An algebraic interpretation of Pauli flow, leading to faster flow-finding algorithms

Piotr Mitosek[1] Miriam Backens[2]

[1]School of Computer Science, University of Birmingham, UK
[2]Inria & Loria, Nancy, France

18 November 2024

Introduction and reminder of earlier talk

Reminder of earlier talk

- Graph states are resource states in MBQC,

Reminder of earlier talk

- Graph states are resource states in MBQC,
- A graph state with a choice of measurement bases form a labelled open graph,

Reminder of earlier talk

- Graph states are resource states in MBQC,
- A graph state with a choice of measurement bases form a labelled open graph,
- Robustly deterministic computation corresponds to the existence of Pauli flow on the labelled open graph,

Reminder of earlier talk

- Graph states are resource states in MBQC,
- A graph state with a choice of measurement bases form a labelled open graph,
- Robustly deterministic computation corresponds to the existence of Pauli flow on the labelled open graph,
- Pauli flow is more general than (extended) generalized flow,

Reminder of earlier talk

- Graph states are resource states in MBQC,
- A graph state with a choice of measurement bases form a labelled open graph,
- Robustly deterministic computation corresponds to the existence of Pauli flow on the labelled open graph,
- Pauli flow is more general than (extended) generalized flow,
- Pauli flow can be found in polynomial time,

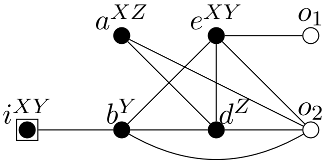
Reminder of earlier talk

- Graph states are resource states in MBQC,
- A graph state with a choice of measurement bases form a labelled open graph,
- Robustly deterministic computation corresponds to the existence of Pauli flow on the labelled open graph,
- Pauli flow is more general than (extended) generalized flow,
- Pauli flow can be found in polynomial time,
- Currently, polynomial time circuit extraction from ZX requires the diagram to exhibit Pauli flow.

Reminder – labelled open graphs

A labelled open graph is a tuple (G, I, O, λ) where:

- $G = (V, E)$ is a simple graph,
- $I \subseteq V$ is a set of inputs,
- $O \subseteq V$ is a set of outputs,
- $\lambda\colon \bar{O} \to \{XY, YZ, XZ, X, Y, Z\}$ is a measurement labelling.



Reminder – Pauli flow

A Pauli flow on a labelled open graph is a pair $(c, \prec)$ where:

- $c\colon \bar{O} \to \mathcal{P}(\bar{I})$ is a correction function,
- $\prec$ is a partial order on $\bar{O}$,

satisfying many, many conditions. . .

(Focused) Pauli flow conditions

For all $u \in \bar{O}$:

- $\forall v \in c(u). u \neq v \wedge \lambda(v) \notin \{X, Y\} \Rightarrow u \prec v$
- $\forall v \in \mathrm{Odd}(c(u)). u \neq v \wedge \lambda(v) \notin \{Y, Z\} \Rightarrow u \prec v$
- $\forall v \in \bar{O}. \neg(u \prec v) \wedge u \neq v \wedge \lambda(v) = Y \Rightarrow (v \in c(u) \Leftrightarrow v \in \mathrm{Odd}(c(u)))$
- $\lambda(u) = XY \Rightarrow u \notin c(u) \wedge u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = XZ \Rightarrow u \in c(u) \wedge u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = YZ \Rightarrow u \in c(u) \wedge u \notin \mathrm{Odd}(c(u))$
- $\lambda(u) = X \Rightarrow u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = Z \Rightarrow u \in c(u)$
- $\lambda(u) = Y \Rightarrow (u \in c(u) \oplus u \in \mathrm{Odd}(c(u)))$, where $\oplus$ stands for XOR.
- $\forall w \in (\bar{O} \setminus \{u\}) \cap c(u). \lambda(w) \in \{XY, X, Y\}$
- $\forall w \in (\bar{O} \setminus \{u\}) \cap \mathrm{Odd}(c(u)). \lambda(w) \in \{XZ, YZ, Y, Z\}$
- $\forall w \in (\bar{O} \setminus \{u\}). \lambda(w) = Y \Rightarrow (w \in c(u) \Leftrightarrow w \in \mathrm{Odd}(c(u)))$

(Focused) Pauli flow conditions

For all $u \in \bar{O}$:

- $\forall v \in c(u). u \neq v \wedge \lambda(v) \notin \{X, Y\} \Rightarrow u \prec v$
- $\forall v \in \mathrm{Odd}(c(u)). u \neq v \wedge \lambda(v) \notin \{Y, Z\} \Rightarrow u \prec v$
- $\forall v \in \bar{O}. \neg(u \prec v) \wedge u \neq v \wedge \lambda(v) = Y \Rightarrow (v \in c(u) \Leftrightarrow v \in \mathrm{Odd}(c(u)))$
- $\lambda(u) = XY \Rightarrow u \notin c(u) \wedge u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = XZ \Rightarrow u \in c(u) \wedge u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = YZ \Rightarrow u \in c(u) \wedge u \notin \mathrm{Odd}(c(u))$
- $\lambda(u) = X \Rightarrow u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = Z \Rightarrow u \in c(u)$
- $\lambda(u) = Y \Rightarrow (u \in c(u) \oplus u \in \mathrm{Odd}(c(u)))$, where $\oplus$ stands for XOR.
- $\forall w \in (\bar{O} \setminus \{u\}) \cap c(u). \lambda(w) \in \{XY, X, Y\}$
- $\forall w \in (\bar{O} \setminus \{u\}) \cap \mathrm{Odd}(c(u)). \lambda(w) \in \{XZ, YZ, Y, Z\}$
- $\forall w \in (\bar{O} \setminus \{u\}). \lambda(w) = Y \Rightarrow (w \in c(u) \Leftrightarrow w \in \mathrm{Odd}(c(u)))$

(Focused) Pauli flow conditions

For all $u \in \bar{O}$:

- $\forall v \in c(u). u \neq v \wedge \lambda(v) \notin \{X, Y\} \Rightarrow u \prec v$
- $\forall v \in \mathrm{Odd}(c(u)). u \neq v \wedge \lambda(v) \notin \{Y, Z\} \Rightarrow u \prec v$
- $\forall v \in \bar{O}. \neg(u \prec v) \wedge u \neq v \wedge \lambda(v) = Y \Rightarrow (v \in c(u) \Leftrightarrow v \in \mathrm{Odd}(c(u)))$
- $\lambda(u) = XY \Rightarrow u \notin c(u) \wedge u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = XZ \Rightarrow u \in c(u) \wedge u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = YZ \Rightarrow u \in c(u) \wedge u \notin \mathrm{Odd}(c(u))$
- $\lambda(u) = X \Rightarrow u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = Z \Rightarrow u \in c(u)$
- $\lambda(u) = Y \Rightarrow (u \in c(u) \oplus u \in \mathrm{Odd}(c(u)))$, where $\oplus$ stands for XOR.
- $\forall w \in (\bar{O} \setminus \{u\}) \cap c(u). \lambda(w) \in \{XY, X, Y\}$
- $\forall w \in (\bar{O} \setminus \{u\}) \cap \mathrm{Odd}(c(u)). \lambda(w) \in \{XZ, YZ, Y, Z\}$
- $\forall w \in (\bar{O} \setminus \{u\}). \lambda(w) = Y \Rightarrow (w \in c(u) \Leftrightarrow w \in \mathrm{Odd}(c(u)))$

(Focused) Pauli flow conditions

For all $u \in \bar{O}$:

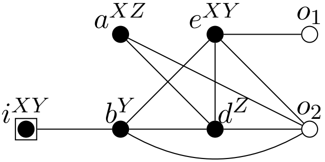
- $\forall v \in c(u). u \neq v \wedge \lambda(v) \notin \{X, Y\} \Rightarrow u \prec v$
- $\forall v \in \mathrm{Odd}(c(u)). u \neq v \wedge \lambda(v) \notin \{Y, Z\} \Rightarrow u \prec v$
- $\forall v \in \bar{O}. \neg(u \prec v) \wedge u \neq v \wedge \lambda(v) = Y \Rightarrow (v \in c(u) \Leftrightarrow v \in \mathrm{Odd}(c(u)))$
- $\lambda(u) = XY \Rightarrow u \notin c(u) \wedge u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = XZ \Rightarrow u \in c(u) \wedge u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = YZ \Rightarrow u \in c(u) \wedge u \notin \mathrm{Odd}(c(u))$
- $\lambda(u) = X \Rightarrow u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = Z \Rightarrow u \in c(u)$
- $\lambda(u) = Y \Rightarrow (u \in c(u) \oplus u \in \mathrm{Odd}(c(u)))$, where $\oplus$ stands for XOR.
- $\forall w \in (\bar{O} \setminus \{u\}) \cap c(u). \lambda(w) \in \{XY, X, Y\}$
- $\forall w \in (\bar{O} \setminus \{u\}) \cap \mathrm{Odd}(c(u)). \lambda(w) \in \{XZ, YZ, Y, Z\}$
- $\forall w \in (\bar{O} \setminus \{u\}). \lambda(w) = Y \Rightarrow (w \in c(u) \Leftrightarrow w \in \mathrm{Odd}(c(u)))$

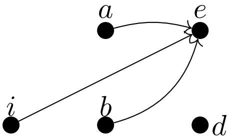
(Focused) Pauli flow conditions

For all $u \in \bar{O}$:

- $\forall v \in c(u). u \neq v \wedge \lambda(v) \notin \{X, Y\} \Rightarrow u \prec v$
- $\forall v \in \mathrm{Odd}(c(u)). u \neq v \wedge \lambda(v) \notin \{Y, Z\} \Rightarrow u \prec v$
- $\forall v \in \bar{O}. \neg(u \prec v) \wedge u \neq v \wedge \lambda(v) = Y \Rightarrow (v \in c(u) \Leftrightarrow v \in \mathrm{Odd}(c(u)))$
- $\lambda(u) = XY \Rightarrow u \notin c(u) \wedge u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = XZ \Rightarrow u \in c(u) \wedge u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = YZ \Rightarrow u \in c(u) \wedge u \notin \mathrm{Odd}(c(u))$
- $\lambda(u) = X \Rightarrow u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = Z \Rightarrow u \in c(u)$
- $\lambda(u) = Y \Rightarrow (u \in c(u) \oplus u \in \mathrm{Odd}(c(u)))$, where $\oplus$ stands for XOR.
- $\forall w \in (\bar{O} \setminus \{u\}) \cap c(u). \lambda(w) \in \{XY, X, Y\}$
- $\forall w \in (\bar{O} \setminus \{u\}) \cap \mathrm{Odd}(c(u)). \lambda(w) \in \{XZ, YZ, Y, Z\}$
- $\forall w \in (\bar{O} \setminus \{u\}). \lambda(w) = Y \Rightarrow (w \in c(u) \Leftrightarrow w \in \mathrm{Odd}(c(u)))$

Pauli flow – example with flow





	λ	$c(v)$	$\mathrm{Odd}(c(v))$
i	XY	b, e, o_1	i, b, o_1
a	XZ	a, e, o_1, o_2	a, d, o_1
b	Y	e	b, d, o_1, o_2
e	XY	o_1	e
d	Z	d, o_2	d, o_2

Issues with Pauli flow

1. Pauli flow definition is very long and hard to work with.
2. Flow-finding algorithms have to solve many linear systems and hence are slow:
   - gflow finding runs in $\mathcal{O}(n^4)$,
   - Pauli flow finding runs in $\mathcal{O}(n^5)$.

Issues with Pauli flow

1. Pauli flow definition is very long and hard to work with.
2. Flow-finding algorithms have to solve many linear systems and hence are slow:
    - gflow finding runs in $\mathcal{O}(n^4)$,
    - Pauli flow finding runs in $\mathcal{O}(n^5)$.

Issues with Pauli flow

1. Pauli flow definition is very long and hard to work with.
2. Flow-finding algorithms have to solve many linear systems and hence are slow:
   - gflow finding runs in $\mathcal{O}(n^4)$,
   - Pauli flow finding runs in $\mathcal{O}(n^5)$.

Issues with Pauli flow

1. Pauli flow definition is very long and hard to work with.
2. Flow-finding algorithms have to solve many linear systems and hence are slow:
    - gflow finding runs in $\mathcal{O}(n^4)$,
    - Pauli flow finding runs in $\mathcal{O}(n^5)$.

This work:

1. We propose a new algebraic interpretation of Pauli flow that simplifies conditions for flow existence.
2. We reduce complexity of Pauli flow-finding to $\mathcal{O}(n^3)$.

Issues with Pauli flow

1. Pauli flow definition is very long and hard to work with.
2. Flow-finding algorithms have to solve many linear systems and hence are slow:
   - gflow finding runs in $\mathcal{O}(n^4)$,
   - Pauli flow finding runs in $\mathcal{O}(n^5)$.

This work:

1. We propose a new algebraic interpretation of Pauli flow that simplifies conditions for flow existence.
2. We reduce complexity of Pauli flow-finding to $\mathcal{O}(n^3)$.

Issues with Pauli flow

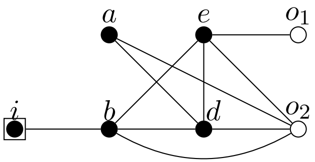
1. Pauli flow definition is very long and hard to work with.
2. Flow-finding algorithms have to solve many linear systems and hence are slow:
    - gflow finding runs in $\mathcal{O}(n^4)$,
    - Pauli flow finding runs in $\mathcal{O}(n^5)$.

This work:

1. We propose a new algebraic interpretation of Pauli flow that simplifies conditions for flow existence.
2. We reduce complexity of Pauli flow-finding to $\mathcal{O}(n^3)$.

Algebraic interpretation of Pauli flow

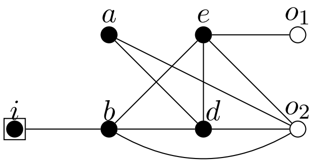
Reduced Adjacency Matrix



$$
\begin{array}{c|ccccccc|c}
 & i & a & b & e & d & o_1 & o_2 & \\
\hline
i & 0 & 0 & 1 & 0 & 0 & 0 & 0 & i \\
a & 0 & 0 & 0 & 0 & 1 & 0 & 1 & a \\
b & 1 & 0 & 0 & 1 & 1 & 0 & 1 & b \\
e & 0 & 0 & 1 & 0 & 1 & 1 & 1 & e \\
d & 0 & 1 & 1 & 1 & 0 & 0 & 1 & d \\
o_1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & o_1 \\
o_2 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & o_2 \\
\hline
 & i & a & b & e & d & o_1 & o_2 &
\end{array}
$$

- Start with adjacency matrix
- Remove output rows
- Remove input columns

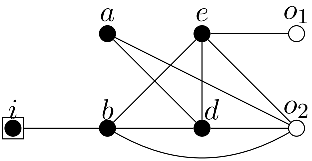
Reduced Adjacency Matrix



$$
\begin{array}{c} \\ i \\ a \\ b \\ e \\ d \\ o_1 \\ o_2 \\ \\ \end{array}
\begin{array}{c}
\begin{array}{ccccccc} i & a & b & e & d & o_1 & o_2 \end{array} \\
\begin{pmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0
\end{pmatrix} \\
\begin{array}{ccccccc} i & a & b & e & d & o_1 & o_2 \end{array}
\end{array}
\begin{array}{c} \\ i \\ a \\ b \\ e \\ d \\ o_1 \\ o_2 \\ \\ \end{array}
$$

- Start with adjacency matrix
- Remove output rows
- Remove input columns

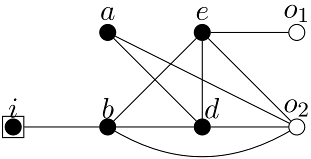
Reduced Adjacency Matrix



	i	a	b	e	d	o_1	o_2	
i	0	0	1	0	0	0	0	i
a	0	0	0	0	1	0	1	a
b	1	0	0	1	1	0	1	b
e	0	0	1	0	1	1	1	e
d	0	1	1	1	0	0	1	d
o_1	0	0	0	1	0	0	0	o_1
o_2	0	1	1	1	1	0	0	o_2
	i	a	b	e	d	o_1	o_2	

- Start with adjacency matrix
- Remove output rows
- Remove input columns

Reduced Adjacency Matrix

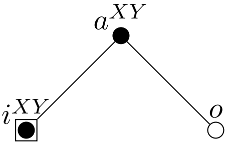


	a	b	e	d	o_1	o_2	
i	0	1	0	0	0	0	i
a	0	0	0	1	0	1	a
b	0	0	1	1	0	1	b
e	0	1	0	1	1	1	e
d	1	1	1	0	0	1	d
	a	b	e	d	o_1	o_2	

- Start with adjacency matrix
- Remove output rows
- Remove input columns

Previous algebraic interpretation – XY only case[1]

- Let $\Gamma = (G, I, O, \lambda)$ where $\lambda \equiv XY$.
- Let A be the reduced adjacency matrix of Γ.
- Γ has gflow if and only if A has a right inverse C that is a DAG
- Columns of C encode the correction function.
- DAG formed by C encodes the partial order.

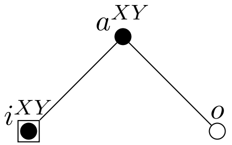


$$A = \begin{array}{c} \\ i \\ a \\ \\ \end{array}\begin{array}{c} \begin{matrix} a & o \end{matrix} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{matrix} a & o \end{matrix} \end{array}\begin{array}{c} \\ i \\ a \\ \\ \end{array} \qquad C = \begin{array}{c} \\ a \\ o \\ \\ \end{array}\begin{array}{c} \begin{matrix} i & a \end{matrix} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{matrix} i & a \end{matrix} \end{array}\begin{array}{c} \\ a \\ o \\ \\ \end{array}$$

[1] Mhalla et al. (2010), arXiv:1006.2616

Previous algebraic interpretation – XY only case[1]

- Let $\Gamma = (G, I, O, \lambda)$ where $\lambda \equiv XY$.
- Let A be the reduced adjacency matrix of Γ.
- Γ has gflow if and only if A has a right inverse C that is a DAG
- Columns of C encode the correction function.
- DAG formed by C encodes the partial order.

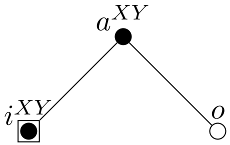


$$A = \begin{array}{c} \\ i \\ a \\ \\ \end{array}\begin{array}{c} \begin{array}{cc} a & o \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{array}{cc} a & o \end{array} \end{array}\begin{array}{c} \\ i \\ a \\ \\ \end{array} \qquad C = \begin{array}{c} \\ a \\ o \\ \\ \end{array}\begin{array}{c} \begin{array}{cc} i & a \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{array}{cc} i & a \end{array} \end{array}\begin{array}{c} \\ a \\ o \\ \\ \end{array}$$

[1]Mhalla et al. (2010), arXiv:1006.2616

Previous algebraic interpretation – XY only case[1]

- Let $\Gamma = (G, I, O, \lambda)$ where $\lambda \equiv XY$.
- Let A be the reduced adjacency matrix of Γ.
- Γ has gflow if and only if A has a right inverse C that is a DAG
- Columns of C encode the correction function.
- DAG formed by C encodes the partial order.



