Matroids in Quantum Computing and Quantum Cryptography

Pradeep Sarvepalli

Joint work with Robert Raussendorf

Department of Physics and Astronomy
University of British Columbia, Vancouver

Applications of Matroid Theory and Combinatorial Optimization
to Information and Coding Theory

Quantum Computing and Quantum Cryptography

Why quantum computing and quantum cryptography?

- Quantum algorithms can provide speedup over classical algorithms.
  - Shor's algorithm for factoring integers is exponentially faster than any classical algorithm.
  - Grover's algorithm provides a quadratic speedup for searching.
- It might provide a means to efficiently simulate quantum systems.
- Quantum cryptography is more secure than classical cryptography.

Quantum Computing and Quantum Cryptography

Use of matroids in quantum computing and cryptography.

- Temporally unstructured quantum computation

  D. Shepherd and M. Bremner, Proc. Roy. Soc. A, 2009

- Equivalence of quantum states

  On local unitary and local Clifford orbits of stabilizer states, Preprint, 2009

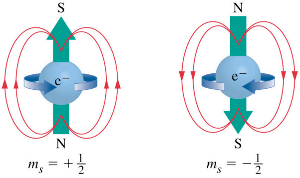
- Quantum secret sharing

  Matroids and quantum secret sharing, Preprint, 2009

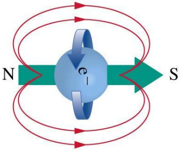
Outline

Qubits i.e. Quantum Bits

Qubits are 2-state quantum systems



Source: General Chemistry, Principles and Modern Applications



$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle$$

State space of a qubit is $\mathbb{C}^2$.

Observing qubits affects their state.

$$a\,|0\rangle + b\,|1\rangle \xrightarrow{\text{Observe}} \begin{array}{ll} |0\rangle & Pr(|0\rangle) = |a|^2 \\ |1\rangle & Pr(|1\rangle) = |b|^2 \end{array}$$

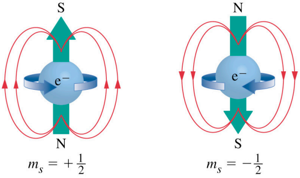
Qubits are denoted as $|0\rangle = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$ and $|1\rangle = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$

The state of n qubits is a unit vector in $\mathbb{C}^{2^n} = \underbrace{\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n}$.

$$|\psi\rangle = \sum_{x_i \in \mathbb{F}_2} \alpha_{x_1,\ldots,x_n} |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle\,; \quad \sum_{x_i \in \mathbb{F}_2} |\alpha_{x_1,\ldots,x_n}|^2 = 1.$$

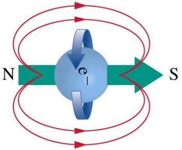
Qubits i.e. Quantum Bits

Qubits are 2-state quantum systems



Source: General Chemistry, Principles and Modern Applications

Qubits are denoted as $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$



$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle$$

State space of a qubit is $\mathbb{C}^2$.

Observing qubits affects their state.

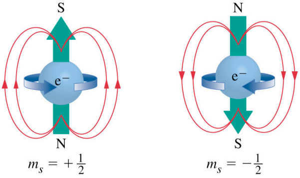
$$a\,|0\rangle + b\,|1\rangle \overset{\text{Observe}}{\longmapsto} \begin{array}{ll} |0\rangle & Pr(|0\rangle) = |a|^2 \\ |1\rangle & Pr(|1\rangle) = |b|^2 \end{array}$$

The state of n qubits is a unit vector in $\mathbb{C}^{2^n} = \underbrace{\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n}$.

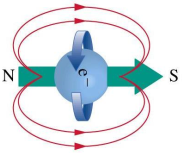
$$|\psi\rangle = \sum_{x_i \in \mathbb{F}_2} \alpha_{x_1,\ldots,x_n} |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle\,; \quad \sum_{x_i \in \mathbb{F}_2} |\alpha_{x_1,\ldots,x_n}|^2 = 1.$$

Qubits i.e. Quantum Bits

Qubits are 2-state quantum systems



Source: General Chemistry, Principles and Modern Applications



$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle$$

State space of a qubit is $\mathbb{C}^2$.

Observing qubits affects their state.

$$a\,|0\rangle + b\,|1\rangle \overset{\text{Observe}}{\longmapsto} \begin{array}{ll} |0\rangle & Pr(|0\rangle) = |a|^2 \\ |1\rangle & Pr(|1\rangle) = |b|^2 \end{array}$$

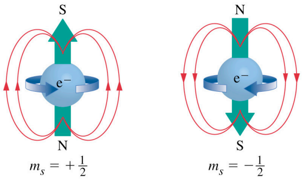
Qubits are denoted as $|0\rangle = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$ and $|1\rangle = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$

The state of n qubits is a unit vector in $\mathbb{C}^{2^n} = \underbrace{\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n}$.

