Efficient Construction of Flows for the One-Way Measurement Model

Niel de Beaudrap

Institute for Quantum Computing
University of Waterloo

CQISC 2006, Calgary

Outline

1. Introduction
   - The One-Way Measurement Model
   - Flows in one-way patterns
2. Phase Map Decompositions
3. Graph-theoretic results
   - Path covers
   - Influencing walks, Vicious circuits
   - Uniqueness when $|I| = |O|$
4. Efficient algorithms when $|I| = |O|$

Outline

1. Introduction
   - The One-Way Measurement Model
   - Flows in one-way patterns
2. Phase Map Decompositions
3. Graph-theoretic results
   - Path covers
   - Influencing walks, Vicious circuits
   - Uniqueness when $|I| = |O|$
4. Efficient algorithms when $|I| = |O|$

Outline

1. Introduction
   - The One-Way Measurement Model
   - Flows in one-way patterns
2. Phase Map Decompositions
3. Graph-theoretic results
   - Path covers
   - Influencing walks, Vicious circuits
   - Uniqueness when $|I| = |O|$
4. Efficient algorithms when $|I| = |O|$

Outline

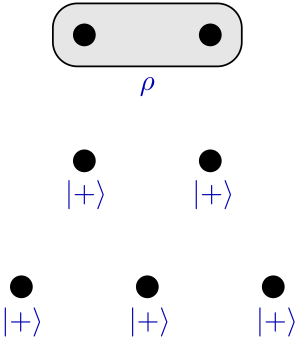
1. Introduction
   - The One-Way Measurement Model
   - Flows in one-way patterns
2. Phase Map Decompositions
3. Graph-theoretic results
   - Path covers
   - Influencing walks, Vicious circuits
   - Uniqueness when $|I| = |O|$
4. Efficient algorithms when $|I| = |O|$

Outline

1. Introduction
   - The One-Way Measurement Model
   - Flows in one-way patterns
2. Phase Map Decompositions
3. Graph-theoretic results
   - Path covers
   - Influencing walks, Vicious circuits
   - Uniqueness when $|I| = |O|$
4. Efficient algorithms when $|I| = |O|$

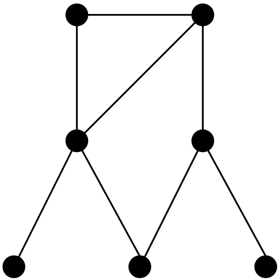
The One-Way Measurement Model
(a sketch)

1. Prepare a state ρ to be transformed in an input system I, and a collection of ancillas in the $|+\rangle$ state;
2. act on the ancillas and the qubits of I using controlled-Z operators to form an *entanglement graph* G ;
3. perform a sequence of single-qubit measurements on the qubits of G, leaving only an output system O ;
4. perform Pauli corrections on O, yielding the output state.



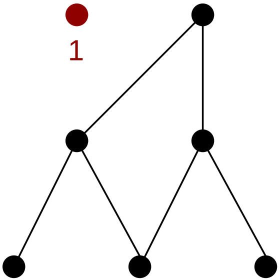
The One-Way Measurement Model
(a sketch)

1. Prepare a state ρ to be transformed in an input system I, and a collection of ancillas in the $|+\rangle$ state;
2. act on the ancillas and the qubits of I using controlled-Z operators to form an *entanglement graph* G ;
3. perform a sequence of single-qubit measurements on the qubits of G, leaving only an output system O ;
4. perform Pauli corrections on O, yielding the output state.



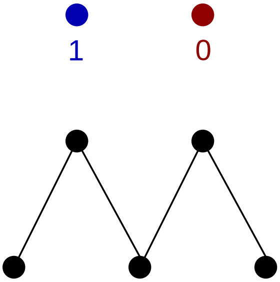
The One-Way Measurement Model
(a sketch)

1. Prepare a state ρ to be transformed in an input system I, and a collection of ancillas in the $|+\rangle$ state;
2. act on the ancillas and the qubits of I using controlled-Z operators to form an *entanglement graph* G ;
3. perform a sequence of single-qubit measurements on the qubits of G, leaving only an output system O ;
4. perform Pauli corrections on O, yielding the output state.



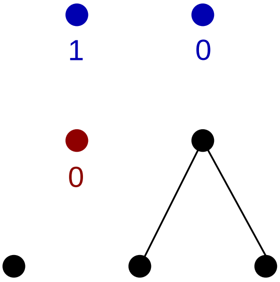
The One-Way Measurement Model
(a sketch)

1. Prepare a state ρ to be transformed in an input system I, and a collection of ancillas in the $|+\rangle$ state;
2. act on the ancillas and the qubits of I using controlled-Z operators to form an *entanglement graph* G ;
3. perform a sequence of single-qubit measurements on the qubits of G, leaving only an output system O ;
4. perform Pauli corrections on O, yielding the output state.



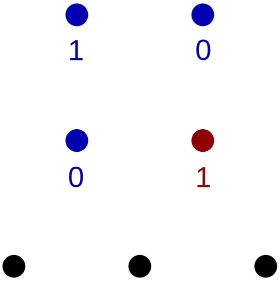
The One-Way Measurement Model
(a sketch)

1. Prepare a state ρ to be transformed in an input system I, and a collection of ancillas in the $|+\rangle$ state;
2. act on the ancillas and the qubits of I using controlled-Z operators to form an *entanglement graph* G ;
3. perform a sequence of single-qubit measurements on the qubits of G, leaving only an output system O ;
4. perform Pauli corrections on O, yielding the output state.



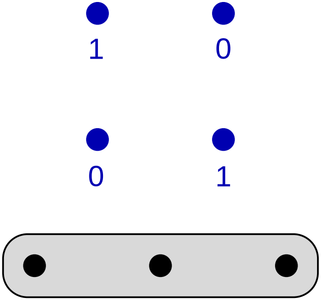
The One-Way Measurement Model
(a sketch)

1. Prepare a state ρ to be transformed in an input system I, and a collection of ancillas in the $|+\rangle$ state;
2. act on the ancillas and the qubits of I using controlled-Z operators to form an *entanglement graph* G ;
3. perform a sequence of single-qubit measurements on the qubits of G, leaving only an output system O ;
4. perform Pauli corrections on O, yielding the output state.



The One-Way Measurement Model
(a sketch)

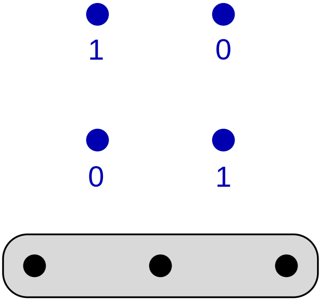
1. Prepare a state ρ to be transformed in an input system I, and a collection of ancillas in the $|+\rangle$ state;
2. act on the ancillas and the qubits of I using controlled-Z operators to form an *entanglement graph* G ;
3. perform a sequence of single-qubit measurements on the qubits of G, leaving only an output system O ;
4. perform Pauli corrections on O, yielding the output state.



$\mathcal{A}_{1001}(\rho)$

The One-Way Measurement Model
(a sketch)

1. Prepare a state ρ to be transformed in an input system I, and a collection of ancillas in the $|+\rangle$ state;
2. act on the ancillas and the qubits of I using controlled-Z operators to form an *entanglement graph* G ;
3. perform a sequence of single-qubit measurements on the qubits of G, leaving only an output system O ;
4. perform Pauli corrections on O, yielding the output state.