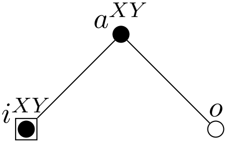
$$A = \begin{array}{c} \\ i \\ a \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} a & o \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{array}{cc} a & o \end{array} \end{array} \begin{array}{c} \\ i \\ a \\ \\ \end{array} \qquad C = \begin{array}{c} \\ a \\ o \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} i & a \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{array}{cc} i & a \end{array} \end{array} \begin{array}{c} \\ a \\ o \\ \\ \end{array}$$

Has gflow.

[1] Mhalla et al. (2010), arXiv:1006.2616

Previous algebraic interpretation – XY only case[1]

- Let $\Gamma = (G, I, O, \lambda)$ where $\lambda \equiv XY$.
- Let A be the reduced adjacency matrix of Γ.
- Γ has gflow if and only if A has a right inverse C that is a DAG
- Columns of C encode the correction function.
- DAG formed by C encodes the partial order.



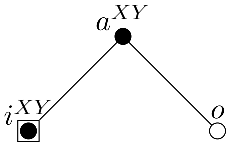
$$A = \begin{array}{c|cc|c} & a & o & \\ i & 1 & 0 & i \\ a & 0 & 1 & a \\ & a & o & \end{array} \qquad C = \begin{array}{c|cc|c} & i & a & \\ a & 1 & 0 & a \\ o & 0 & 1 & o \\ & i & a & \end{array}$$

Has gflow.

[1] Mhalla et al. (2010), arXiv:1006.2616

Previous algebraic interpretation – XY only case[1]

- Let $\Gamma = (G, I, O, \lambda)$ where $\lambda \equiv XY$.
- Let A be the reduced adjacency matrix of Γ.
- Γ has gflow if and only if A has a right inverse C that is a DAG
- Columns of C encode the correction function.
- DAG formed by C encodes the partial order.



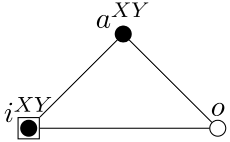
$$A = \begin{array}{c} \\ i \\ a \\ \\ \end{array}\begin{array}{c} \begin{array}{cc} a & o \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{array}{cc} a & o \end{array} \end{array}\begin{array}{c} \\ i \\ a \\ \\ \end{array} \qquad C = \begin{array}{c} \\ a \\ o \\ \\ \end{array}\begin{array}{c} \begin{array}{cc} i & a \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{array}{cc} i & a \end{array} \end{array}\begin{array}{c} \\ a \\ o \\ \\ \end{array}$$

Has gflow.

[1] Mhalla et al. (2010), arXiv:1006.2616

Previous algebraic interpretation – XY only case[1]

- Let $\Gamma = (G, I, O, \lambda)$ where $\lambda \equiv XY$.
- Let A be the reduced adjacency matrix of Γ.
- Γ has gflow if and only if A has a right inverse C that is a DAG
- Columns of C encode the correction function.
- DAG formed by C encodes the partial order.

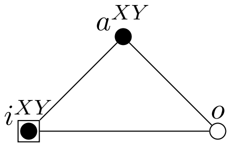


$$A = \begin{array}{c|cc} & a & o \\ \hline i & 1 & 1 \\ a & 0 & 1 \end{array} \qquad C = \begin{array}{c|cc} & i & a \\ \hline a & 1 & 1 \\ o & 0 & 1 \end{array}$$

[1] Mhalla et al. (2010), arXiv:1006.2616

Previous algebraic interpretation – XY only case[1]

- Let $\Gamma = (G, I, O, \lambda)$ where $\lambda \equiv XY$.
- Let A be the reduced adjacency matrix of Γ.
- Γ has gflow if and only if A has a right inverse C that is a DAG
- Columns of C encode the correction function.
- DAG formed by C encodes the partial order.

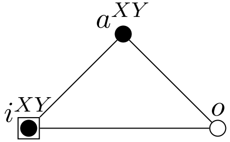


$$A = \begin{array}{c} \\ i \\ a \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} a & o \end{array} \\ \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \\ \begin{array}{cc} a & o \end{array} \end{array} \begin{array}{c} \\ i \\ a \\ \\ \end{array} \qquad C = \begin{array}{c} \\ a \\ o \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} i & a \end{array} \\ \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \\ \begin{array}{cc} i & a \end{array} \end{array} \begin{array}{c} \\ a \\ o \\ \\ \end{array}$$

[1] Mhalla et al. (2010), arXiv:1006.2616

Previous algebraic interpretation – XY only case[1]

- Let $\Gamma = (G, I, O, \lambda)$ where $\lambda \equiv XY$.
- Let A be the reduced adjacency matrix of Γ.
- Γ has gflow if and only if A has a right inverse C that is a DAG
- Columns of C encode the correction function.
- DAG formed by C encodes the partial order.

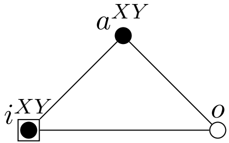


$$A = \begin{array}{c} \\ i \\ a \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} a & o \end{array} \\ \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \\ \begin{array}{cc} a & o \end{array} \end{array} \begin{array}{c} \\ i \\ a \\ \\ \end{array} \qquad C = \begin{array}{c} \\ a \\ o \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} i & a \end{array} \\ \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \\ \begin{array}{cc} i & a \end{array} \end{array} \begin{array}{c} \\ a \\ o \\ \\ \end{array}$$

[1] Mhalla et al. (2010), arXiv:1006.2616

Previous algebraic interpretation – XY only case[1]

- Let $\Gamma = (G, I, O, \lambda)$ where $\lambda \equiv XY$.
- Let A be the reduced adjacency matrix of Γ.
- Γ has gflow if and only if A has a right inverse C that is a DAG
- Columns of C encode the correction function.
- DAG formed by C encodes the partial order.

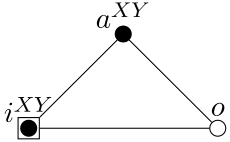


$$A = \begin{array}{c} \\ i \\ a \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} a & o \end{array} \\ \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \\ \begin{array}{cc} a & o \end{array} \end{array} \begin{array}{c} \\ i \\ a \\ \\ \end{array} \qquad C = \begin{array}{c} \\ a \\ o \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} i & a \end{array} \\ \begin{pmatrix} 1 & \boxed{1} \\ 0 & 1 \end{pmatrix} \\ \begin{array}{cc} i & a \end{array} \end{array} \begin{array}{c} \\ a \\ o \\ \\ \end{array}$$

[1] Mhalla et al. (2010), arXiv:1006.2616

Previous algebraic interpretation – XY only case[1]

- Let $\Gamma = (G, I, O, \lambda)$ where $\lambda \equiv XY$.
- Let A be the reduced adjacency matrix of Γ.
- Γ has gflow if and only if A has a right inverse C that is a DAG
- Columns of C encode the correction function.
- DAG formed by C encodes the partial order.



$$A = \begin{array}{c|cc|c} & a & o & \\ \hline i & 1 & 1 & i \\ a & 0 & 1 & a \\ \hline & a & o & \end{array} \qquad C = \begin{array}{c|cc|c} & i & a & \\ \hline a & 1 & \boxed{1} & a \\ o & 0 & 1 & o \\ \hline & i & a & \end{array}$$

No gflow.

[1] Mhalla et al. (2010), arXiv:1006.2616

Other measurements

- Presented version works only for XY-measurements.
- Alternative for X and Z also exists.
- For all six types of measurements, we need something different.

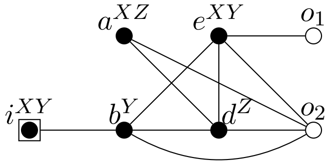
Other measurements

- Presented version works only for XY-measurements.
- Alternative for X and Z also exists.
- For all six types of measurements, we need something different.

Other measurements

- Presented version works only for XY-measurements.
- Alternative for X and Z also exists.
- For all six types of measurements, we need something different.

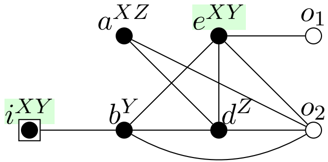
Flow-demand matrix



	a	b	e	d	o_1	o_2	
i							i
a							a
b							b
e							e
d							d
	a	b	e	d	o_1	o_2	

- Shape $\bar{O} \times \bar{I}$
- Rows of XY, X: adjacency
- Rows of Z, YZ, XZ: only 1 at intersection
- Rows of Y: adjacency and 1 at intersection

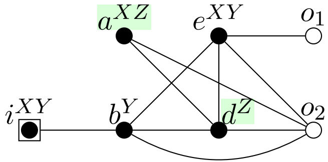
Flow-demand matrix



	a	b	e	d	o_1	o_2
i	0	1	0	0	0	0
a						
b						
e	0	1	0	1	1	1
d						

- Shape $\bar{O} \times \bar{I}$
- Rows of XY, X: adjacency
- Rows of Z, YZ, XZ: only 1 at intersection
- Rows of Y: adjacency and 1 at intersection

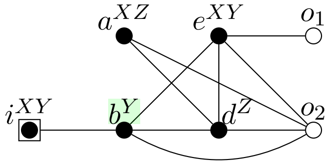
Flow-demand matrix



	a	b	e	d	o_1	o_2	
i	0	1	0	0	0	0	i
a	1	0	0	0	0	0	a
b							b
e	0	1	0	1	1	1	e
d	0	0	0	1	0	0	d
	a	b	e	d	o_1	o_2	

- Shape $\bar{O} \times \bar{I}$
- Rows of XY, X: adjacency
- Rows of Z, YZ, XZ: only 1 at intersection
- Rows of Y: adjacency and 1 at intersection

Flow-demand matrix

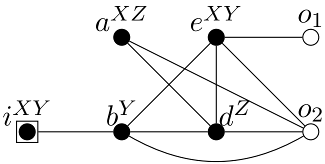


	a	b	e	d	o_1	o_2	
i	0	1	0	0	0	0	i
a	1	0	0	0	0	0	a
b	0	1	1	1	0	1	b
e	0	1	0	1	1	1	e
d	0	0	0	1	0	0	d
	a	b	e	d	o_1	o_2	

- Shape $\bar{O} \times \bar{I}$
- Rows of XY, X: adjacency
- Rows of Z, YZ, XZ: only 1 at intersection
- Rows of Y: adjacency and 1 at intersection

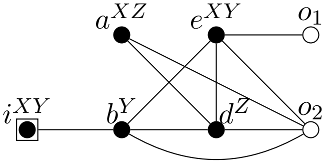
Correction matrix

- Let $c\colon \bar{O} \to \mathcal{P}(\bar{I})$ be a candidate for correction function.
- We encode it into $\bar{I} \times \bar{O}$ matrix.
- Column of v in C represents $c(v)$.