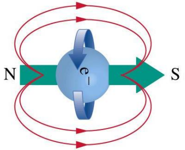
$$|\psi\rangle = \sum_{x_i \in \mathbb{F}_2} \alpha_{x_1,\ldots,x_n} |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle\,; \quad \sum_{x_i \in \mathbb{F}_2} |\alpha_{x_1,\ldots,x_n}|^2 = 1.$$

Qubits i.e. Quantum Bits

Qubits are 2-state quantum systems



Source: General Chemistry, Principles and Modern Applications



$|\psi\rangle = a\,|0\rangle + b\,|1\rangle$

State space of a qubit is $\mathbb{C}^2$.

Observing qubits affects their state.

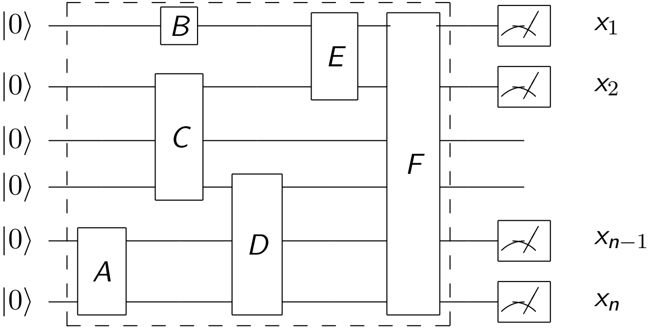
$$a\,|0\rangle + b\,|1\rangle \overset{\text{Observe}}{\longmapsto} \begin{array}{ll} |0\rangle & Pr(|0\rangle) = |a|^2 \\ |1\rangle & Pr(|1\rangle) = |b|^2 \end{array}$$

Qubits are denoted as $|0\rangle = \left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right]$ and $|1\rangle = \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]$

The state of n qubits is a unit vector in $\mathbb{C}^{2^n} = \underbrace{\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n}$.

$$|\psi\rangle = \sum_{x_i \in \mathbb{F}_2} \alpha_{x_1,\ldots,x_n} |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle\,; \quad \sum_{x_i \in \mathbb{F}_2} |\alpha_{x_1,\ldots,x_n}|^2 = 1.$$

Quantum Circuit Model



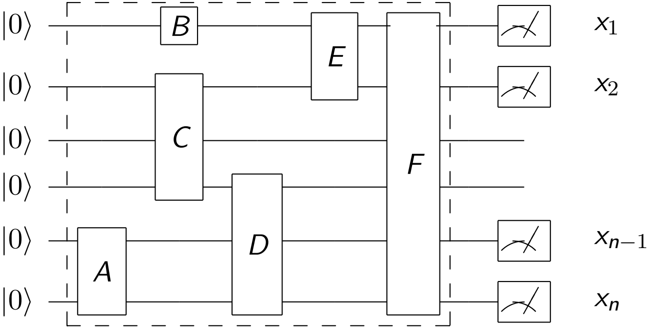
Unitary gates picture

$$|\psi(t)\rangle = U|\psi(0)\rangle$$

Hamiltonian picture

$$|\psi(t)\rangle = e^{-\frac{iHt}{\hbar}}|\psi(0)\rangle$$

Quantum Circuit Model



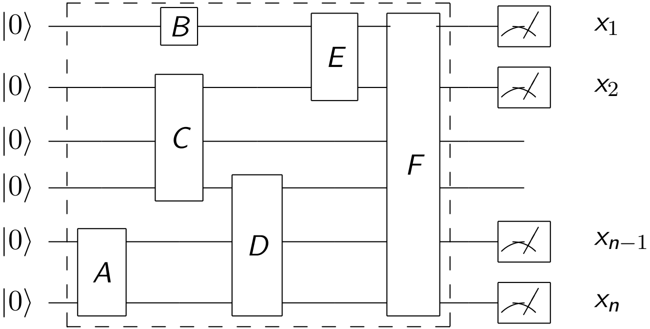
Unitary gates picture

$$|\psi(t)\rangle = U|\psi(0)\rangle$$

Hamiltonian picture

$$|\psi(t)\rangle = e^{-\frac{iHt}{\hbar}}|\psi(0)\rangle$$

Quantum Circuit Model



Unitary gates picture

$$|\psi(t)\rangle = U|\psi(0)\rangle$$

Hamiltonian picture

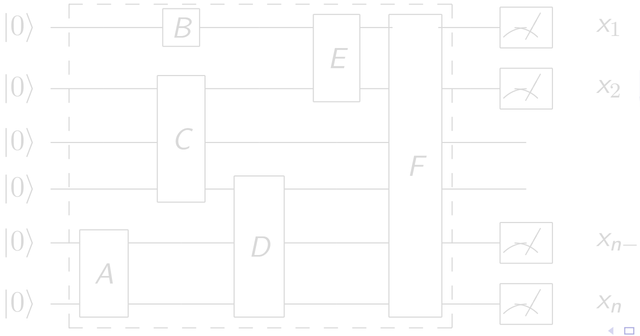
$$|\psi(t)\rangle = e^{-\frac{iHt}{\hbar}}|\psi(0)\rangle$$

Instantaneous Quantum Computation Paradigm---IQP

Temporally unstructured quantum computation, D. Shepherd and M. Bremner, Proc. Roy. Soc. A, 2009

If we understand the causes power of quantum computation we can exploit it to design new algorithms.