$\mathcal{A}_{1001}(\rho)$

$\longmapsto \mathcal{A}(\rho)$

The One-Way Measurement Model

- Each qubit $x \notin O$ is measured with some operator on the equator of the Bloch sphere, determined by a measurement angle α_x

$$M_\alpha = |+_\alpha\rangle\langle +_\alpha| - |-_\alpha\rangle\langle -_\alpha|$$

$$|\pm_\alpha\rangle = \frac{1}{\sqrt{2}}\Big(|0\rangle \pm e^{i\alpha} |1\rangle \Big)$$

$$|+_\alpha\rangle \longmapsto 0$$
$$|-_\alpha\rangle \longmapsto 1$$

- Measurements may (and almost always do) depend on the results $s_x \in \{0, 1\}$ of previous measurements

$$\textit{e.g.} \qquad \alpha_z \longmapsto (-1)^{s_x + s_y} \alpha_z$$

- Dependencies arise from how information is propagated through the entanglement graph
  $\Rightarrow$ the graph restricts the order of measurements

The One-Way Measurement Model

- Each qubit $x \notin O$ is measured with some operator on the equator of the Bloch sphere, determined by a measurement angle α_x

$$M_\alpha = |{+_\alpha}\rangle\langle{+_\alpha}| - |{-_\alpha}\rangle\langle{-_\alpha}|$$

$$|{\pm_\alpha}\rangle = \tfrac{1}{\sqrt{2}}\Big(|0\rangle \pm e^{i\alpha}|1\rangle\Big)$$

$$|{+_\alpha}\rangle \longmapsto 0$$

$$|{-_\alpha}\rangle \longmapsto 1$$

- Measurements may (and almost always do) depend on the results $s_x \in \{0,1\}$ of previous measurements

$$e.g. \qquad \alpha_z \longmapsto (-1)^{s_x + s_y}\,\alpha_z$$

- Dependencies arise from how information is propagated through the entanglement graph
$\Rightarrow$ the graph restricts the order of measurements

The One-Way Measurement Model

- Each qubit $x \notin O$ is measured with some operator on the equator of the Bloch sphere, determined by a measurement angle α_x

$$M_\alpha = |+_\alpha\rangle\langle+_\alpha| - |-_\alpha\rangle\langle-_\alpha|$$

$$|\pm_\alpha\rangle = \frac{1}{\sqrt{2}}\Big(|0\rangle \pm e^{i\alpha}|1\rangle\Big)$$

$$|+_\alpha\rangle \longmapsto 0$$
$$|-_\alpha\rangle \longmapsto 1$$

- Measurements may (and almost always do) depend on the results $s_x \in \{0, 1\}$ of previous measurements

$$\textit{e.g.} \qquad \alpha_z \longmapsto (-1)^{s_x + s_y}\,\alpha_z$$

- Dependencies arise from how information is propagated through the entanglement graph
  $\Rightarrow$ the graph restricts the order of measurements

About flows

[Danos, Kashefi 2005]

Flows describe how information is "transmitted" in an entanglement graph $G = (V, E)$, from the inputs $I \subseteq V$ to the outputs $O \subseteq V$.

Definition

A **flow** on (G, I, O) is an ordered pair $(f, \preccurlyeq)$

- $f : O^{\mathsf{c}} \longrightarrow I^{\mathsf{c}}$ is a function on vertices
- $\preccurlyeq$ is a partial order on V

  (*i.e.* a reflexive, transitive, & antisymmetric relation)

which satisfy the following three conditions for all vertices:

$$\begin{aligned} &(\mathrm{F}i) && x \sim f(x)\,; \\ &(\mathrm{F}ii) && x \preccurlyeq f(x)\,; \\ &(\mathrm{F}iii) && y \sim f(x) \quad \Rightarrow \quad x \preccurlyeq y\,. \end{aligned}$$

About flows

[Danos, Kashefi 2005]

Flows describe how information is "transmitted" in an entanglement graph $G = (V, E)$, from the inputs $I \subseteq V$ to the outputs $O \subseteq V$.

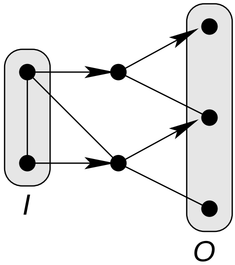
Definition

A **flow** on (G, I, O) is an ordered pair $(f, \preccurlyeq)$

- $f : O^{\mathrm{c}} \longrightarrow I^{\mathrm{c}}$ is a function on vertices
- $\preccurlyeq$ is a partial order on V
  (*i.e.* a reflexive, transitive, & antisymmetric relation)

which satisfy the following three conditions for all vertices:

$$\begin{aligned} &(\mathrm{F}i) && x \sim f(x)\,; \\ &(\mathrm{F}ii) && x \preccurlyeq f(x)\,; \\ &(\mathrm{F}iii) && y \sim f(x) \quad \Rightarrow \quad x \preccurlyeq y\,. \end{aligned}$$



About flows

[Danos, Kashefi 2005]

Flows describe how information is “transmitted” in an entanglement graph $G = (V, E)$, from the inputs $I \subseteq V$ to the outputs $O \subseteq V$.

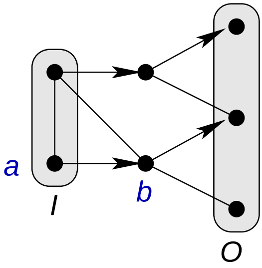
Definition

A **flow** on (G, I, O) is an ordered pair $(f, \preccurlyeq)$

- $f : O^c \longrightarrow I^c$ is a function on vertices
- $\preccurlyeq$ is a partial order on V
  (*i.e.* a reflexive, transitive, & antisymmetric relation)

which satisfy the following three conditions for all vertices:

$$\begin{aligned} &(\mathrm{F}i) & x \sim f(x)\,; \\ &(\mathrm{F}ii) & x \preccurlyeq f(x)\,; \\ &(\mathrm{F}iii) & y \sim f(x) \quad \Rightarrow \quad x \preccurlyeq y\,. \end{aligned}$$



$a \preccurlyeq b$

About flows

[Danos, Kashefi 2005]

Flows describe how information is “transmitted” in an entanglement graph $G = (V, E)$, from the inputs $I \subseteq V$ to the outputs $O \subseteq V$.

Definition

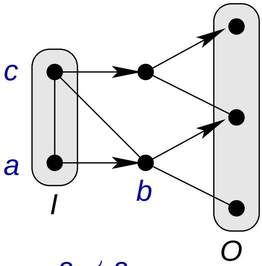
A **flow** on (G, I, O) is an ordered pair $(f, \preccurlyeq)$

- $f : O^{\mathrm{c}} \longrightarrow I^{\mathrm{c}}$ is a function on vertices
- $\preccurlyeq$ is a partial order on V

  (*i.e.* a reflexive, transitive, & antisymmetric relation)

which satisfy the following three conditions for all vertices:

$$\begin{array}{rl} \text{(F}i\text{)} & x \sim f(x)\,; \\ \text{(F}ii\text{)} & x \preccurlyeq f(x)\,; \\ \text{(F}iii\text{)} & y \sim f(x) \quad \Rightarrow \quad x \preccurlyeq y\,. \end{array}$$



$a \preccurlyeq a$

$a \preccurlyeq b$

$a \preccurlyeq c$

About flows

[Danos, Kashefi 2005]

Flows describe how information is "transmitted" in an entanglement graph $G = (V, E)$, from the inputs $I \subseteq V$ to the outputs $O \subseteq V$.