Correction matrix

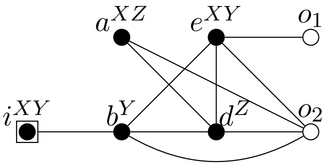
- Let $c\colon \bar{O} \to \mathcal{P}(\bar{I})$ be a candidate for correction function.
- We encode it into $\bar{I} \times \bar{O}$ matrix.
- Column of v in C represents $c(v)$.



	$c(v)$
i	b, e, o_1
a	a, e, o_1, o_2
b	e
e	o_1
d	d, o_2

Correction matrix

- Let $c\colon \bar{O} \to \mathcal{P}(\bar{I})$ be a candidate for correction function.
- We encode it into $\bar{I} \times \bar{O}$ matrix.
- Column of v in C represents $c(v)$.

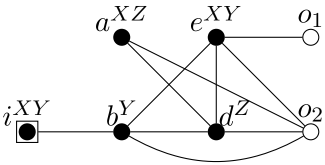


	$c(v)$
i	b, e, o_1
a	a, e, o_1, o_2
b	e
e	o_1
d	d, o_2

$$\begin{array}{c} \\ a \\ b \\ e \\ d \\ o_1 \\ o_2 \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} i & a & b & e & d \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \\ \begin{array}{ccccc} i & a & b & e & d \end{array} \end{array} \begin{array}{c} \\ a \\ b \\ e \\ d \\ o_1 \\ o_2 \\ \\ \end{array}$$

Correction matrix

- Let $c\colon \bar{O} \to \mathcal{P}(\bar{I})$ be a candidate for correction function.
- We encode it into $\bar{I} \times \bar{O}$ matrix.
- Column of v in C represents $c(v)$.

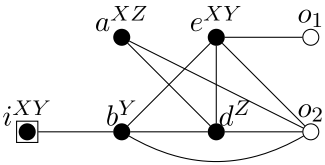


	$c(v)$
i	b, e, o_1
a	a, e, o_1, o_2
b	e
e	o_1
d	d, o_2

	i	a	b	e	d	
a	0	1	0	0	0	a
b	1	0	0	0	0	b
e	1	1	1	0	0	e
d	0	0	0	0	1	d
o_1	1	1	0	1	0	o_1
o_2	0	1	0	0	1	o_2
	i	a	b	e	d	

Correction matrix

- Let $c\colon \bar{O} \to \mathcal{P}(\bar{I})$ be a candidate for correction function.
- We encode it into $\bar{I} \times \bar{O}$ matrix.
- Column of v in C represents $c(v)$.



	$c(v)$
i	b, e, o_1
a	a, e, o_1, o_2
b	e
e	o_1
d	d, o_2

$$
\begin{array}{c|ccccc|c}
 & i & a & b & e & d & \\
a & 0 & 1 & 0 & 0 & 0 & a \\
b & 1 & 0 & 0 & 0 & 0 & b \\
e & 1 & 1 & 1 & 0 & 0 & e \\
d & 0 & 0 & 0 & 0 & 1 & d \\
o_1 & 1 & 1 & 0 & 1 & 0 & o_1 \\
o_2 & 0 & 1 & 0 & 0 & 1 & o_2 \\
 & i & a & b & e & d &
\end{array}
$$

Claim

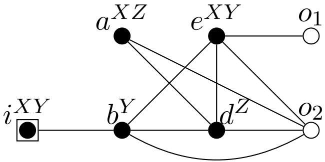
Flow-demand matrix M times correction matrix C equals identity if and only if . . .

$MC = Id$ conditions

For all $u \in \bar{O}$:

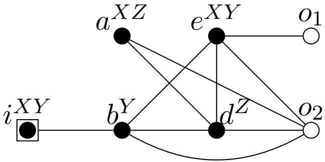
- $\forall v \in c(u).u \neq v \wedge \lambda(v) \notin \{X, Y\} \Rightarrow u \prec v$
- $\forall v \in \mathrm{Odd}(c(u)).u \neq v \wedge \lambda(v) \notin \{Y, Z\} \Rightarrow u \prec v$
- $\forall v \in \bar{O}.\neg(u \prec v) \wedge u \neq v \wedge \lambda(v) = Y \Rightarrow (v \in c(u) \Leftrightarrow v \in \mathrm{Odd}(c(u)))$
- $\lambda(u) = XY \Rightarrow u \notin c(u) \wedge u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = XZ \Rightarrow u \in c(u) \wedge u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = YZ \Rightarrow u \in c(u) \wedge u \notin \mathrm{Odd}(c(u))$
- $\lambda(u) = X \Rightarrow u \in \mathrm{Odd}(c(u))$
- $\lambda(u) = Z \Rightarrow u \in c(u)$
- $\lambda(u) = Y \Rightarrow (u \in c(u) \oplus u \in \mathrm{Odd}(c(u)))$, where $\oplus$ stands for XOR.
- $\forall w \in (\bar{O} \setminus \{u\}) \cap c(u).\lambda(w) \in \{XY, X, Y\}$
- $\forall w \in (\bar{O} \setminus \{u\}) \cap \mathrm{Odd}(c(u)).\lambda(w) \in \{XZ, YZ, Y, Z\}$
- $\forall w \in (\bar{O} \setminus \{u\}).\lambda(w) = Y \Rightarrow (w \in c(u) \Leftrightarrow w \in \mathrm{Odd}(c(u)))$

Example



	λ	$c(v)$	$\mathrm{Odd}(c(v))$
i	XY	b, e, o_1	i, b, o_1
a	XZ	a, e, o_1, o_2	a, d, o_1
b	Y	e	b, d, o_1, o_2
e	XY	o_1	e
d	Z	d, o_2	d, o_2

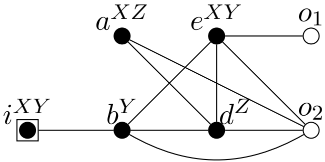
Example



	λ	$c(v)$	$\mathrm{Odd}(c(v))$
i	XY	b, e, o_1	i, b, o_1
a	XZ	a, e, o_1, o_2	a, d, o_1
b	Y	e	b, d, o_1, o_2
e	XY	o_1	e
d	Z	d, o_2	d, o_2

$$
\begin{array}{c}
\begin{array}{cccccc} a & b & e & d & o_1 & o_2 \end{array} \\
\begin{array}{c} i \\ a \\ b \\ e \\ d \end{array}
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}
\begin{array}{c} i \\ a \\ b \\ e \\ d \end{array} \\
\begin{array}{cccccc} a & b & e & d & o_1 & o_2 \end{array}
\end{array}
$$

Example

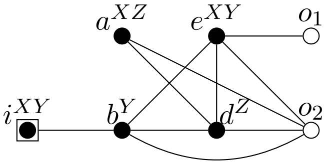


	λ	$c(v)$	$\mathrm{Odd}(c(v))$
i	XY	b, e, o_1	i, b, o_1
a	XZ	a, e, o_1, o_2	a, d, o_1
b	Y	e	b, d, o_1, o_2
e	XY	o_1	e
d	Z	d, o_2	d, o_2

$$\begin{array}{c|cccccc|c} & a & b & e & d & o_1 & o_2 & \\ \hline i & 0 & 1 & 0 & 0 & 0 & 0 & i \\ a & 1 & 0 & 0 & 0 & 0 & 0 & a \\ b & 0 & 1 & 1 & 1 & 0 & 1 & b \\ e & 0 & 1 & 0 & 1 & 1 & 1 & e \\ d & 0 & 0 & 0 & 1 & 0 & 0 & d \\ \hline & a & b & e & d & o_1 & o_2 & \end{array}$$

$$\begin{array}{c|ccccc|c} & i & a & b & e & d & \\ \hline a & 0 & 1 & 0 & 0 & 0 & a \\ b & 1 & 0 & 0 & 0 & 0 & b \\ e & 1 & 1 & 1 & 0 & 0 & e \\ d & 0 & 0 & 0 & 0 & 1 & d \\ o_1 & 1 & 1 & 0 & 1 & 0 & o_1 \\ o_2 & 0 & 1 & 0 & 0 & 1 & o_2 \\ \hline & i & a & b & e & d & \end{array}$$