Studying restricted models with limited resources could help.



IQP

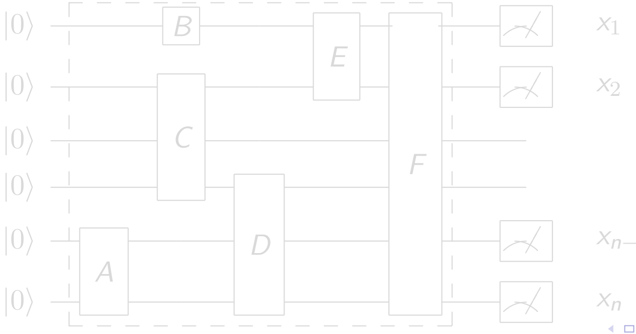
All the gates are abelian and can be implemented simultaneously.

Instantaneous Quantum Computation Paradigm---IQP

Temporally unstructured quantum computation, D. Shepherd and M. Bremner, Proc. Roy. Soc. A, 2009

If we understand the causes power of quantum computation we can exploit it to design new algorithms.

Studying restricted models with limited resources could help.



IQP

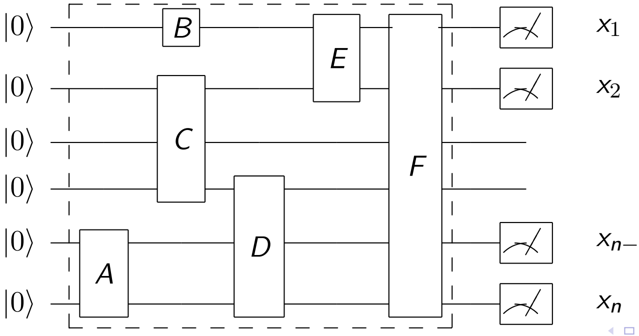
All the gates are abelian and can be implemented simultaneously.

Instantaneous Quantum Computation Paradigm---IQP

Temporally unstructured quantum computation, D. Shepherd and M. Bremner, Proc. Roy. Soc. A, 2009

If we understand the causes power of quantum computation we can exploit it to design new algorithms.

Studying restricted models with limited resources could help.



IQP

All the gates are abelian and can be implemented simultaneously.

IQP

Commuting gates $\Leftrightarrow$ Hamiltonians are additive.

We choose the IQP gates to be of the form

$$H_i = \theta_i(\otimes_{j=1}^{n} X^{p_j}) \text{ where } X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Each gate corresponds to n-bit string.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} A & C & D & E & F \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

The effective Hamiltonian is

$$H = \theta(X_1 + X_1X_2X_3 + X_3X_4 + X_1X_1 + X_1X_2X_3X_4)$$

IQP

Commuting gates ⇔ Hamiltonians are additive.

We choose the IQP gates to be of the form

$$H_i = \theta_i(\otimes_{j=1}^{n} X^{p_j}) \text{ where } X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Each gate corresponds to n-bit string.

$$\begin{array}{c|ccccc} & A & C & D & E & F \\ \hline 1 & 0 & 1 & 0 & 1 & 1 \\ 2 & 0 & 1 & 0 & 1 & 1 \\ 3 & 0 & 1 & 1 & 0 & 1 \\ 4 & 1 & 0 & 1 & 0 & 1 \end{array}$$

The effective Hamiltonian is

$$H = \theta(X_1 + X_1X_2X_3 + X_3X_4 + X_1X_1 + X_1X_2X_3X_4)$$

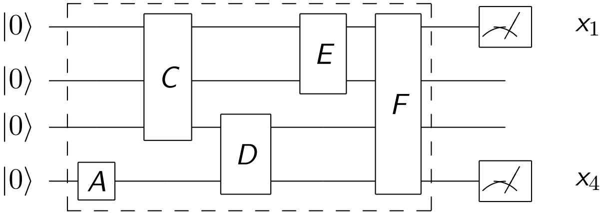
IQP

Commuting gates $\Leftrightarrow$ Hamiltonians are additive.

We choose the IQP gates to be of the form

$$H_i = \theta_i(\otimes_{j=1}^{n} X^{p_j}) \text{ where } X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Each gate corresponds to n-bit string.



$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} A & C & D & E & F \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

The effective Hamiltonian is

$$H = \theta(X_1 + X_1X_2X_3 + X_3X_4 + X_1X_1 + X_1X_2X_3X_4)$$

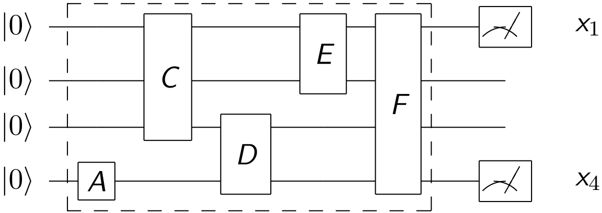
IQP

Commuting gates $\Leftrightarrow$ Hamiltonians are additive.