Definition

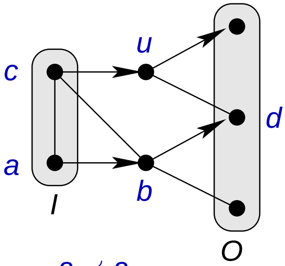
A **flow** on (G, I, O) is an ordered pair $(f, \preccurlyeq)$

- $f : O^{\mathrm{c}} \longrightarrow I^{\mathrm{c}}$ is a function on vertices
- $\preccurlyeq$ is a partial order on V

  (*i.e.* a reflexive, transitive, & antisymmetric relation)

which satisfy the following three conditions for all vertices:

$$\begin{aligned} &(\mathrm{F}i) && x \sim f(x)\,; \\ &(\mathrm{F}ii) && x \preccurlyeq f(x)\,; \\ &(\mathrm{F}iii) && y \sim f(x) \quad \Rightarrow \quad x \preccurlyeq y\,. \end{aligned}$$



$a \preccurlyeq a$

$a \preccurlyeq b \preccurlyeq d$

$a \preccurlyeq c \preccurlyeq u$

etc.

About flows

[Danos, Kashefi 2005]

The existence of a flow is an entirely graph-theoretic property, but guarantees properties important for quantum computation:

- For any choice of measurement angles $\{\alpha_v\}_{v \in O^c}$, there is a one-way pattern with a measurement order consistent with $\preccurlyeq$, which performs a unitary injection;
- In that pattern, the result of each measurement is uniformly random.
- Also: every unitary operator can be implemented by a one-way pattern whose measurement order is described by a flow.

This makes them a potentially useful tool for analysing:

About flows [Danos, Kashefi 2005]

The existence of a flow is an entirely graph-theoretic property, but guarantees properties important for quantum computation:

- For any choice of measurement angles $\{\alpha_v\}_{v \in O^c}$, there is a one-way pattern with a measurement order consistent with $\preccurlyeq$, which performs a unitary injection;
- In that pattern, the result of each measurement is uniformly random.
- Also: every unitary operator can be implemented by a one-way pattern whose measurement order is described by a flow.

This makes them a potentially useful tool for analysing:

About flows

[Danos, Kashefi 2005]

The existence of a flow is an entirely graph-theoretic property, but guarantees properties important for quantum computation:

- For any choice of measurement angles $\{\alpha_v\}_{v \in O^c}$, there is a one-way pattern with a measurement order consistent with $\preccurlyeq$, which performs a unitary injection;
- In that pattern, the result of each measurement is uniformly random.
- Also: every unitary operator can be implemented by a one-way pattern whose measurement order is described by a flow.

This makes them a potentially useful tool for analysing:

Outline

1. Introduction
   - The One-Way Measurement Model
   - Flows in one-way patterns
2. Phase Map Decompositions
3. Graph-theoretic results
   - Path covers
   - Influencing walks, Vicious circuits
   - Uniqueness when $|I| = |O|$
4. Efficient algorithms when $|I| = |O|$

Phase Map Decomposition

[B, Danos, Kashefi 2006]

Consider a one-way pattern implementing a unitary operator U :

- entanglement graph $G = (V, E)$, input/output vertices $I, O \subseteq V$, such that (G, I, O) has a flow
- measurement angles $\{\alpha_v\}_{v \in O^c}$

Considering U, between standard bases of $\mathcal{H}_I$ and $\mathcal{H}_O$:

$$U_{I \to O} \quad = \quad R_{V \to O} \circ \Phi_G \circ P_{I \to V} \quad ,$$

- P : a *preparation map*, setting all qubits in I^c in the state $|+\rangle$
- R : a *restriction map*, post-selecting $|+\rangle$ for all qubits in O^c
- Φ_G : a diagonal unitary operator (or *phase map*) with the structure

$$\Phi_G \quad = \quad \left[\bigotimes_{v \in O^c} e^{-i\alpha_v |1\rangle\langle 1|_v} \right] \left[\prod_{uv \in E} \wedge Z_{u,v} \right]$$

Phase Map Decomposition

[B, Danos, Kashefi 2006]

Consider a one-way pattern implementing a unitary operator U :

- entanglement graph $G = (V, E)$, input/output vertices $I, O \subseteq V$, such that (G, I, O) has a flow
- measurement angles $\{\alpha_v\}_{v \in O^c}$

Considering U, between standard bases of $\mathcal{H}_I$ and $\mathcal{H}_O$:

$$U_{I \to O} \quad = \quad R_{V \to O} \circ \Phi_G \circ P_{I \to V} \quad ,$$

- P : a *preparation map*, setting all qubits in I^c in the state $|+\rangle$
- R : a *restriction map*, post-selecting $|+\rangle$ for all qubits in O^c
- Φ_G : a diagonal unitary operator (or *phase map*) with the structure

$$\Phi_G \quad = \left[\bigotimes_{v \in O^c} e^{-i\alpha_v |1\rangle\langle 1|_v} \right] \left[\prod_{uv \in E} \wedge Z_{u,v} \right]$$

Phase Map Decomposition

[B, Danos, Kashefi 2006]

Consider a one-way pattern implementing a unitary operator U :

- entanglement graph $G = (V, E)$, input/output vertices $I, O \subseteq V$, such that (G, I, O) has a flow
- measurement angles $\{\alpha_v\}_{v \in O^c}$

Considering U, between standard bases of $\mathcal{H}_I$ and $\mathcal{H}_O$:

$$U_{I \to O} \quad = \quad R_{V \to O} \circ \Phi_G \circ P_{I \to V} \quad ,$$

- P : a *preparation map*, setting all qubits in I^c in the state $|+\rangle$
- R : a *restriction map*, post-selecting $|+\rangle$ for all qubits in O^c
- Φ_G : a diagonal unitary operator (or *phase map*) with the structure

$$\Phi_G \quad = \quad \left[\bigotimes_{v \in O^c} \mathrm{e}^{-i\alpha_v |1\rangle\langle 1|_v} \right] \left[\prod_{uv \in E} \wedge Z_{u,v} \right]$$

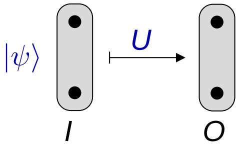
Phase Map Decomposition [B, Danos, Kashefi 2006]

$$U_{I \to O} \quad = \quad R_{V \to O} \circ \Phi_G \circ P_{I \to V} \quad ,$$

- P : a *preparation map*, setting all qubits in I^c in the state $|+\rangle$
- R : a *restriction map*, post-selecting $|+\rangle$ for all qubits in O^c
- Φ_G : a diagonal unitary operator (or *phase map*) with the structure

$$\Phi_G \quad = \quad \left[\bigotimes_{v \in O^c} \mathrm{e}^{-i\alpha_v |1\rangle\langle 1|_v} \right] \left[\prod_{uv \in E} \wedge Z_{u,v} \right]$$

