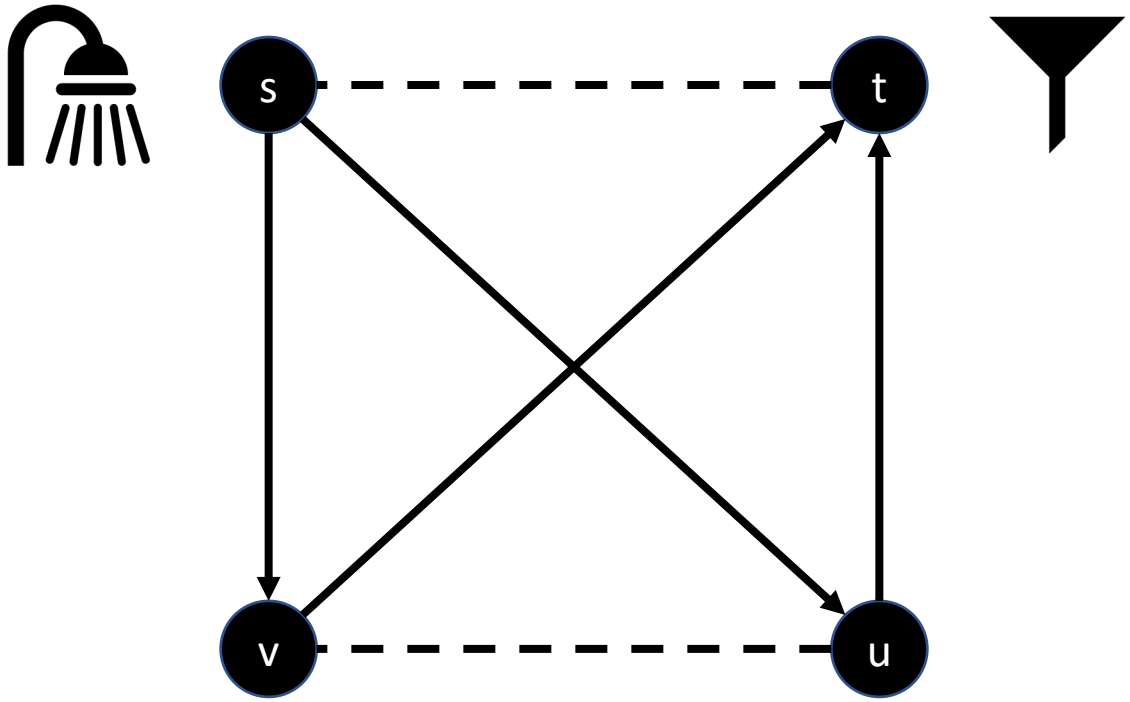
Effective Capacitance

[Jarret, Jeffery, Kimmel, Piedrafita, ’18]

Time: query *times* time for quantum walk on skeleton

[Belovs, Reichardt, ’12] [Jeffery, Kimmel, ’17]