We choose the IQP gates to be of the form

$$H_i = \theta_i(\otimes_{j=1}^{n} X^{p_j}) \text{ where } X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Each gate corresponds to n-bit string.



$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{ccccc} A & C & D & E & F \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

The effective Hamiltonian is

$$H = \theta(X_1 + X_1X_2X_3 + X_3X_4 + X_1X_1 + X_1X_2X_3X_4)$$

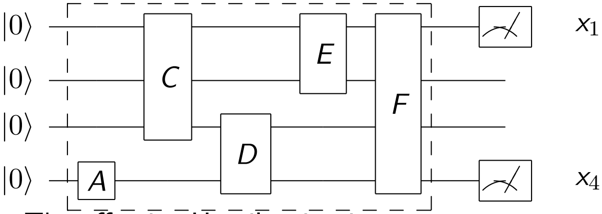
IQP

Commuting gates $\Leftrightarrow$ Hamiltonians are additive.

We choose the IQP gates to be of the form

$$H_i = \theta_i(\otimes_{j=1}^{n} X^{p_j}) \text{ where } X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Each gate corresponds to n-bit string.

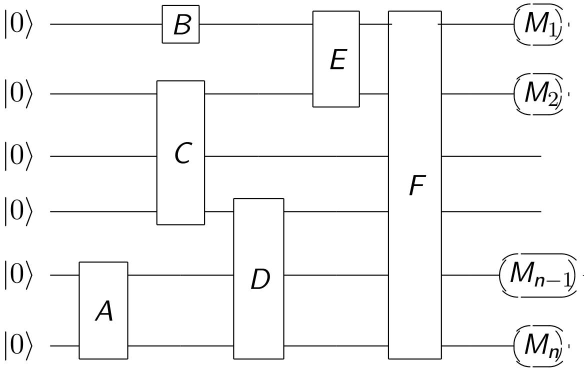


$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} A & C & D & E & F \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

The effective Hamiltonian is

$$H = \theta(X_1 + X_1X_2X_3 + X_3X_4 + X_1X_1 + X_1X_2X_3X_4)$$

Probability Distributions in IQP

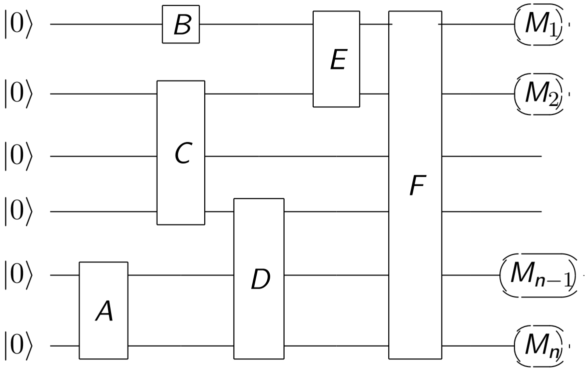


We are interested in the probability distributions at the end of the computation, i.e. $Pr(X = x)$, X is outcome of measurement.

Claim

These distributions cannot be simulated efficiently on a classical computer.

Probability Distributions in IQP

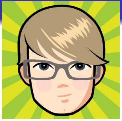


We are interested in the probability distributions at the end of the computation, i.e. $Pr(X = x)$, X is outcome of measurement.

Claim

These distributions cannot be simulated efficiently on a classical computer.

An Interactive Protocol



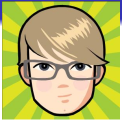
Alice chooses a matroid/code.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}$$

Alice picks a ``bias".
Alice hides the code in a larger code.

$$\left[\begin{array}{cccc|cccc} b_{11} & b_{12} & \dots & b_{1l} & a_{11} & a_{12} & \dots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1} & a_{n2} & \dots & b_{1l} & a_{n1} & a_{n2} & \dots & a_{nm} \end{array}\right]$$

An Interactive Protocol



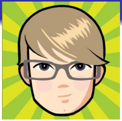
Alice chooses a matroid/code.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}$$

Alice picks a ``bias".
Alice hides the code in a larger code.

$$\left[\begin{array}{cccc|cccc} b_{11} & b_{12} & \dots & b_{1l} & a_{11} & a_{12} & \dots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1} & a_{n2} & \dots & b_{1l} & a_{n1} & a_{n2} & \dots & a_{nm} \end{array}\right]$$

An Interactive Protocol



Alice chooses a matroid/code.

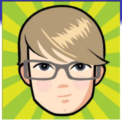
$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

Alice picks a ``bias".