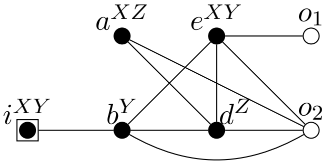
Example



	λ	$c(v)$	$\mathrm{Odd}(c(v))$
i	XY	b, e, o_1	i, b, o_1
a	XZ	a, e, o_1, o_2	a, d, o_1
b	Y	e	b, d, o_1, o_2
e	XY	o_1	e
d	Z	d, o_2	d, o_2

$$
\begin{array}{c|cccccc|c}
 & a & b & e & d & o_1 & o_2 & \\ \hline
i & 0 & 1 & 0 & 0 & 0 & 0 & i \\
a & 1 & 0 & 0 & 0 & 0 & 0 & a \\
b & 0 & 1 & 1 & 1 & 0 & 1 & b \\
e & 0 & 1 & 0 & 1 & 1 & 1 & e \\
d & 0 & 0 & 0 & 1 & 0 & 0 & d \\ \hline
 & a & b & e & d & o_1 & o_2 &
\end{array}
\;
\begin{array}{c|ccccc|c}
 & i & a & b & e & d & \\ \hline
a & 0 & 1 & 0 & 0 & 0 & a \\
b & 1 & 0 & 0 & 0 & 0 & b \\
e & 1 & 1 & 1 & 0 & 0 & e \\
d & 0 & 0 & 0 & 0 & 1 & d \\
o_1 & 1 & 1 & 0 & 1 & 0 & o_1 \\
o_2 & 0 & 1 & 0 & 0 & 1 & o_2 \\ \hline
 & i & a & b & e & d &
\end{array}
=
\begin{array}{c|ccccc|c}
 & i & a & b & e & d & \\ \hline
i & 1 & 0 & 0 & 0 & 0 & i \\
a & 0 & 1 & 0 & 0 & 0 & a \\
b & 0 & 0 & 1 & 0 & 0 & b \\
e & 0 & 0 & 0 & 1 & 0 & e \\
d & 0 & 0 & 0 & 0 & 1 & d \\ \hline
 & i & a & b & e & d &
\end{array}
$$

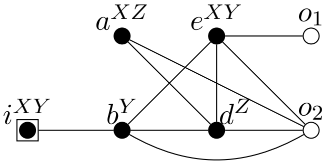
Example



	λ	$c(v)$	$\mathrm{Odd}(c(v))$
i	XY	b, e, o_1	i, b, o_1
a	XZ	a, e, o_1, o_2	a, d, o_1
b	Y	e	b, d, o_1, o_2
e	XY	o_1	e
d	Z	d, o_2	d, o_2

$$\begin{array}{c|cccccc|c} & a & b & e & d & o_1 & o_2 & \\ \hline i & 0 & 1 & 0 & 0 & 0 & 0 & i \\ a & 1 & 0 & 0 & 0 & 0 & 0 & a \\ b & 0 & 1 & 1 & 1 & 0 & 1 & b \\ e & 0 & 1 & 0 & 1 & 1 & 1 & e \\ d & 0 & 0 & 0 & 1 & 0 & 0 & d \\ \hline & a & b & e & d & o_1 & o_2 & \end{array} \quad \begin{array}{c|ccccc|c} & i & a & b & e & d & \\ \hline a & 0 & 1 & 0 & 0 & 0 & a \\ b & 1 & 0 & 0 & 0 & 0 & b \\ e & 1 & 1 & 1 & 0 & 0 & e \\ d & 0 & 0 & 0 & 0 & 1 & d \\ o_1 & 1 & 1 & 0 & 1 & 0 & o_1 \\ o_2 & 0 & 1 & 0 & 0 & 1 & o_2 \\ \hline & i & a & b & e & d & \end{array} = \begin{array}{c|ccccc|c} & i & a & b & e & d & \\ \hline i & 1 & 0 & 0 & 0 & 0 & i \\ a & 0 & 1 & 0 & 0 & 0 & a \\ b & 0 & 0 & 1 & 0 & 0 & b \\ e & 0 & 0 & 0 & 1 & 0 & e \\ d & 0 & 0 & 0 & 0 & 1 & d \\ \hline & i & a & b & e & d & \end{array}$$

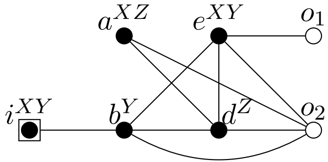
Example



	λ	$c(v)$	$\mathrm{Odd}(c(v))$
i	XY	b, e, o_1	i, b, o_1
a	XZ	a, e, o_1, o_2	a, d, o_1
b	Y	e	b, d, o_1, o_2
e	XY	o_1	e
d	Z	d, o_2	d, o_2

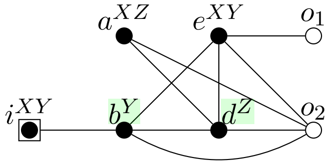
$$
\begin{array}{c}
 \\ i \\ a \\ b \\ e \\ d \\ \\
\end{array}
\begin{array}{c}
\begin{array}{cccccc} a & b & e & d & o_1 & o_2 \end{array} \\
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix} \\
\begin{array}{cccccc} a & b & e & d & o_1 & o_2 \end{array}
\end{array}
\begin{array}{c}
 \\ i \\ a \\ b \\ e \\ d \\ \\
\end{array}
\begin{array}{c}
 \\ a \\ b \\ e \\ d \\ o_1 \\ o_2 \\ \\
\end{array}
\begin{array}{c}
\begin{array}{ccccc} i & a & b & e & d \end{array} \\
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1
\end{pmatrix} \\
\begin{array}{ccccc} i & a & b & e & d \end{array}
\end{array}
\begin{array}{c}
 \\ a \\ b \\ e \\ d \\ o_1 \\ o_2 \\ \\
\end{array}
=
\begin{array}{c}
 \\ i \\ a \\ b \\ e \\ d \\ \\
\end{array}
\begin{array}{c}
\begin{array}{ccccc} i & a & b & e & d \end{array} \\
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{pmatrix} \\
\begin{array}{ccccc} i & a & b & e & d \end{array}
\end{array}
\begin{array}{c}
 \\ i \\ a \\ b \\ e \\ d \\ \\
\end{array}
$$

Order-demand matrix



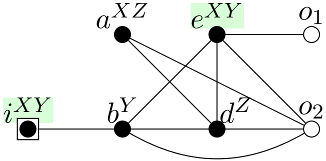
	a	b	e	d	o_1	o_2	
i							i
a							a
b							b
e							e
d							d
	a	b	e	d	o_1	o_2	

- Shape $\bar{O} \times \bar{I}$
- Rows of X, Y, Z: identically 0
- Rows of XY: only 1 at intersection
- Rows of YZ: adjacency
- Rows of XZ: adjacency and 1 at intersection



	a	b	e	d	o_1	o_2	
i							i
a							a
b	0	0	0	0	0	0	b
e							e
d	0	0	0	0	0	0	d
	a	b	e	d	o_1	o_2	

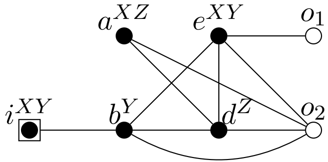
- Shape $\bar{O} \times \bar{I}$
- Rows of X, Y, Z: identically 0
- Rows of XY: only 1 at intersection
- Rows of YZ: adjacency
- Rows of XZ: adjacency and 1 at intersection



	a	b	e	d	o_1	o_2	
i	0	0	0	0	0	0	i
a							a
b	0	0	0	0	0	0	b
e	0	0	1	0	0	0	e
d	0	0	0	0	0	0	d
	a	b	e	d	o_1	o_2	

- Shape $\bar{O} \times \bar{I}$
- Rows of X, Y, Z: identically 0
- Rows of XY: only 1 at intersection
- Rows of YZ: adjacency
- Rows of XZ: adjacency and 1 at intersection

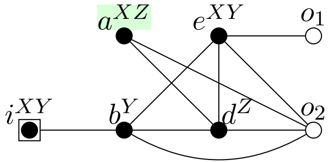
Order-demand matrix



$$
\begin{array}{c} \\ i \\ a \\ b \\ e \\ d \\ \\ \end{array}
\begin{array}{c}
\begin{array}{cccccc} a & b & e & d & o_1 & o_2 \end{array} \\
\left(\begin{array}{cccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}\right) \\
\begin{array}{cccccc} a & b & e & d & o_1 & o_2 \end{array}
\end{array}
\begin{array}{c} \\ i \\ a \\ b \\ e \\ d \\ \\ \end{array}
$$

- Shape $\bar{O} \times \bar{I}$
- Rows of X, Y, Z: identically 0
- Rows of XY: only 1 at intersection
- Rows of YZ: adjacency
- Rows of XZ: adjacency and 1 at intersection

Order-demand matrix



	a	b	e	d	o_1	o_2	
i	0	0	0	0	0	0	i
a	1	0	0	1	0	1	a
b	0	0	0	0	0	0	b
e	0	0	1	0	0	0	e
d	0	0	0	0	0	0	d
	a	b	e	d	o_1	o_2	

- Shape $\bar{O} \times \bar{I}$
- Rows of X, Y, Z: identically 0
- Rows of XY: only 1 at intersection
- Rows of YZ: adjacency
- Rows of XZ: adjacency and 1 at intersection

Full picture

Claim

Flow-demand matrix M times correction matrix C equals identity if and only if . . .

Full picture

Claim

Flow-demand matrix M times correction matrix C equals identity if and only if ...

Claim

Order-demand matrix N times correction matrix C forms a DAG if and only if ...

Full picture

Claim

Flow-demand matrix M times correction matrix C equals identity if and only if . . .

Claim

Order-demand matrix N times correction matrix C forms a DAG if and only if . . .