Phase Map Decomposition [B, Danos, Kashefi 2006]



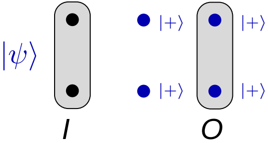
$$|\psi\rangle_I \quad \longmapsto \quad U\,|\psi\rangle_O$$

$$U_{I\to O} \quad = \quad R_{V\to O} \circ \Phi_G \circ P_{I\to V} \quad ,$$

- P : a *preparation map*, setting all qubits in I^c in the state $|+\rangle$
- R : a *restriction map*, post-selecting $|+\rangle$ for all qubits in O^c
- Φ_G : a diagonal unitary operator (or *phase map*) with the structure

$$\Phi_G \quad = \quad \left[\bigotimes_{v\in O^c} \mathrm{e}^{-i\alpha_v |1\rangle\langle 1|_v}\right] \left[\prod_{uv\in E} \wedge Z_{u,v}\right]$$

Phase Map Decomposition [B, Danos, Kashefi 2006]



$$|\psi\rangle_I \otimes |+\rangle\,|+\rangle\,|+\rangle\,|+\rangle\,|+\rangle\,|+\rangle$$

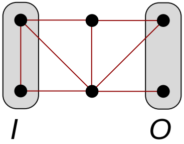
$$U_{I\to O} \;=\; R_{V\to O} \circ \Phi_G \circ P_{I\to V} \;,$$

- P : a *preparation map*, setting all qubits in I^{c} in the state $|+\rangle$
- R : a *restriction map*, post-selecting $|+\rangle$ for all qubits in O^{c}
- Φ_G : a diagonal unitary operator (or *phase map*) with the structure

$$\Phi_G \;=\; \left[\bigotimes_{v\in O^{\mathrm{c}}} \mathrm{e}^{-i\alpha_v |1\rangle\langle 1|_v}\right]\left[\prod_{uv\in E} \wedge Z_{u,v}\right]$$

Phase Map Decomposition

[B, Danos, Kashefi 2006]



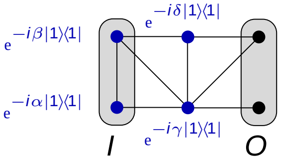
$$
\begin{aligned}
\longmapsto\quad & |+_\alpha\rangle\,|+_\beta\rangle\,\;|+_\gamma\rangle\;\;\;|+_\delta\rangle\;\otimes U\,|\psi\rangle_O \\
+\quad & |-_\alpha\rangle\,|+_\beta\rangle\,|+_{-\gamma}\rangle\,|+_{-\delta}\rangle\otimes P_{1000}U\,|\psi\rangle_O \\
+\quad & |+_\alpha\rangle\,|-_\beta\rangle\;\;|+_\gamma\rangle\;\;|+_{-\delta}\rangle\otimes P_{0100}U\,|\psi\rangle_O \\
& \qquad\qquad\vdots \\
+\quad & |-_\alpha\rangle\,|-_\beta\rangle\,|-_{-\gamma}\rangle\;\;|-_\delta\rangle\;\otimes P_{1111}U\,|\psi\rangle_O
\end{aligned}
$$

$$U_{I\to O} \;=\; R_{V\to O} \circ \Phi_G \circ P_{I\to V}\;,$$

- P : a *preparation map*, setting all qubits in I^{c} in the state $|+\rangle$
- R : a *restriction map*, post-selecting $|+\rangle$ for all qubits in O^{c}
- Φ_G : a diagonal unitary operator (or *phase map*) with the structure

$$\Phi_G \;=\; \left[\bigotimes_{v\in O^{\mathrm{c}}} \mathrm{e}^{-i\alpha_v\,|1\rangle\langle 1|_v}\right]\left[\prod_{uv\in E} \wedge Z_{u,v}\right]$$

Phase Map Decomposition [B, Danos, Kashefi 2006]



$$
\begin{array}{rl}
\longmapsto & |+\rangle\,|+\rangle\,|+\rangle\,|+\rangle \otimes U\,|\psi\rangle_O \\
+ & |-\rangle\,|+\rangle\,|+_{-2\gamma}\rangle\,|+_{-2\delta}\rangle \otimes P_{1000} U\,|\psi\rangle_O \\
+ & |+\rangle\,|-\rangle\,|+\rangle\,|+_{-2\delta}\rangle \otimes P_{0100} U\,|\psi\rangle_O \\
 & \vdots \\
+ & |-\rangle\,|-\rangle\,|-_{-2\gamma}\rangle\,|-\rangle \otimes P_{1111} U\,|\psi\rangle_O
\end{array}
$$

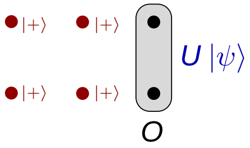
$$U_{I\to O} = R_{V\to O} \circ \Phi_G \circ P_{I\to V} \,,$$

- P : a *preparation map*, setting all qubits in I^c in the state $|+\rangle$
- R : a *restriction map*, post-selecting $|+\rangle$ for all qubits in O^c
- Φ_G : a diagonal unitary operator (or *phase map*) with the structure

$$\Phi_G = \left[\bigotimes_{v\in O^c} e^{-i\alpha_v |1\rangle\langle 1|_v} \right] \left[\prod_{uv\in E} \wedge Z_{u,v} \right]$$

Phase Map Decomposition [B, Danos, Kashefi 2006]

$$\longmapsto \quad |+\rangle \; |+\rangle \; |+\rangle \; |+\rangle \; \otimes U\,|\psi\rangle_O$$



$$U_{I\to O} \;=\; R_{V\to O} \circ \Phi_G \circ P_{I\to V} \;,$$

- P : a *preparation map*, setting all qubits in I^{c} in the state $|+\rangle$
- R : a *restriction map*, post-selecting $|+\rangle$ for all qubits in O^{c}
- Φ_G : a diagonal unitary operator (or *phase map*) with the structure

$$\Phi_G \;=\; \left[\bigotimes_{v\in O^{\mathrm{c}}} \mathrm{e}^{-i\alpha_v |1\rangle\langle 1|_v}\right]\left[\prod_{uv\in E} \wedge Z_{u,v}\right]$$

Phase Map Decomposition [B, Danos, Kashefi 2006]

- Phase Map Decompositions describe how one-way measurement patterns evolve in one special case. So, if we can find:
  - Φ_G realising a phase map decomposition for $U_{I \to O}$
  - a flow for (G, I, O)

  then we can obtain a one-way pattern for U .

Phase Map Decomposition [B, Danos, Kashefi 2006]

- Phase Map Decompositions describe how one-way measurement patterns evolve in one special case. So, if we can find:
  - Φ_G realising a phase map decomposition for $U_{I \to O}$
  - a flow for (G, I, O)

  then we can obtain a one-way pattern for U .

For which families of unitary operators U
can we efficiently find phase map decompositions?

Phase Map Decomposition [B, Danos, Kashefi 2006]

- Phase Map Decompositions describe how one-way measurement patterns evolve in one special case. So, if we can find:
  - Φ_G realising a phase map decomposition for $U_{I \to O}$
  - a flow for (G, I, O)

  then we can obtain a one-way pattern for U .