Alice hides the code in a larger code.

$$\left[\begin{array}{cccc|cccc} b_{11} & b_{12} & \cdots & b_{1l} & a_{11} & a_{12} & \cdots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1} & a_{n2} & \cdots & b_{1l} & a_{n1} & a_{n2} & \cdots & a_{nm} \end{array}\right]$$

An Interactive Protocol



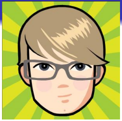
Alice chooses a matroid/code.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}$$

Alice picks a ``bias".
Alice hides the code in a larger code.

$$\left[\begin{array}{cccc|cccc} b_{11} & b_{12} & \dots & b_{1l} & a_{11} & a_{12} & \dots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1} & a_{n2} & \dots & b_{1l} & a_{n1} & a_{n2} & \dots & a_{nm} \end{array}\right]$$

An Interactive Protocol



Alice sends it to Bob

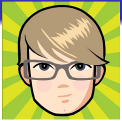
$$\left[\begin{array}{cccc|cccc} b_{11} & b_{12} & \dots & b_{1l} & a_{11} & a_{12} & \dots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1} & a_{n2} & \dots & b_{1l} & a_{n1} & a_{n2} & \dots & a_{nm} \end{array}\right]$$

Bob runs the computation induced by the code, N times.

$$O = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \vdots & \dots & \vdots & \\ x_1^{(N)} & x_2^{(N)} & \dots & x_n^{(N)} \end{bmatrix}$$

Alice tests Bob's output for bias and performs a hypothesis test if Bob really computed or cheated.

An Interactive Protocol



Alice sends it to Bob

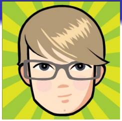
$$\left[\begin{array}{cccc|cccc} b_{11} & b_{12} & \dots & b_{1l} & a_{11} & a_{12} & \dots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1} & a_{n2} & \dots & b_{1l} & a_{n1} & a_{n2} & \dots & a_{nm} \end{array}\right]$$

Bob runs the computation induced by the code, N times.

$$O = \left[\begin{array}{cccc} x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \vdots & \dots & \vdots & \\ x_1^{(N)} & x_2^{(N)} & \dots & x_n^{(N)} \end{array}\right]$$

Alice tests Bob's output for bias and performs a hypothesis test if Bob really computed or cheated.

An Interactive Protocol



Alice sends it to Bob

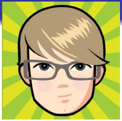
$$\left[\begin{array}{cccc|cccc} b_{11} & b_{12} & \dots & b_{1l} & a_{11} & a_{12} & \dots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1} & a_{n2} & \dots & b_{1l} & a_{n1} & a_{n2} & \dots & a_{nm} \end{array}\right]$$

Bob runs the computation induced by the code, N times.

$$O = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \vdots & \dots & \vdots & \\ x_1^{(N)} & x_2^{(N)} & \dots & x_n^{(N)} \end{bmatrix}$$

Alice tests Bob's output for bias and performs a hypothesis test if Bob really computed or cheated.

An Interactive Protocol



Alice sends it to Bob

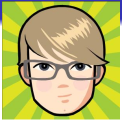
$$\left[\begin{array}{cccc|cccc} b_{11} & b_{12} & \dots & b_{1l} & a_{11} & a_{12} & \dots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1} & a_{n2} & \dots & b_{1l} & a_{n1} & a_{n2} & \dots & a_{nm} \end{array}\right]$$

Bob runs the computation induced by the code, N times.

$$O = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \vdots & \dots & \vdots & \\ x_1^{(N)} & x_2^{(N)} & \dots & x_n^{(N)} \end{bmatrix}$$

Alice tests Bob's output for bias and performs a hypothesis test if Bob really computed or cheated.

An Interactive Protocol



Alice sends it to Bob

$$\left[\begin{array}{cccc|cccc} b_{11} & b_{12} & \dots & b_{1l} & a_{11} & a_{12} & \dots & a_{1m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1} & a_{n2} & \dots & b_{1l} & a_{n1} & a_{n2} & \dots & a_{nm} \end{array}\right]$$

Bob runs the computation induced by the code, N times.

$$O = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \vdots & \dots & \vdots & \\ x_1^{(N)} & x_2^{(N)} & \dots & x_n^{(N)} \end{bmatrix}$$

Alice tests Bob's output for bias and performs a hypothesis test if Bob really computed or cheated.

Probability Distributions in IQP

The probability distributions are determined by the code.

$$Pr(X = x) = |\langle x| e^{-iHt/\hbar} |0^n\rangle|^2$$

Given a vector s, we define the bias of the distribution with respect to s as

$$\text{Bias} = Pr(X \cdot s = 0) = \sum_{x : x \cdot s = 0} |\langle x| e^{-iHt/\hbar} |0^n\rangle|^2$$

It turns out the bias is related to the evaluation of the weight enumerator of the code hidden in the larger code as long as the additional columns are orthogonal to s.

$$\text{Bias} = \mathbb{E}_{c \in C}[\cos^2(\theta(n - 2\,\text{wt}(c)))]$$

Instead of comparing the probability distributions, Alice only compares their biases.