Theorem

Given a labelled open graph Γ, let M be its flow-demand matrix and N its order-demand matrix. Then, Γ has Pauli flow if and only if there exists a correction matrix C such that $MC = Id$ and NC forms a DAG.

Full picture

Claim

Flow-demand matrix M times correction matrix C equals identity if and only if . . .

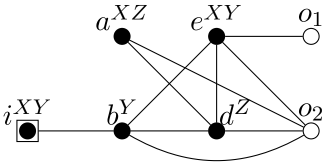
Claim

Order-demand matrix N times correction matrix C forms a DAG if and only if . . .

Theorem

Given a labelled open graph Γ, let M be its flow-demand matrix and N its order-demand matrix. Then, Γ has Pauli flow if and only if there exists a correction matrix C such that $MC = Id$ and NC forms a DAG.

Full example



M	a	b	e	d	o_1	o_2
i	0	1	0	0	0	0
a	1	0	0	0	0	0
b	0	1	1	1	0	1
e	0	1	0	1	1	1
d	0	0	0	1	0	0

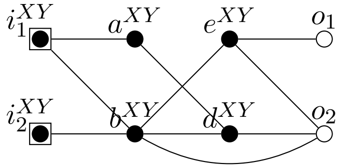
N	a	b	e	d	o_1	o_2
i	0	0	0	0	0	0
a	1	0	0	1	0	1
b	0	0	0	0	0	0
e	0	0	1	0	0	0
d	0	0	0	0	0	0

C	i	a	b	e	d
a	0	1	0	0	0
b	1	0	0	0	0
e	1	1	1	0	0
d	0	0	0	0	1
o_1	1	1	0	1	0
o_2	0	1	0	0	1

NC	i	a	b	e	d
i	0	0	0	0	0
a	0	0	0	0	0
b	0	0	0	0	0
e	1	1	1	0	0
d	0	0	0	0	0

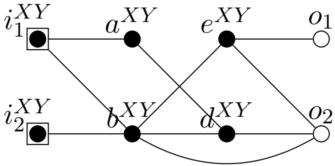
Using algebraic interpretation in proofs

Previous flow-reversibility – XY only case[1]

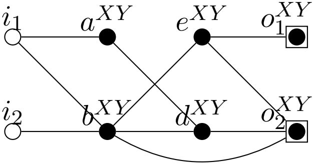


[1] Mhalla et al. (2010), arXiv:1006.2616

Previous flow-reversibility – XY only case[1]

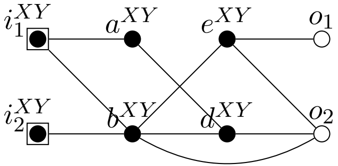


$\downarrow$ switch I and O $\downarrow$

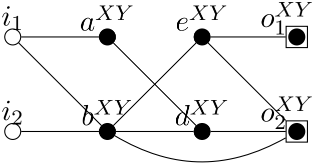


[1]Mhalla et al. (2010), arXiv:1006.2616

Previous flow-reversibility – XY only case[1]



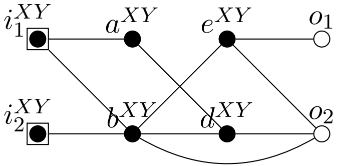
$\downarrow$ switch I and O $\downarrow$



$$
\begin{array}{c} \\ a \\ b \\ e \\ d \\ o_1 \\ o_2 \\ \\ \end{array}
\begin{array}{c} \begin{array}{cccccc} i_1 & i_2 & a & b & e & d \end{array} \\
\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\
\begin{array}{cccccc} i_1 & i_2 & a & b & e & d \end{array} \end{array}
\begin{array}{c} \\ a \\ b \\ c \\ d \\ o_1 \\ o_2 \\ \\ \end{array}
$$

[1] Mhalla et al. (2010), arXiv:1006.2616

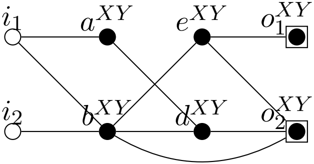
Previous flow-reversibility – XY only case[1]



$$\begin{array}{c|cccccc|c} & i_1 & i_2 & a & b & e & d & \\ \hline a & 1 & 1 & 0 & 0 & 0 & 0 & a \\ b & 0 & 1 & 0 & 0 & 0 & 0 & b \\ e & 1 & 0 & 1 & 1 & 0 & 1 & c \\ d & 0 & 0 & 1 & 0 & 0 & 0 & d \\ o_1 & 1 & 1 & 0 & 0 & 1 & 1 & o_1 \\ o_2 & 1 & 0 & 0 & 0 & 0 & 1 & o_2 \\ \hline & i_1 & i_2 & a & b & e & d & \end{array}$$

$\downarrow$ switch I and O $\downarrow$

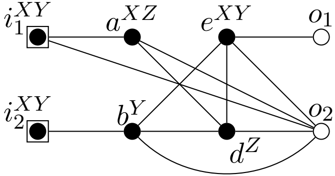
$\downarrow$ transpose $\downarrow$



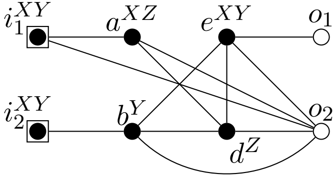
$$\begin{array}{c|cccccc|c} & a & b & e & d & o_1 & o_2 & \\ \hline i_1 & 1 & 0 & 1 & 0 & 1 & 1 & i_1 \\ i_2 & 1 & 1 & 0 & 0 & 1 & 0 & i_2 \\ a & 0 & 0 & 1 & 1 & 0 & 0 & a \\ b & 0 & 0 & 1 & 0 & 0 & 0 & b \\ e & 0 & 0 & 0 & 0 & 1 & 0 & c \\ d & 0 & 0 & 1 & 0 & 1 & 1 & d \\ \hline & a & b & e & d & o_1 & o_2 & \end{array}$$

[1] Mhalla et al. (2010), arXiv:1006.2616

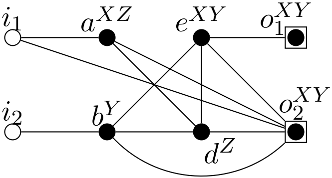
Flow-reversibility – any measurement labels



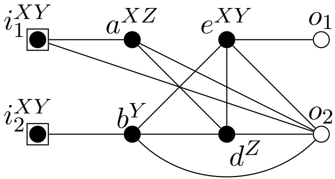
Flow-reversibility – any measurement labels



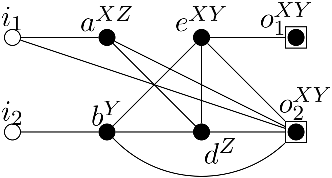
$\downarrow$ switch I and O $\downarrow$



Flow-reversibility – any measurement labels

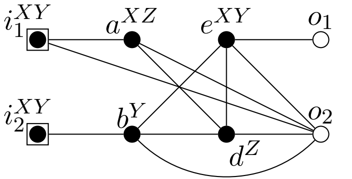


$\downarrow$ switch I and O $\downarrow$



$$\begin{array}{c} \\ a \\ b \\ e \\ d \\ o_1 \\ o_2 \\ \\ \end{array} \begin{array}{c} \begin{matrix} i_1 & i_2 & a & b & e & d \end{matrix} \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \\ \begin{matrix} i_1 & i_2 & a & b & e & d \end{matrix} \end{array} \begin{array}{c} \\ a \\ b \\ e \\ d \\ o_1 \\ o_2 \\ \\ \end{array}$$

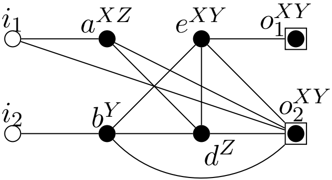
Flow-reversibility – any measurement labels



$$\begin{array}{c|cccccc|c} & i_1 & i_2 & a & b & e & d & \\ \hline a & 0 & 0 & 1 & 0 & 0 & 0 & a \\ b & 0 & 1 & 0 & 0 & 0 & 0 & b \\ e & 1 & 1 & 1 & 1 & 0 & 1 & e \\ d & 0 & 0 & 0 & 0 & 0 & 1 & d \\ o_1 & 1 & 1 & 1 & 0 & 1 & 1 & o_1 \\ o_2 & 1 & 0 & 1 & 0 & 0 & 0 & o_2 \\ \hline & i_1 & i_2 & a & b & e & d & \end{array}$$

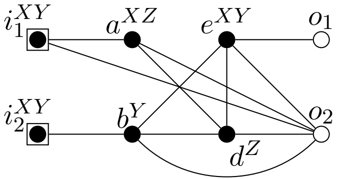
$\downarrow$ switch I and O $\downarrow$

$\downarrow$ $\downarrow$

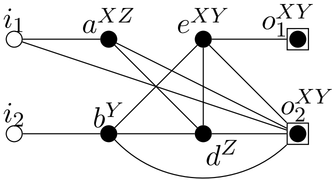


$$\begin{array}{c|cccccc|c} & a & b & e & d & o_1 & o_2 & \\ \hline i_1 & 1 & 0 & 1 & 0 & 1 & 1 & i_1 \\ i_2 & 0 & 1 & 1 & 0 & 1 & 0 & i_2 \\ a & 1 & 0 & 0 & 0 & 0 & 0 & a \\ b & 0 & 0 & 1 & 1 & 0 & 0 & b \\ e & 0 & 0 & 0 & 0 & 1 & 0 & c \\ d & 0 & 0 & 0 & 1 & 0 & 0 & d \\ \hline & a & b & e & d & o_1 & o_2 & \end{array}$$

Flow-reversibility – any measurement labels



↓ switch I and O ↓



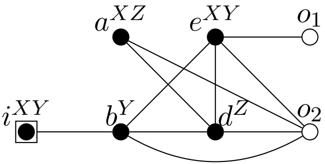
	i_1	i_2	a	b	e	d	
a	0	0	1	0	0	0	a
b	0	1	0	0	0	0	b
e	1	1	1	1	0	1	e
d	0	0	0	0	0	1	d
o_1	1	1	1	0	1	1	o_1
o_2	1	0	1	0	0	0	o_2
	i_1	i_2	a	b	e	d	

↓ not just transpose ↓

	a	b	e	d	o_1	o_2	
i_1	1	0	1	0	1	1	i_1
i_2	0	1	1	0	1	0	i_2
a	1	0	0	0	0	0	a
b	0	0	1	1	0	0	b
e	0	0	0	0	1	0	c
d	0	0	0	1	0	0	d
	a	b	e	d	o_1	o_2	

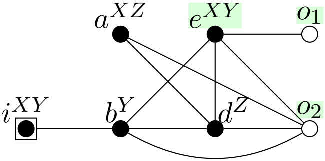
Focused sets

- Some stabilizers have trivial net effect on correction.
- Such stabilizers are determined by focused sets.
- Focused sets are parametrized by kernel of flow-demand matrix.