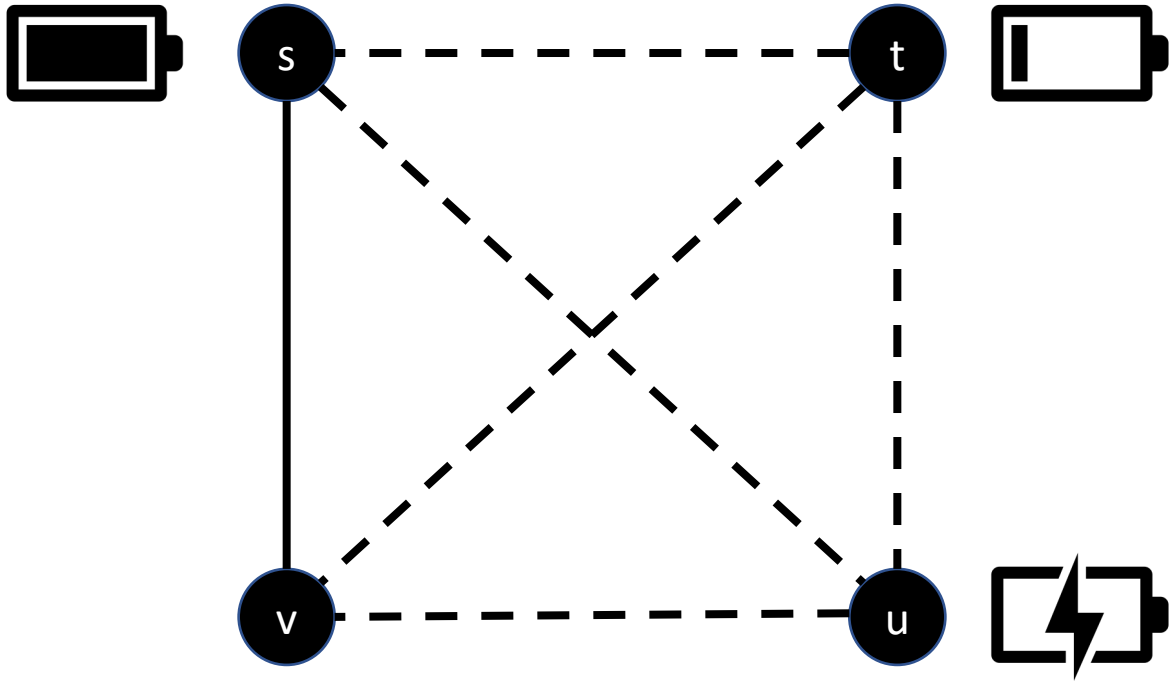
st-Connectivity Resistance



Bounded by longest path

$$R_{s,t}(G) = \min_{flows} \sum_{edges} (flow\ on\ edge)^2$$

st-Connectivity Capacitance



Bounded by biggest cut

$$C_{s,t}(G) = \min_{potentials} \sum_{\substack{edges \\ in\ skeleton}} (potential\ difference)^2$$

Cycle Detection

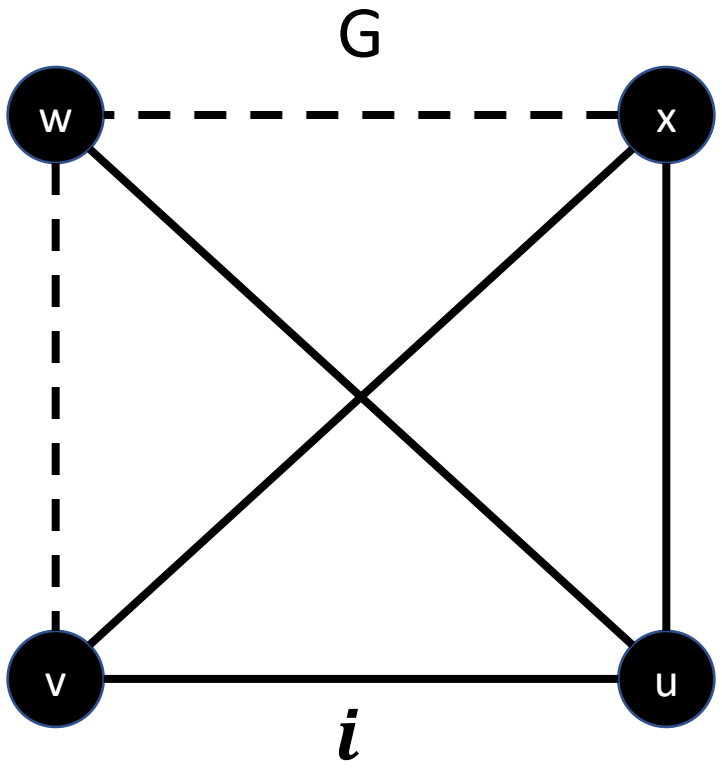
Is edge $\boldsymbol{i}$ (between u and v) on a cycle?