Probability Distributions in IQP

The probability distributions are determined by the code.

$$Pr(X = x) = |\langle x| e^{-iHt/\hbar} |0^n\rangle|^2$$

Given a vector s, we define the bias of the distribution with respect to s as

$$\text{Bias} = Pr(X \cdot s = 0) = \sum_{x : x \cdot s = 0} |\langle x| e^{-iHt/\hbar} |0^n\rangle|^2$$

It turns out the bias is related to the evaluation of the weight enumerator of the code hidden in the larger code as long as the additional columns are orthogonal to s.

$$\text{Bias} = \mathbb{E}_{c \in C}[\cos^2(\theta(n - 2\,\text{wt}(c)))]$$

Instead of comparing the probability distributions, Alice only compares their biases.

Probability Distributions in IQP

The probability distributions are determined by the code.

$$Pr(X = x) = |\langle x| e^{-iHt/\hbar} |0^n\rangle|^2$$

Given a vector s, we define the bias of the distribution with respect to s as

$$\text{Bias} = Pr(X \cdot s = 0) = \sum_{x : x \cdot s = 0} |\langle x| e^{-iHt/\hbar} |0^n\rangle|^2$$

It turns out the bias is related to the evaluation of the weight enumerator of the code hidden in the larger code as long as the additional columns are orthogonal to s.

$$\text{Bias} = \mathbb{E}_{c \in C}[\cos^2(\theta(n - 2\,\text{wt}(c))]$$

Instead of comparing the probability distributions, Alice only compares their biases.

Probability Distributions in IQP

The probability distributions are determined by the code.

$$Pr(X = x) = |\langle x| e^{-iHt/\hbar} |0^n\rangle|^2$$

Given a vector s, we define the bias of the distribution with respect to s as

$$\text{Bias} = Pr(X \cdot s = 0) = \sum_{x : x \cdot s = 0} |\langle x| e^{-iHt/\hbar} |0^n\rangle|^2$$

It turns out the bias is related to the evaluation of the weight enumerator of the code hidden in the larger code as long as the additional columns are orthogonal to s.

$$\text{Bias} = \mathbb{E}_{c \in C}[\cos^2(\theta(n - 2\,\text{wt}(c))]$$

Instead of comparing the probability distributions, Alice only compares their biases.

Matroids and IQP---The Big Picture

We aim to design problems that cannot be solved efficiently on a classical computer but can simulated efficiently in the IQP model.

- ◇ A computation in IQP is induced using a code/matroid.
- ◇ ``Hide" a code A, in a larger code B.
- ◇ The ``Hidden Matroid/Code Problem": given B to extract the A with the promise A is hidden in B
- ◇ A simpler problem to extract a property of A.
- ◇ Goal is to show that the property cannot be extracted efficiently with a classical computer.
- ◇ The property we extract is essentially a ``bias" in the probability distribution of the output of the computation.

IQP---Takeaway

IQP An abelian quantum computation model with only commuting gates.

We can view the computation as being induced by a binary code or a matroid.

Open Are there interesting problems in this paradigm?

Claim The probability distributions in IQP are not efficiently simulated classically.

A two party protocol has been presented in favor. This protocol relies on the hardness of extracting the property of a hidden matroid/code.

Q1 Can this hidden matroid/code property be extracted efficiently classically?

Q2 Does the use of weighted matroids lead to computations which are hard classically?

Stabilizer States

Recall that an n-qubit state is in general given by

$$|\psi\rangle = \sum_{x_i \in \mathbb{F}_2} \alpha_{x_1,\ldots,x_n} |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle\,; \quad \sum_{x_i \in \mathbb{F}_2} |\alpha_{x_1,\ldots,x_n}|^2 = 1.$$

Pauli group

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad Y = iXZ. \quad X^2 = Z^2 = Y^2 = I$$

$$[X, Z] = [Z, Y] = [Y, X] = 0$$

$$\mathcal{P}_n = \{i^c g_1 \otimes g_2 \otimes \cdots \otimes g_n \mid g_i \in \{I, X, Y, Z\}\}$$

Stabilizer states are quantum states fixed by an abelian subgroup of $\mathcal{P}_n$.

The subgroup should not contain -I.

Stabilizer States

Recall that an n-qubit state is in general given by

$$|\psi\rangle = \sum_{x_i \in \mathbb{F}_2} \alpha_{x_1,\ldots,x_n} |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle \,; \quad \sum_{x_i \in \mathbb{F}_2} |\alpha_{x_1,\ldots,x_n}|^2 = 1.$$

Pauli group

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad Y = iXZ. \quad X^2 = Z^2 = Y^2 = I$$

$$[X, Z] = [Z, Y] = [Y, X] = 0$$

$$\mathcal{P}_n = \{ i^c g_1 \otimes g_2 \otimes \cdots \otimes g_n \mid g_i \in \{I, X, Y, Z\} \}$$

Stabilizer states are quantum states fixed by an abelian subgroup of $\mathcal{P}_n$.