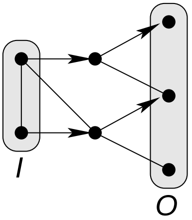
For which families of unitary operators U
can we efficiently find phase map decompositions?

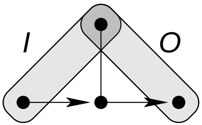
- Candidate sub-problem: given (G, I, O), determine whether it has a flow (and find one if it does).

Outline

1. Introduction
   - The One-Way Measurement Model
   - Flows in one-way patterns
2. Phase Map Decompositions
3. Graph-theoretic results
   - Path covers
   - Influencing walks, Vicious circuits
   - Uniqueness when $|I| = |O|$
4. Efficient algorithms when $|I| = |O|$

Flows describe “Path covers”





Paths taken by following edges $x \to f(x)$:

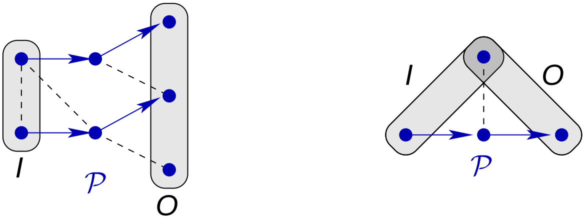
- inputs can only be at the beginning of paths, outputs at the end
- for distinct $x, y \in O^{c}$, we have $f(x) \neq f(y)$

$\Rightarrow$ f describes a set $\mathcal{P}$ of non-intersecting paths* in G, ending in O. Call this a *path cover* for (G, I, O).

(* paths not guaranteed to have non-zero length)

- If $|I| = |O|$, then $\mathcal{P}$ is a collection of paths from I to O.

Flows describe "Path covers"



Paths taken by following edges $x \to f(x)$:

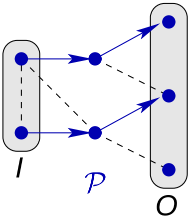
- inputs can only be at the beginning of paths, outputs at the end
- for distinct $x, y \in O^{\mathsf{c}}$, we have $f(x) \neq f(y)$

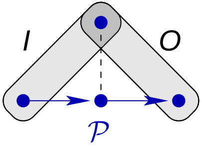
$\Rightarrow$ f describes a set $\mathcal{P}$ of non-intersecting paths* in G, ending in O. Call this a *path cover* for (G, I, O).

(* paths not guaranteed to have non-zero length)

- If $|I| = |O|$, then $\mathcal{P}$ is a collection of paths from I to O.

Flows describe "Path covers"





Paths taken by following edges $x \to f(x)$:

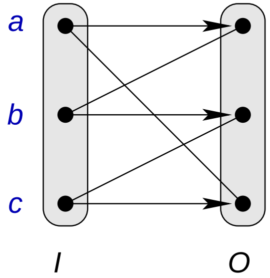
- inputs can only be at the beginning of paths, outputs at the end
- for distinct $x, y \in O^c$, we have $f(x) \neq f(y)$

$\Rightarrow$ f describes a set $\mathcal{P}$ of non-intersecting paths* in G, ending in O. Call this a *path cover* for (G, I, O).

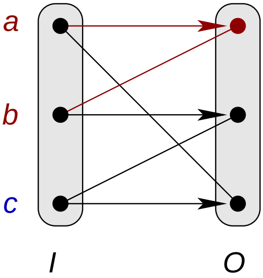
(* paths not guaranteed to have non-zero length)

- If $|I| = |O|$, then $\mathcal{P}$ is a collection of paths from I to O.

Not all path covers are described by flows

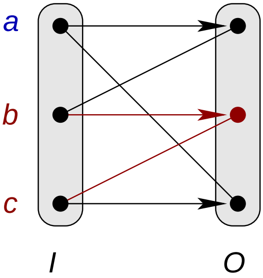


Not all path covers are described by flows



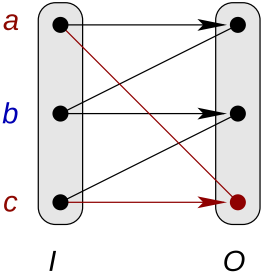
$$b \sim f(a) \quad \Rightarrow \quad a \preccurlyeq b$$

Not all path covers are described by flows



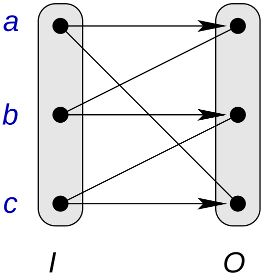
$$b \sim f(a) \quad \Rightarrow \quad a \preccurlyeq b$$
$$c \sim f(b) \quad \Rightarrow \quad b \preccurlyeq c$$

Not all path covers are described by flows



$$b \sim f(a) \;\Rightarrow\; a \preccurlyeq b$$
$$c \sim f(b) \;\Rightarrow\; b \preccurlyeq c$$
$$a \sim f(c) \;\Rightarrow\; c \preccurlyeq a$$

Not all path covers are described by flows



$$
\begin{aligned}
b \sim f(a) \quad &\Rightarrow \quad a \preccurlyeq b \\
c \sim f(b) \quad &\Rightarrow \quad b \preccurlyeq c \\
a \sim f(c) \quad &\Rightarrow \quad c \preccurlyeq a
\end{aligned}
$$

$$a \preccurlyeq b \preccurlyeq c \preccurlyeq a$$

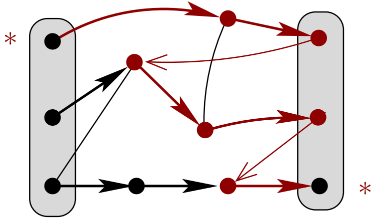
Influencing walks, Vicious circuits

Definition

An **influencing walk** of a path-cover $\mathcal{P}$ is a directed walk in G which can be decomposed into paths of the following two types (using arcs of $\mathcal{P}$):

1. a single arc , $x \longrightarrow y$;
2. a single arc $x \longrightarrow y$,
   followed by any edge $yz \in E$

A **vicious circuit** is an influencing walk which starts and ends at the same vertex.



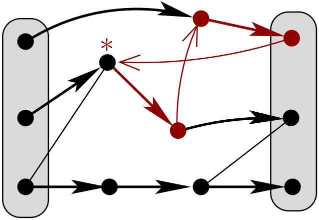
Influencing walks, Vicious circuits

Definition

An **influencing walk** of a path-cover $\mathcal{P}$ is a directed walk in G which can be decomposed into paths of the following two types (using arcs of $\mathcal{P}$):

1. a single arc , $x \rightarrow y$;
2. a single arc $x \rightarrow y$,
   followed by any edge $yz \in E$

A **vicious circuit** is an influencing walk which starts and ends at the same vertex.



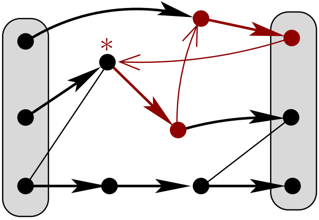
Influencing walks, Vicious circuits

Definition

An **influencing walk** of a path-cover $\mathcal{P}$ is a directed walk in G which can be decomposed into paths of the following two types (using arcs of $\mathcal{P}$):

1. a single arc , $x \to y$;
2. a single arc $x \to y$,
   followed by any edge $yz \in E$

A **vicious circuit** is an influencing walk which starts and ends at the same vertex.