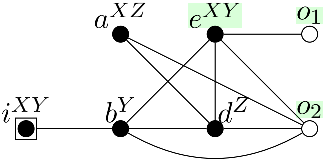
Focused sets

- Some stabilizers have trivial net effect on correction.
- Such stabilizers are determined by focused sets.
- Focused sets are parametrized by kernel of flow-demand matrix.



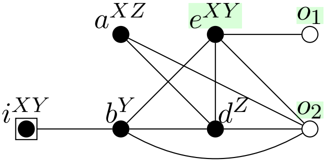
Focused sets

- Some stabilizers have trivial net effect on correction.
- Such stabilizers are determined by focused sets.
- Focused sets are parametrized by kernel of flow-demand matrix.



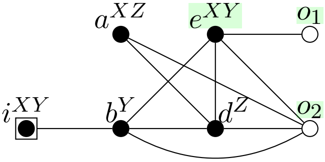
Focused sets

- Some stabilizers have trivial net effect on correction.
- Such stabilizers are determined by focused sets.
- Focused sets are parametrized by kernel of flow-demand matrix.



Focused sets

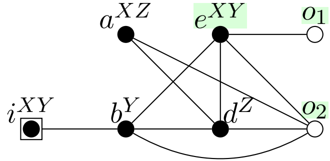
- Some stabilizers have trivial net effect on correction.
- Such stabilizers are determined by focused sets.
- Focused sets are parametrized by kernel of flow-demand matrix.



	a	b	e	d	o_1	o_2	
i	0	1	0	0	0	0	i
a	1	0	0	0	0	0	a
b	0	1	1	1	0	1	b
e	0	1	0	1	1	1	e
d	0	0	0	1	0	0	d
	a	b	e	d	o_1	o_2	

Focused sets

- Some stabilizers have trivial net effect on correction.
- Such stabilizers are determined by focused sets.
- Focused sets are parametrized by kernel of flow-demand matrix.



$$\ker \begin{array}{c} \\ i \\ a \\ b \\ e \\ d \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & e & d & o_1 & o_2 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \\ \begin{array}{cccccc} a & b & e & d & o_1 & o_2 \end{array} \end{array} \begin{array}{c} i \\ a \\ b \\ e \\ d \end{array} = \mathrm{Span} \left(\begin{array}{c} a \\ b \\ e \\ d \\ o_1 \\ o_2 \end{array} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \begin{array}{c} a \\ b \\ e \\ d \\ o_1 \\ o_2 \end{array} \right)$$

Finding flow

Flow-finding problem

Input: *a labelled open graph* $\Gamma = (G, I, O, \lambda)$*.*
Output: $(c, \prec)$ *forming Pauli flow on* Γ *or a message that no such flow exists.*

- The problem is already known to be in P,
- The existing algorithm complexity is $\mathcal{O}(n^5)$.

Flow-finding problem

Input: *a labelled open graph* $\Gamma = (G, I, O, \lambda)$.
Output: $(c, \prec)$ *forming Pauli flow on* Γ *or a message that no such flow exists.*

- The problem is already known to be in P,
- The existing algorithm complexity is $\mathcal{O}(n^5)$.

Flow-finding problem

Input: *a labelled open graph* $\Gamma = (G, I, O, \lambda)$*.*
Output: $(c, \prec)$ *forming Pauli flow on* Γ *or a message that no such flow exists.*

- The problem is already known to be in P,
- The existing algorithm complexity is $\mathcal{O}(n^5)$.

Algebraic interpretation

Theorem

Given a labelled open graph Γ, let M be its flow-demand matrix and N its order-demand matrix. Then, Γ has Pauli flow if and only if there exists a correction matrix C such that $MC = Id$ and NC forms a DAG.

- Flow-demand matrix M has shape $\bar{O} \times \bar{I}$.
- When $|I| = |O|$ then M is square.
- Hence if C such that $MC = Id$ exists, then C is unique.

Algebraic interpretation

Theorem

Given a labelled open graph Γ, let M be its flow-demand matrix and N its order-demand matrix. Then, Γ has Pauli flow if and only if there exists a correction matrix C such that $MC = Id$ and NC forms a DAG.

- Flow-demand matrix M has shape $\bar{O} \times \bar{I}$.
- When $|I| = |O|$ then M is square.
- Hence if C such that $MC = Id$ exists, then C is unique.

Algebraic interpretation

Theorem

Given a labelled open graph Γ, let M be its flow-demand matrix and N its order-demand matrix. Then, Γ has Pauli flow if and only if there exists a correction matrix C such that $MC = Id$ and NC forms a DAG.

- Flow-demand matrix M has shape $\bar{O} \times \bar{I}$.
- When $|I| = |O|$ then M is square.
- Hence if C such that $MC = Id$ exists, then C is unique.

Algebraic interpretation

Theorem

Given a labelled open graph Γ, let M be its flow-demand matrix and N its order-demand matrix. Then, Γ has Pauli flow if and only if there exists a correction matrix C such that $MC = Id$ and NC forms a DAG.

- Flow-demand matrix M has shape $\bar{O} \times \bar{I}$.
- When $|I| = |O|$ then M is square.
- Hence if C such that $MC = Id$ exists, then C is unique.

Case $|I| = |O|$

Given a labelled open graph:

1. Compute the flow-demand matrix M and the order-demand matrix N.
2. Compute a unique inverse C of M.
   - Output "No flow" if C does not exist.
3. Compute NC.
4. Check if NC is a DAG.
   - Output "No flow" if NC is not a DAG.
5. Output correction matrix C and matrix of the induced order NC.

Case $|I| = |O|$

Given a labelled open graph:

1. Compute the flow-demand matrix M and the order-demand matrix N.
2. Compute a unique inverse C of M.
    - Output "No flow" if C does not exist.
3. Compute NC.
4. Check if NC is a DAG.
    - Output "No flow" if NC is not a DAG.
5. Output correction matrix C and matrix of the induced order NC.

Case $|I| = |O|$

Given a labelled open graph:

1. Compute the flow-demand matrix M and the order-demand matrix N.
2. Compute a unique inverse C of M.
   - Output "No flow" if C does not exist.
3. Compute NC.
4. Check if NC is a DAG.
   - Output "No flow" if NC is not a DAG.
5. Output correction matrix C and matrix of the induced order NC.

Case $|I| = |O|$

Given a labelled open graph:

1. Compute the flow-demand matrix M and the order-demand matrix N.
2. Compute a unique inverse C of M.
   - Output "No flow" if C does not exist.
3. Compute NC.
4. Check if NC is a DAG.
   - Output "No flow" if NC is not a DAG.
5. Output correction matrix C and matrix of the induced order NC.

Case $|I| = |O|$

Given a labelled open graph:

1. Compute the flow-demand matrix M and the order-demand matrix N.
2. Compute a unique inverse C of M.
   - Output "No flow" if C does not exist.
3. Compute NC.
4. Check if NC is a DAG.
   - Output "No flow" if NC is not a DAG.
5. Output correction matrix C and matrix of the induced order NC.

Case $|I| = |O|$

Given a labelled open graph:

1. Compute the flow-demand matrix M and the order-demand matrix N.
2. Compute a unique inverse C of M.
   - Output "No flow" if C does not exist.
3. Compute NC.
4. Check if NC is a DAG.
   - Output "No flow" if NC is not a DAG.
5. Output correction matrix C and matrix of the induced order NC.

Time complexity is dominated by matrix inverse and matrix multiplication:

- naïve approach $\mathcal{O}(n^3)$,
- Strassen algorithm $\sim \mathcal{O}(n^{2.8})$.

Case $|I| = |O|$

Given a labelled open graph:

1. Compute the flow-demand matrix M and the order-demand matrix N.
2. Compute a unique inverse C of M.
   - Output "No flow" if C does not exist.
3. Compute NC.
4. Check if NC is a DAG.
   - Output "No flow" if NC is not a DAG.
5. Output correction matrix C and matrix of the induced order NC.

Time complexity is dominated by matrix inverse and matrix multiplication:

- naïve approach $\mathcal{O}(n^3)$,
- Strassen algorithm $\sim \mathcal{O}(n^{2.8})$.

Case $|I| = |O|$

Given a labelled open graph:

1. Compute the flow-demand matrix M and the order-demand matrix N.
2. Compute a unique inverse C of M.
   - Output “No flow” if C does not exist.
3. Compute NC.
4. Check if NC is a DAG.
   - Output “No flow” if NC is not a DAG.
5. Output correction matrix C and matrix of the induced order NC.