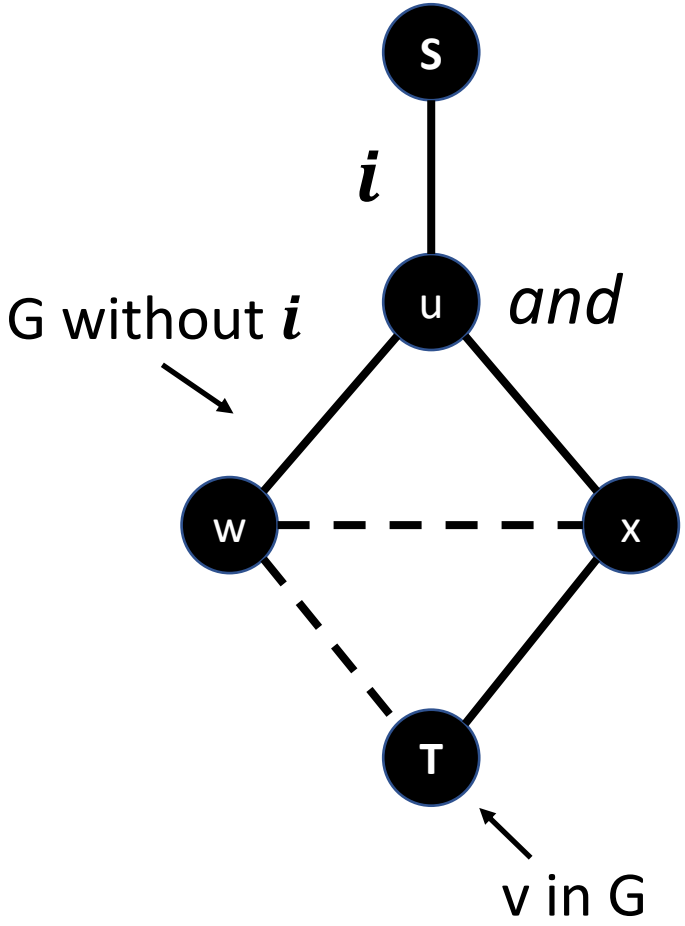
$\Leftrightarrow$

Does edge $\boldsymbol{i}$ exist?

and

Is there a path from u to v without $\boldsymbol{i}$?



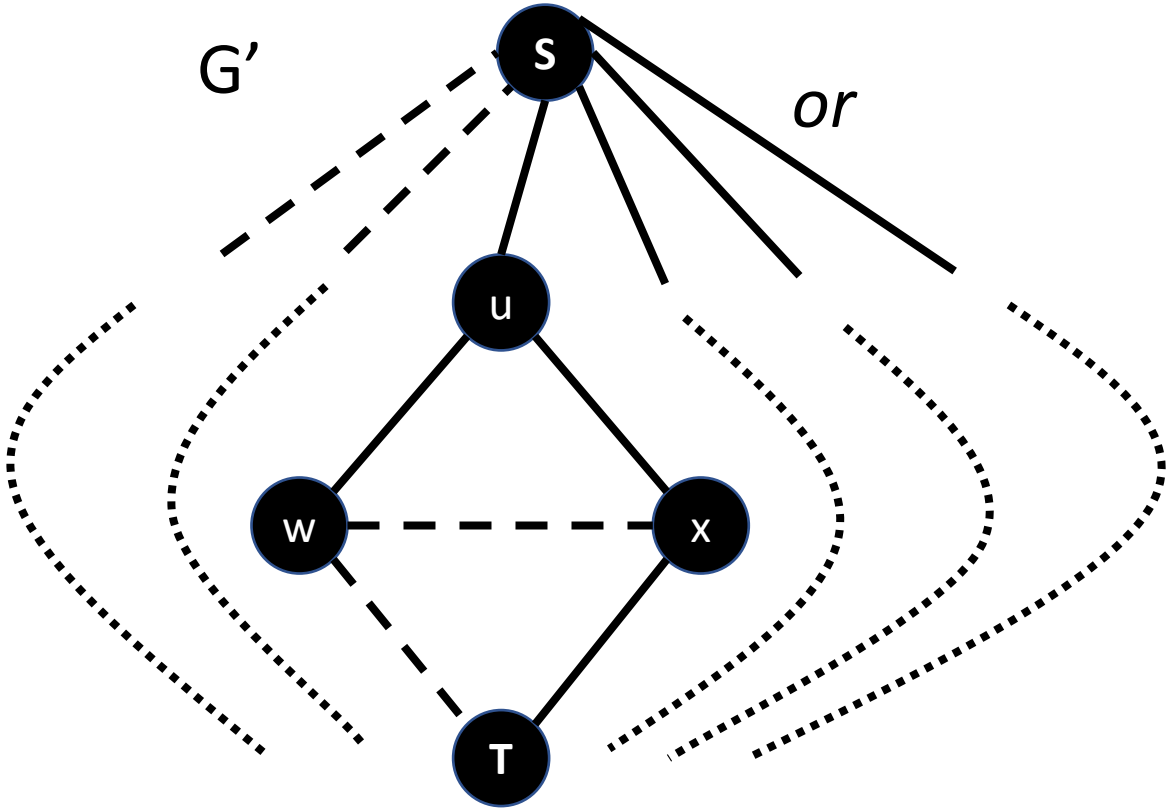


Cycle Detection

Is there a cycle in G?