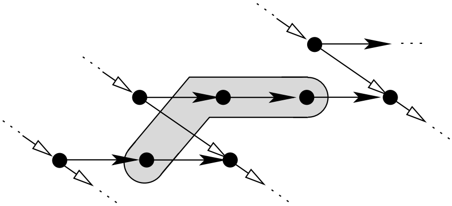


Theorem:

a geometry (G, I, O) has a flow iff it has a path cover with no vicious circuits.

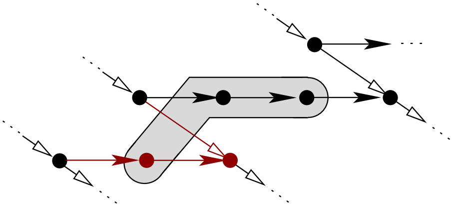
Uniqueness when $|I| = |O|$

- $\mathcal{P}$ a path cover (solid arrows)
- $\mathcal{F}$ a (different) collection of $I - O$ paths (hollow arrows), with the same number of paths
- shaded area: vertices not covered by $\mathcal{F}$



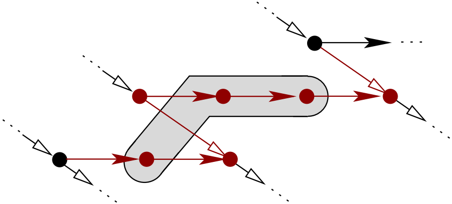
Uniqueness when $|I| = |O|$

- $\mathcal{P}$ a path cover (solid arrows)
- $\mathcal{F}$ a (different) collection of $I - O$ paths (hollow arrows), with the same number of paths
- shaded area: vertices not covered by $\mathcal{F}$



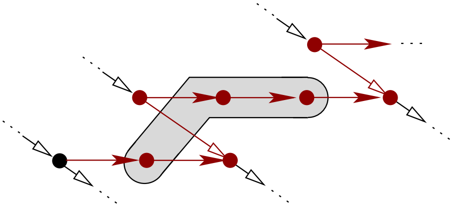
Uniqueness when $|I| = |O|$

- $\mathcal{P}$ a path cover (solid arrows)
- $\mathcal{F}$ a (different) collection of $I - O$ paths (hollow arrows), with the same number of paths
- shaded area: vertices not covered by $\mathcal{F}$



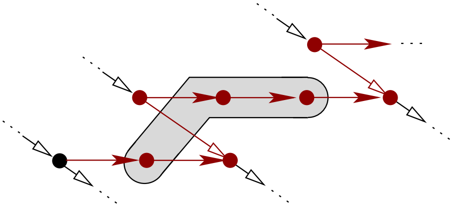
Uniqueness when $|I| = |O|$

- $\mathcal{P}$ a path cover (solid arrows)
- $\mathcal{F}$ a (different) collection of $I - O$ paths (hollow arrows), with the same number of paths
- shaded area: vertices not covered by $\mathcal{F}$



Uniqueness when $|I| = |O|$

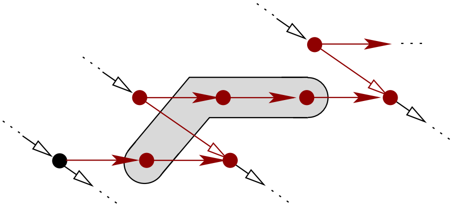
- $\mathcal{P}$ a path cover (solid arrows)
- $\mathcal{F}$ a (different) collection of I – O paths (hollow arrows), with the same number of paths
- shaded area: vertices not covered by $\mathcal{F}$



$\Rightarrow$ we can construct an infinitely long influencing walk — which will eventually close up into a vicious circuit.

Uniqueness when $|I| = |O|$

- $\mathcal{P}$ a path cover (solid arrows)
- $\mathcal{F}$ a (different) collection of I – O paths (hollow arrows), with the same number of paths
- shaded area: vertices not covered by $\mathcal{F}$



Theorem:

if $|I| = |O|$ and (G, I, O) has a path cover $\mathcal{P}$ without vicious circuits, then $\mathcal{P}$ is the only maximum-size collection of disjoint I – O paths.

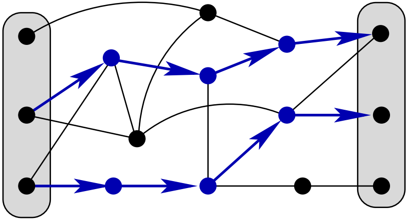
Outline

1. Introduction
   - The One-Way Measurement Model
   - Flows in one-way patterns
2. Phase Map Decompositions
3. Graph-theoretic results
   - Path covers
   - Influencing walks, Vicious circuits
   - Uniqueness when $|I| = |O|$
4. Efficient algorithms when $|I| = |O|$

Finding a function f when $|I| = |O|$

If (G, I, O) has a flow, then to find a path cover without vicious circuits, we simply build a maximum collection of non-intersecting paths.

- Use a modified version of an augmenting path algorithm (e.g. Ford-Fulkerson):

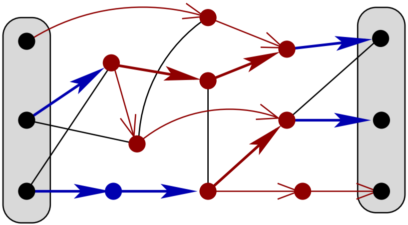


- Time required to find a single augmenting path: $O(m)$
- Time required to find a maximum-size family of paths: $O(km)$
- By uniqueness: if paths don't cover all vertices, there is no flow.

Finding a function f when $|I| = |O|$

If (G, I, O) has a flow, then to find a path cover without vicious circuits, we simply build a maximum collection of non-intersecting paths.

- Use a modified version of an augmenting path algorithm (e.g. Ford-Fulkerson):

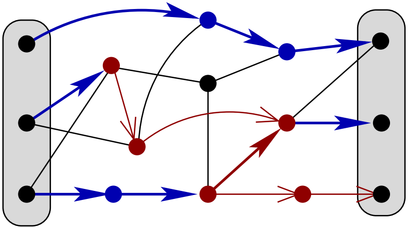


- Time required to find a single augmenting path: $O(m)$
- Time required to find a maximum-size family of paths: $O(km)$
- By uniqueness: if paths don't cover all vertices, there is no flow.

Finding a function f when $|I| = |O|$

If (G, I, O) has a flow, then to find a path cover without vicious circuits, we simply build a maximum collection of non-intersecting paths.

- Use a modified version of an augmenting path algorithm (e.g. Ford-Fulkerson):

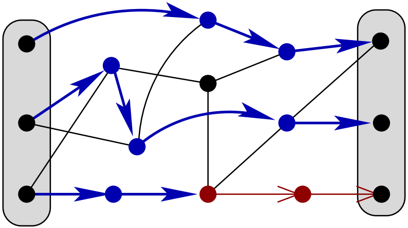


- Time required to find a single augmenting path: $O(m)$
- Time required to find a maximum-size family of paths: $O(km)$
- By uniqueness: if paths don't cover all vertices, there is no flow.

Finding a function f when $|I| = |O|$

If (G, I, O) has a flow, then to find a path cover without vicious circuits, we simply build a maximum collection of non-intersecting paths.