Time complexity is dominated by matrix inverse and matrix multiplication:

- naïve approach $\mathcal{O}(n^3)$,
- Strassen algorithm $\sim \mathcal{O}(n^{2.8})$.

Lower bound

- Let M be any $m \times m$ matrix over $\mathbb{F}_2$.
- Let $I = \{i_1, \ldots, i_m\}$, $O = \{o_1, \ldots, o_m\}$, and $V = I \cup O$.
- Let E correspond to M, i.e. $i_u o_v \in E$ if and only if $M_{u,v} = 1$.
- Let $\lambda(v) = X$ for $v \in I = \bar{O}$.
- Then the labelled open graph $((V, E), I, O, \lambda)$ has flow-demand matrix M.
- Hence it has Pauli flow if and only if M is invertible.

Lower bound

- Let M be any $m \times m$ matrix over $\mathbb{F}_2$.
- Let $I = \{i_1, \ldots, i_m\}$, $O = \{o_1, \ldots, o_m\}$, and $V = I \cup O$.
- Let E correspond to M, i.e. $i_u o_v \in E$ if and only if $M_{u,v} = 1$.
- Let $\lambda(v) = X$ for $v \in I = \bar{O}$.
- Then the labelled open graph $((V, E), I, O, \lambda)$ has flow-demand matrix M.
- Hence it has Pauli flow if and only if M is invertible.

Lower bound

- Let M be any $m \times m$ matrix over $\mathbb{F}_2$.
- Let $I = \{i_1, \ldots, i_m\}$, $O = \{o_1, \ldots, o_m\}$, and $V = I \cup O$.
- Let E correspond to M, i.e. $i_u o_v \in E$ if and only if $M_{u,v} = 1$.
- Let $\lambda(v) = X$ for $v \in I = \bar{O}$.
- Then the labelled open graph $((V, E), I, O, \lambda)$ has flow-demand matrix M.
- Hence it has Pauli flow if and only if M is invertible.

Lower bound

- Let M be any $m \times m$ matrix over $\mathbb{F}_2$.
- Let $I = \{i_1, \ldots, i_m\}$, $O = \{o_1, \ldots, o_m\}$, and $V = I \cup O$.
- Let E correspond to M, i.e. $i_u o_v \in E$ if and only if $M_{u,v} = 1$.
- Let $\lambda(v) = X$ for $v \in I = \bar{O}$.
- Then the labelled open graph $((V, E), I, O, \lambda)$ has flow-demand matrix M.
- Hence it has Pauli flow if and only if M is invertible.

Lower bound

- Let M be any $m \times m$ matrix over $\mathbb{F}_2$.
- Let $I = \{i_1, \ldots, i_m\}$, $O = \{o_1, \ldots, o_m\}$, and $V = I \cup O$.
- Let E correspond to M, i.e. $i_u o_v \in E$ if and only if $M_{u,v} = 1$.
- Let $\lambda(v) = X$ for $v \in I = \bar{O}$.
- Then the labelled open graph $((V, E), I, O, \lambda)$ has flow-demand matrix M.
- Hence it has Pauli flow if and only if M is invertible.

Lower bound

- Let M be any $m \times m$ matrix over $\mathbb{F}_2$.
- Let $I = \{i_1, \ldots, i_m\}$, $O = \{o_1, \ldots, o_m\}$, and $V = I \cup O$.
- Let E correspond to M, i.e. $i_u o_v \in E$ if and only if $M_{u,v} = 1$.
- Let $\lambda(v) = X$ for $v \in I = \bar{O}$.
- Then the labelled open graph $((V, E), I, O, \lambda)$ has flow-demand matrix M.
- Hence it has Pauli flow if and only if M is invertible.

General case

- We construct flow-demand matrix M and order-demand matrix N.
- Problem: find C such that $MC = Id$ and NC is a DAG.
- When $|I| < |O|$, there can be $2^{(n-|O|)(|O|-|I|)}$ many matrices C such that $MC = Id$.
- We cannot check NC for all such C.

General case

- We construct flow-demand matrix M and order-demand matrix N.
- Problem: find C such that $MC = Id$ and NC is a DAG.
- When $|I| < |O|$, there can be $2^{(n-|O|)(|O|-|I|)}$ many matrices C such that $MC = Id$.
- We cannot check NC for all such C.

General case

- We construct flow-demand matrix M and order-demand matrix N.
- Problem: find C such that $MC = Id$ and NC is a DAG.
- When $|I| < |O|$, there can be $2^{(n-|O|)(|O|-|I|)}$ many matrices C such that $MC = Id$.
- We cannot check NC for all such C.

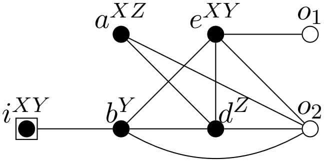
General case

- We construct flow-demand matrix M and order-demand matrix N.
- Problem: find C such that $MC = Id$ and NC is a DAG.
- When $|I| < |O|$, there can be $2^{(n-|O|)(|O|-|I|)}$ many matrices C such that $MC = Id$.
- We cannot check NC for all such C.

General case – cont.

We modify a layer-by-layer approach used in previous flow-finding algorithms:

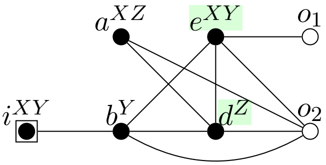
- Solve *some* linear system to find and correct vertices last in the order.
- Next, construct another linear system to find and correct vertices second to last in the order.
- And so on...



General case – cont.

We modify a layer-by-layer approach used in previous flow-finding algorithms:

- Solve *some* linear system to find and correct vertices last in the order.
- Next, construct another linear system to find and correct vertices second to last in the order.
- And so on. . .

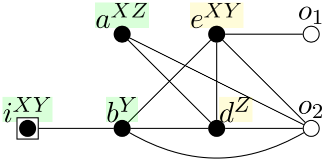


Last layer: e, d

General case – cont.

We modify a layer-by-layer approach used in previous flow-finding algorithms:

- Solve *some* linear system to find and correct vertices last in the order.
- Next, construct another linear system to find and correct vertices second to last in the order.
- And so on. . .



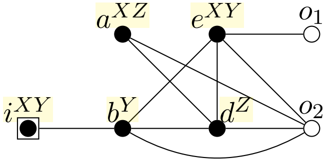
Last layer: e, d

Second to last layer: i, a, b

General case – cont.

We modify a layer-by-layer approach used in previous flow-finding algorithms:

- Solve *some* linear system to find and correct vertices last in the order.
- Next, construct another linear system to find and correct vertices second to last in the order.
- And so on...



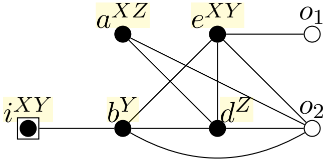
Last layer: e, d
Second to last layer: i, a, b

General case – cont.

We modify a layer-by-layer approach used in previous flow-finding algorithms:

- Solve *some* linear system to find and correct vertices last in the order.
- Next, construct another linear system to find and correct vertices second to last in the order.
- And so on...

Our improvement: instead of constructing a new system for each layer, we adjust the previous system, skipping Gaussian elimination.



Last layer: e, d
Second to last layer: i, a, b

Conclusion

Summary

- The original definition of Pauli flow is hard to work with.
- For a given labelled open graph Γ, we defined flow-demand matrix M and order-demand matrix N.
- We showed, that Γ has Pauli flow if and only if $MC = Id$ and NC is a DAG for some matrix C.
- We used this algebraic interpretation to lower the complexity bound for the flow-finding problem.

Summary

- The original definition of Pauli flow is hard to work with.
- For a given labelled open graph Γ, we defined flow-demand matrix M and order-demand matrix N.
- We showed, that Γ has Pauli flow if and only if $MC = Id$ and NC is a DAG for some matrix C.
- We used this algebraic interpretation to lower the complexity bound for the flow-finding problem.

Summary

- The original definition of Pauli flow is hard to work with.
- For a given labelled open graph Γ, we defined flow-demand matrix M and order-demand matrix N.
- We showed, that Γ has Pauli flow if and only if $MC = Id$ and NC is a DAG for some matrix C.
- We used this algebraic interpretation to lower the complexity bound for the flow-finding problem.

Summary

- The original definition of Pauli flow is hard to work with.
- For a given labelled open graph Γ, we defined flow-demand matrix M and order-demand matrix N.
- We showed, that Γ has Pauli flow if and only if $MC = Id$ and NC is a DAG for some matrix C.
- We used this algebraic interpretation to lower the complexity bound for the flow-finding problem.

Further work

- Is there an algebraic interpretation for other notions of MBQC?
- Can we find new flow-preserving rules?
- Given an unlabelled open graph, can we find interesting measurement labels resulting in Pauli flow?
- Can we speed up circuit extraction?

Further work

- Is there an algebraic interpretation for other notions of MBQC?
- Can we find new flow-preserving rules?
- Given an unlabelled open graph, can we find interesting measurement labels resulting in Pauli flow?
- Can we speed up circuit extraction?

Further work

- Is there an algebraic interpretation for other notions of MBQC?
- Can we find new flow-preserving rules?
- Given an unlabelled open graph, can we find interesting measurement labels resulting in Pauli flow?
- Can we speed up circuit extraction?

Further work

- Is there an algebraic interpretation for other notions of MBQC?
- Can we find new flow-preserving rules?
- Given an unlabelled open graph, can we find interesting measurement labels resulting in Pauli flow?
- Can we speed up circuit extraction?

arXiv:2410.23439

Thank you!