The subgroup should not contain -I.

Stabilizer States

Recall that an n-qubit state is in general given by

$$|\psi\rangle = \sum_{x_i \in \mathbb{F}_2} \alpha_{x_1,\ldots,x_n} |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle\,; \quad \sum_{x_i \in \mathbb{F}_2} |\alpha_{x_1,\ldots,x_n}|^2 = 1.$$

Pauli group

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad Y = iXZ. \quad X^2 = Z^2 = Y^2 = I$$

$$[X, Z] = [Z, Y] = [Y, X] = 0$$

$$\mathcal{P}_n = \{i^c g_1 \otimes g_2 \otimes \cdots \otimes g_n \mid g_i \in \{I, X, Y, Z\}\}$$

Stabilizer states are quantum states fixed by an abelian subgroup of $\mathcal{P}_n$.

The subgroup should not contain -I.

Stabilizer States

Recall that an n-qubit state is in general given by

$$|\psi\rangle = \sum_{x_i \in \mathbb{F}_2} \alpha_{x_1,\ldots,x_n} |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle\,; \quad \sum_{x_i \in \mathbb{F}_2} |\alpha_{x_1,\ldots,x_n}|^2 = 1.$$

Pauli group

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad Y = iXZ. \quad X^2 = Z^2 = Y^2 = I$$

$$[X, Z] = [Z, Y] = [Y, X] = 0$$

$$\mathcal{P}_n = \{i^c g_1 \otimes g_2 \otimes \cdots \otimes g_n \mid g_i \in \{I, X, Y, Z\}\}$$

Stabilizer states are quantum states fixed by an abelian subgroup of $\mathcal{P}_n$.

The subgroup should not contain -I.

Entanglement

Consider the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \neq (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle)$$

- The qubits are highly correlated.

$$|00\rangle + |11\rangle \xmapsto{\text{Observe}} \begin{matrix} \nearrow & |00\rangle & Pr(|00\rangle) = 1/2 \\ \searrow & |11\rangle & Pr(|11\rangle) = 1/2 \end{matrix} \Bigg\}$$

- Observing one qubit changes the state of the other qubit instantaneously.

This phenomenon is called entanglement.

Entanglement

Consider the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \neq (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle)$$

- The qubits are highly correlated.

$$|00\rangle + |11\rangle \overset{\text{Observe}}{\longmapsto} \begin{array}{ll} \nearrow & |00\rangle \quad Pr(|00\rangle) = 1/2 \\ \searrow & |11\rangle \quad Pr(|11\rangle) = 1/2 \end{array} \Bigg\}$$

- Observing one qubit changes the state of the other qubit instantaneously.

This phenomenon is called entanglement.

Entanglement

Consider the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \neq (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle)$$

- The qubits are highly correlated.

$$|00\rangle + |11\rangle \overset{\text{Observe}}{\longmapsto} \begin{array}{ll} \nearrow & |00\rangle \quad Pr(|00\rangle) = 1/2 \\ \searrow & |11\rangle \quad Pr(|11\rangle) = 1/2 \end{array} \Bigg\}$$

- Observing one qubit changes the state of the other qubit instantaneously.

This phenomenon is called entanglement.

Entanglement

Consider the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \neq (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle)$$

- The qubits are highly correlated.

$$|00\rangle + |11\rangle \overset{\text{Observe}}{\longmapsto} \begin{array}{ccc} \nearrow & |00\rangle & Pr(|00\rangle) = 1/2 \\ \searrow & |11\rangle & Pr(|11\rangle) = 1/2 \end{array} \Bigg\}$$

- Observing one qubit changes the state of the other qubit instantaneously.

This phenomenon is called entanglement.

LU-LC Equivalence---Motivation

Entanglement is invariant under local unitary group $\mathcal{U}_n^l = U(2)^{\otimes^n} \leq U(2^n)$

Given $|\psi\rangle$ and $|\varphi\rangle$, does there exist a local unitary such that $|\varphi\rangle = U|\psi\rangle$ i.e. is $|\varphi\rangle \in LU(\psi)$?

- Clifford group: $\mathcal{K}_n$, the normalizer of $\mathcal{P}_n$

$$\mathcal{K}_n = \{U \in U(2^n) \mid U\mathcal{P}_n U^\dagger = \mathcal{P}_n\}.$$

- Local Clifford group: $\mathcal{K}_n^l = \mathcal{K}_1^{\otimes^n} \leq \mathcal{K}_n$.
- The LC equivalence of two stabilizer states can be tested efficiently.

LU-LC Equivalence---Motivation

Entanglement is invariant under local unitary group $\mathcal{U}_n^l = U(2)^{\otimes^n} \leq U(2^n)$

Given $|\psi\rangle$ and $|\varphi\rangle$, does there exist a local unitary such that $|\varphi\rangle = U|\psi\rangle$ i.e. is $|\varphi\rangle \in LU(\psi)$?