- Use a modified version of an augmenting path algorithm (e.g. Ford-Fulkerson):

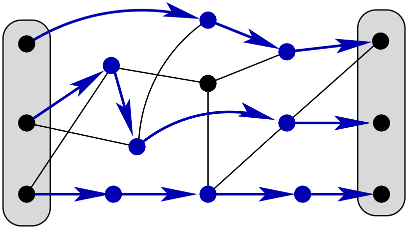


- Time required to find a single augmenting path: $O(m)$
- Time required to find a maximum-size family of paths: $O(km)$
- By uniqueness: if paths don't cover all vertices, there is no flow.

Finding a function f when $|I| = |O|$

If (G, I, O) has a flow, then to find a path cover without vicious circuits, we simply build a maximum collection of non-intersecting paths.

- Use a modified version of an augmenting path algorithm (e.g. Ford-Fulkerson):

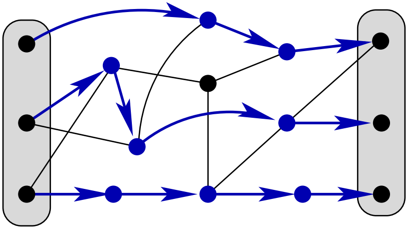


- Time required to find a single augmenting path: $O(m)$
- Time required to find a maximum-size family of paths: $O(km)$
- By uniqueness: if paths don't cover all vertices, there is no flow.

Finding a function f when $|I| = |O|$

If (G, I, O) has a flow, then to find a path cover without vicious circuits, we simply build a maximum collection of non-intersecting paths.

- Use a modified version of an augmenting path algorithm (e.g. Ford-Fulkerson):



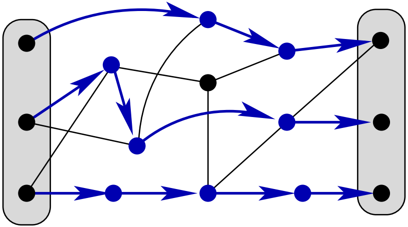
- Time required to find a single augmenting path: $O(m)$
- Time required to find a maximum-size family of paths: $O(km)$
- By uniqueness: if paths don't cover all vertices, there is no flow.

(where $m = |E|$)

Finding a function f when $|I| = |O|$

If (G, I, O) has a flow, then to find a path cover without vicious circuits, we simply build a maximum collection of non-intersecting paths.

- Use a modified version of an augmenting path algorithm (e.g. Ford-Fulkerson):



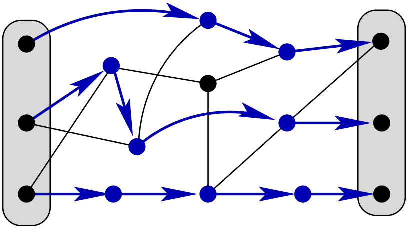
- Time required to find a single augmenting path: $O(m)$
- Time required to find a maximum-size family of paths: $O(km)$
- By uniqueness: if paths don't cover all vertices, there is no flow.

(where $m = |E|$ and $k = |O|$)

Finding a function f when $|I| = |O|$

If (G, I, O) has a flow, then to find a path cover without vicious circuits, we simply build a maximum collection of non-intersecting paths.

- Use a modified version of an augmenting path algorithm (e.g. Ford-Fulkerson):



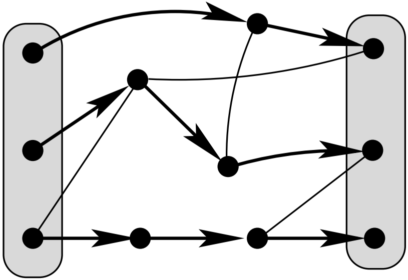
- Time required to find a single augmenting path: $O(m)$
- Time required to find a maximum-size family of paths: $O(km)$
- By uniqueness: if paths don't cover all vertices, there is no flow.

(where $m = |E|$ and $k = |O|$)

Finding a partial order $\preccurlyeq$ when $|I| = |O|$

If (G, I, O) has a flow, then any path-cover $\mathcal{P}$ has no vicious circuits.

- Use a depth-first search along the influencing walks of $\mathcal{P}$ to look for vicious circuits.

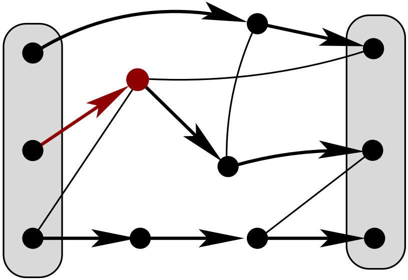


- By uniqueness: if a vicious circuit is found, there is no flow.
- Can reduce work by storing reachability information at vertices
  - If there are no vicious circuits, this builds a partial order $\preccurlyeq$
- Time required: $O(km)$ (where $m = |E|$ and $k = |O|$)

Finding a partial order $\preccurlyeq$ when $|I| = |O|$

If (G, I, O) has a flow, then any path-cover $\mathcal{P}$ has no vicious circuits.

- Use a depth-first search along the influencing walks of $\mathcal{P}$ to look for vicious circuits.

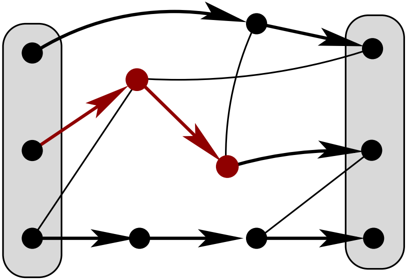


- By uniqueness: if a vicious circuit is found, there is no flow.
- Can reduce work by storing reachability information at vertices
  - If there are no vicious circuits, this builds a partial order $\preccurlyeq$
- Time required: $O(km)$ (where $m = |E|$ and $k = |O|$)

Finding a partial order $\preccurlyeq$ when $|I| = |O|$

If (G, I, O) has a flow, then any path-cover $\mathcal{P}$ has no vicious circuits.

- Use a depth-first search along the influencing walks of $\mathcal{P}$ to look for vicious circuits.

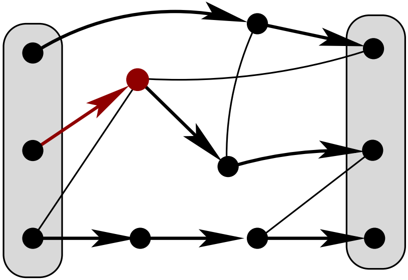


- By uniqueness: if a vicious circuit is found, there is no flow.
- Can reduce work by storing reachability information at vertices
  - If there are no vicious circuits, this builds a partial order $\preccurlyeq$
- Time required: $O(km)$ (where $m = |E|$ and $k = |O|$)

Finding a partial order $\preccurlyeq$ when $|I| = |O|$

If (G, I, O) has a flow, then any path-cover $\mathcal{P}$ has no vicious circuits.

- Use a depth-first search along the influencing walks of $\mathcal{P}$ to look for vicious circuits.