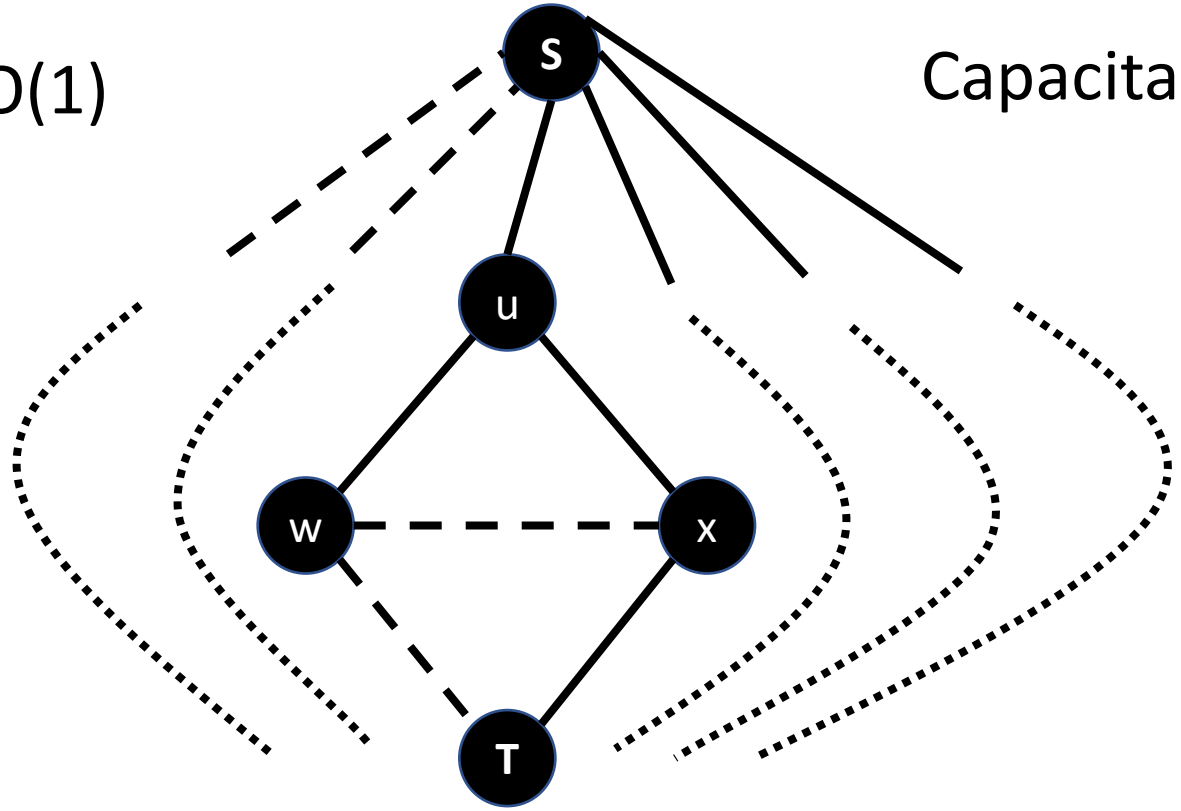
$\Leftrightarrow$

For some edge $\boldsymbol{i}$:
Does $\boldsymbol{i}$ exist?
and
Is there a path from u to v without $\boldsymbol{i}$?



Cycle Detection Complexity

Resistance: O(1)



vertices

Capacitance: O(n m)

edges in skeleton

Complexity: $O(n^{3/2})$

Cycle Detection Bounds

Lower Bound: $\Omega(n^{3/2})$

[Childs, Kothari, ’12]

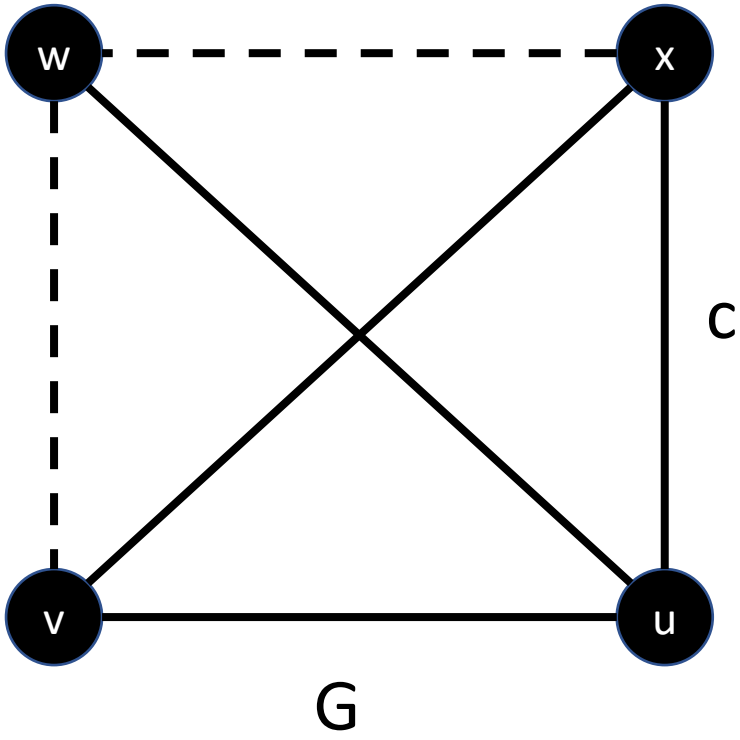
Previous Bound: $\tilde{O}(n^{3/2})$

[Cade, Montenaro, Belovs, ’18]

Our Bound: $O(n^{3/2})$

Circuit Rank Estimation

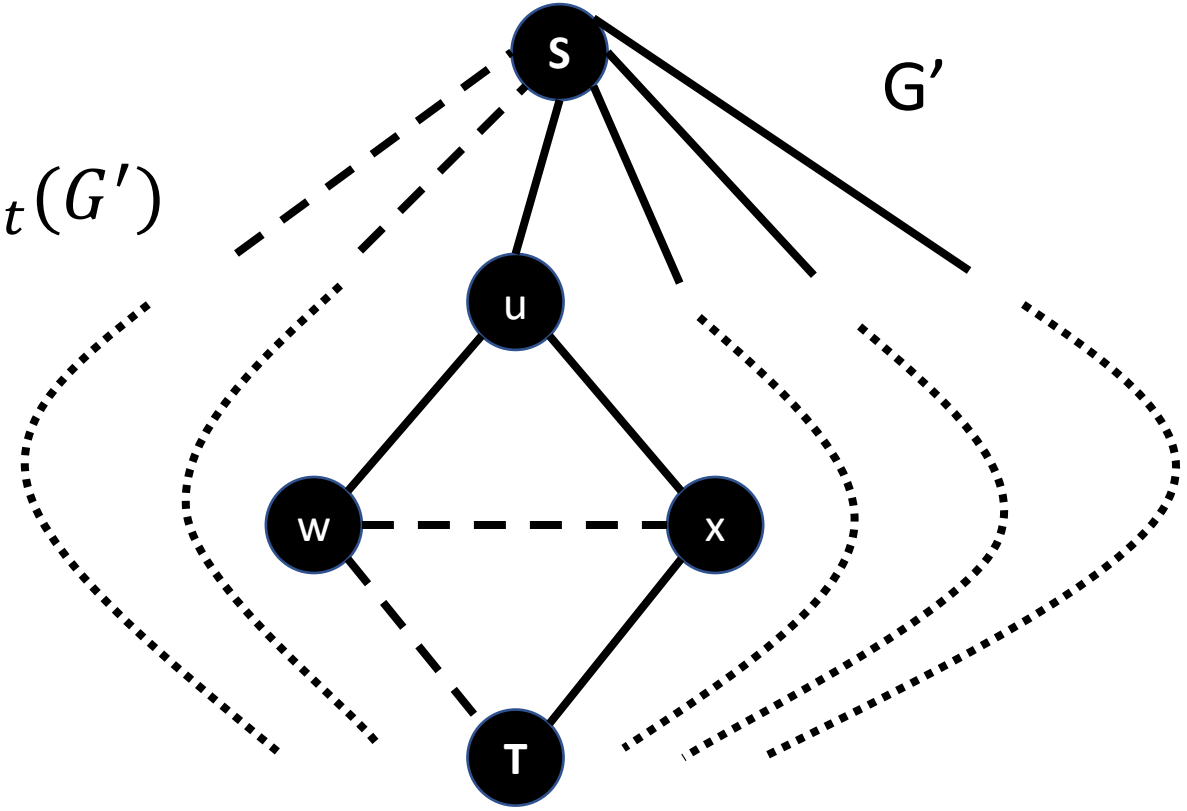
edges to cut until no cycles



G

$r = 1/R_{s,t}(G')$

circuit rank



G’