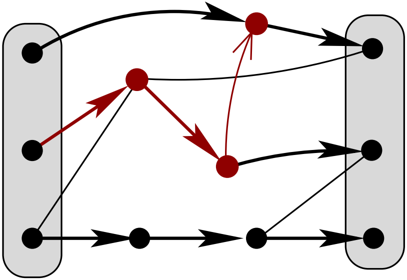


- By uniqueness: if a vicious circuit is found, there is no flow.
- Can reduce work by storing reachability information at vertices
  - If there are no vicious circuits, this builds a partial order $\preccurlyeq$
- Time required: $O(km)$ (where $m = |E|$ and $k = |O|$)

Finding a partial order $\preccurlyeq$ when $|I| = |O|$

If (G, I, O) has a flow, then any path-cover $\mathcal{P}$ has no vicious circuits.

- Use a depth-first search along the influencing walks of $\mathcal{P}$ to look for vicious circuits.

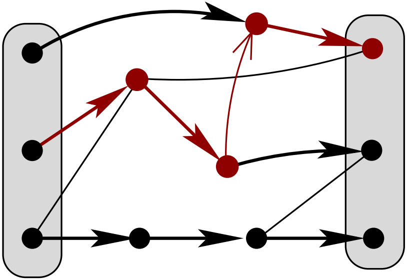


- By uniqueness: if a vicious circuit is found, there is no flow.
- Can reduce work by storing reachability information at vertices
  - If there are no vicious circuits, this builds a partial order $\preccurlyeq$
- Time required: $O(km)$ (where $m = |E|$ and $k = |O|$)

Finding a partial order $\preccurlyeq$ when $|I| = |O|$

If (G, I, O) has a flow, then any path-cover $\mathcal{P}$ has no vicious circuits.

- Use a depth-first search along the influencing walks of $\mathcal{P}$ to look for vicious circuits.

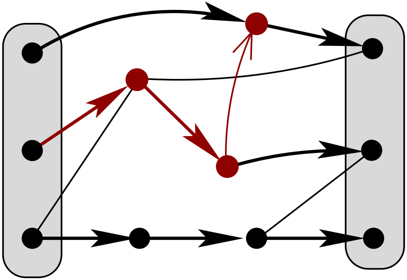


- By uniqueness: if a vicious circuit is found, there is no flow.
- Can reduce work by storing reachability information at vertices
  - If there are no vicious circuits, this builds a partial order $\preccurlyeq$
- Time required: $O(km)$ (where $m = |E|$ and $k = |O|$)

Finding a partial order $\preccurlyeq$ when $|I| = |O|$

If (G, I, O) has a flow, then any path-cover $\mathcal{P}$ has no vicious circuits.

- Use a depth-first search along the influencing walks of $\mathcal{P}$ to look for vicious circuits.

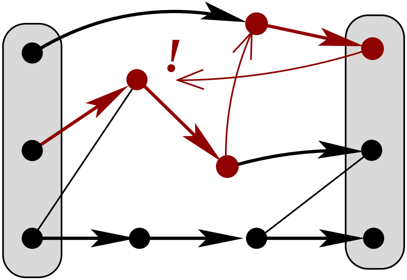


- By uniqueness: if a vicious circuit is found, there is no flow.
- Can reduce work by storing reachability information at vertices
  - If there are no vicious circuits, this builds a partial order $\preccurlyeq$
- Time required: $O(km)$ (where $m = |E|$ and $k = |O|$)

Finding a partial order $\preccurlyeq$ when $|I| = |O|$

If (G, I, O) has a flow, then any path-cover $\mathcal{P}$ has no vicious circuits.

- Use a depth-first search along the influencing walks of $\mathcal{P}$ to look for vicious circuits.

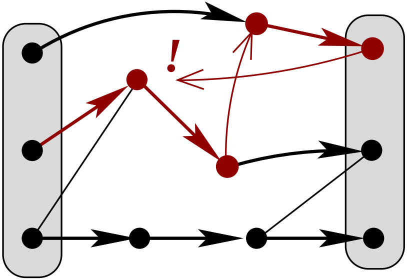


- By uniqueness: if a vicious circuit is found, there is no flow.
- Can reduce work by storing reachability information at vertices
  - If there are no vicious circuits, this builds a partial order $\preccurlyeq$
- Time required: $O(km)$ (where $m = |E|$ and $k = |O|$)

Finding a partial order $\preccurlyeq$ when $|I| = |O|$

If (G, I, O) has a flow, then any path-cover $\mathcal{P}$ has no vicious circuits.

- Use a depth-first search along the influencing walks of $\mathcal{P}$ to look for vicious circuits.

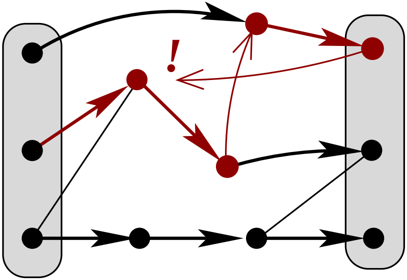


- By uniqueness: if a vicious circuit is found, there is no flow.
- Can reduce work by storing reachability information at vertices
  - If there are no vicious circuits, this builds a partial order $\preccurlyeq$
- Time required: $O(km)$ (where $m = |E|$ and $k = |O|$)

Finding a partial order $\preccurlyeq$ when $|I| = |O|$

If (G, I, O) has a flow, then any path-cover $\mathcal{P}$ has no vicious circuits.

- Use a depth-first search along the influencing walks of $\mathcal{P}$ to look for vicious circuits.

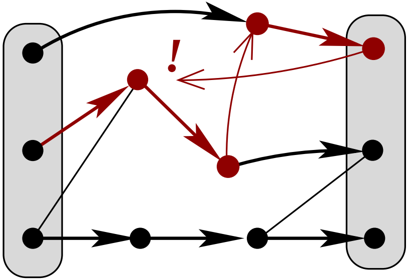


- By uniqueness: if a vicious circuit is found, there is no flow.
- Can reduce work by storing reachability information at vertices
  - If there are no vicious circuits, this builds a partial order $\preccurlyeq$
- Time required: $O(km)$ (where $m = |E|$ and $k = |O|$)

Finding a partial order $\preccurlyeq$ when $|I| = |O|$

If (G, I, O) has a flow, then any path-cover $\mathcal{P}$ has no vicious circuits.

- Use a depth-first search along the influencing walks of $\mathcal{P}$ to look for vicious circuits.



- By uniqueness: if a vicious circuit is found, there is no flow.
- Can reduce work by storing reachability information at vertices
  - If there are no vicious circuits, this builds a partial order $\preccurlyeq$
- Time required: $O(km)$ (where $m = |E|$ and $k = |O|$)

Summary

- Flows help describe how information “flows” in the one-way measurement model.
- Finding a flow for a given geometry (G, I, O) seems a natural sub-problem for finding *phase map decompositions*.
- Using graph-theoretic techniques, we can efficiently find flows when $|I| = |O|$.

- Open problems:
  - Can we find flows efficiently in cases where $|I| < |O|$?
  - Find families of unitaries where we can efficiently find phase-map decompositions!

References

Danos, Kashefi.
Determinism in the one-way model.
quant-ph/0506062 (2005).

B, Danos, Kashefi.
Phase map decompositions for unitaries.
quant-ph/0603266 (2006).

Diestel.
Graph Theory. 3rd ed.
Springer-Verlag (2005).

Cormen, Leiserson, Rivest, Stein.
Introduction to Algorithms, 2nd ed.
MIT Press and McGraw-Hill (2001).