Circuit Rank Estimation Bounds

Lower/Previous Bound: ?

$\sqrt{n^3/r}$ if cactus graph

Our Bound: $\tilde{O}(\epsilon^{3/2}\sqrt{n^4/r})$

multiplicative error

Even Length Cycle Detection

connected cycles

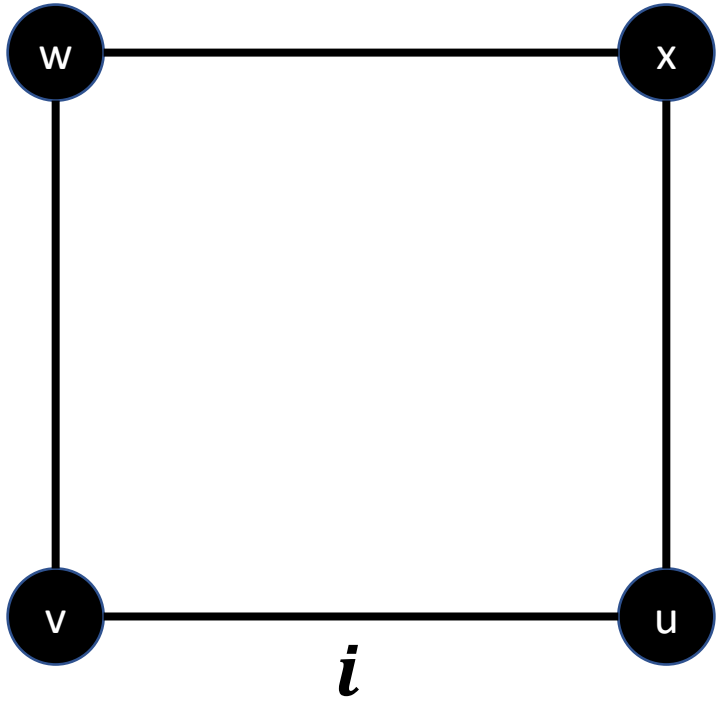
Is edge $\boldsymbol{i}$ on an even length cycle?

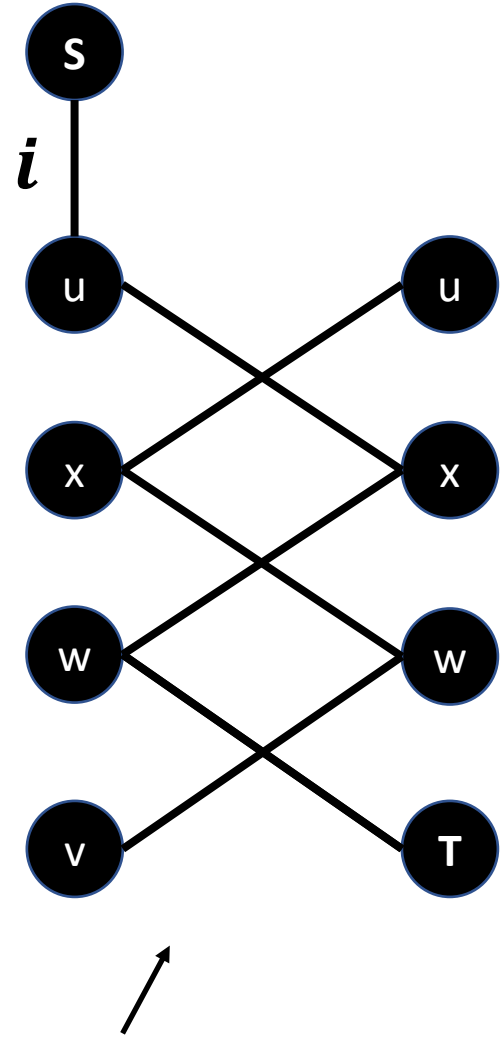
$\Leftrightarrow$

Does edge $\boldsymbol{i}$ exist?

and

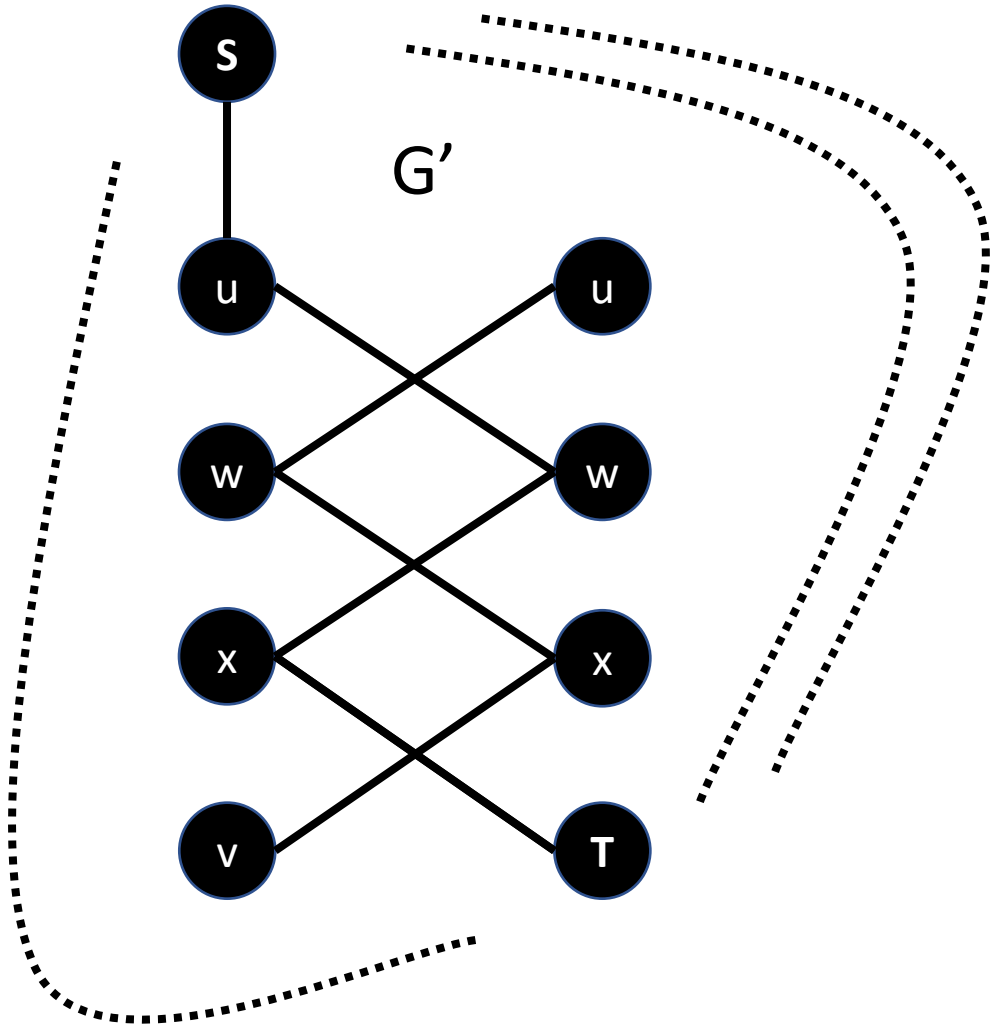
Is there an odd length path from u to v not including $\boldsymbol{i}$?





bipartite double

Even Length Cycle Detection Complexity



Complexity: $O(n^{3/2})$

Even Length Cycle Detection Bounds

Lower/Previous Bound: ?

Classical Bound: $\Theta(n^2)$
[Yuster, Zwick, ’97]

Our Bound: $O(n^{3/2})$

Open Problems

Full analysis of (highly structured) reductions to show time efficiency

Extend cycle length detection to arbitrary modulus

Determine if reducing other Symmetric Logarithm problems to st-connectivity always gives optimal algorithm

Does the st-connectivity approach give intuition for the optimal underlying quantum algorithms?

Thank you!

Kai DeLorenzo

Shelby Kimmel