- Clifford group: $\mathcal{K}_n$, the normalizer of $\mathcal{P}_n$

$$\mathcal{K}_n = \{U \in U(2^n) \mid U\mathcal{P}_n U^\dagger = \mathcal{P}_n\}.$$

- Local Clifford group: $\mathcal{K}_n^l = \mathcal{K}_1^{\otimes^n} \leq \mathcal{K}_n$.
- The LC equivalence of two stabilizer states can be tested efficiently.

LU-LC Equivalence---Motivation

Entanglement is invariant under local unitary group $\mathcal{U}_n^l = U(2)^{\otimes^n} \leq U(2^n)$

Given $|\psi\rangle$ and $|\varphi\rangle$, does there exist a local unitary such that $|\varphi\rangle = U|\psi\rangle$ i.e. is $|\varphi\rangle \in LU(\psi)$?

- Clifford group: $\mathcal{K}_n$, the normalizer of $\mathcal{P}_n$

$$\mathcal{K}_n = \{U \in U(2^n) \mid U\mathcal{P}_n U^\dagger = \mathcal{P}_n\}.$$

- Local Clifford group: $\mathcal{K}_n^l = \mathcal{K}_1^{\otimes^n} \leq \mathcal{K}_n$.
- The LC equivalence of two stabilizer states can be tested efficiently.

LU-LC Equivalence---Motivation

Entanglement is invariant under local unitary group $\mathcal{U}_n^l = U(2)^{\otimes^n} \leq U(2^n)$

Given $|\psi\rangle$ and $|\varphi\rangle$, does there exist a local unitary such that $|\varphi\rangle = U|\psi\rangle$ i.e. is $|\varphi\rangle \in LU(\psi)$?

- Clifford group: $\mathcal{K}_n$, the normalizer of $\mathcal{P}_n$

$$\mathcal{K}_n = \{U \in U(2^n) \mid U\mathcal{P}_n U^\dagger = \mathcal{P}_n\}.$$

- Local Clifford group: $\mathcal{K}_n^l = \mathcal{K}_1^{\otimes^n} \leq \mathcal{K}_n$.
- The LC equivalence of two stabilizer states can be tested efficiently.

LU-LC Equivalence---Motivation

For many stabilizer states their orbits under local unitary (LU) group is same as under the local Clifford (LC) group.

- This motivated the conjecture that these orbits are always same for stabilizer states. [See for instance, http://www.imaph.tu-bs.de/qi/problems/28.html]
- Although this conjecture turned out to be false [Ji et al, 2008], we do not know how to characterize such states with distinct LU and LC orbits.

We are partly motivated to find the structure in such states and we focus on stabilizer states that arise from graphs.

Such states also have applications in foundations of quantum mechanics.

LU-LC Equivalence---Motivation

For many stabilizer states their orbits under local unitary (LU) group is same as under the local Clifford (LC) group.

- This motivated the conjecture that these orbits are always same for stabilizer states. [See for instance, http://www.imaph.tu-bs.de/qi/problems/28.html]
- Although this conjecture turned out to be false [Ji et al, 2008], we do not know how to characterize such states with distinct LU and LC orbits.

We are partly motivated to find the structure in such states and we focus on stabilizer states that arise from graphs.

Such states also have applications in foundations of quantum mechanics.

LU-LC Equivalence---Motivation

For many stabilizer states their orbits under local unitary (LU) group is same as under the local Clifford (LC) group.

- This motivated the conjecture that these orbits are always same for stabilizer states. [See for instance, http://www.imaph.tu-bs.de/qi/problems/28.html]
- Although this conjecture turned out to be false [Ji et al, 2008], we do not know how to characterize such states with distinct LU and LC orbits.

We are partly motivated to find the structure in such states and we focus on stabilizer states that arise from graphs.

Such states also have applications in foundations of quantum mechanics.

LU-LC Equivalence---Motivation

For many stabilizer states their orbits under local unitary (LU) group is same as under the local Clifford (LC) group.

- ◇ This motivated the conjecture that these orbits are always same for stabilizer states. [See for instance, http://www.imaph.tu-bs.de/qi/problems/28.html]
- ◇ Although this conjecture turned out to be false [Ji et al, 2008], we do not know how to characterize such states with distinct LU and LC orbits.

We are partly motivated to find the structure in such states and we focus on stabilizer states that arise from graphs.

Such states also have applications in foundations of quantum mechanics.

Stabilizer States from Graphic Matroids

Assume that we form a stabilizer state from a graphic matroid as follows: X-generators are formed from the cycle matroid of the graph

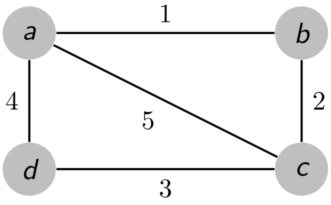
The stabilizer state is stabilized by $S = \begin{bmatrix} S_X \\ S_Z \end{bmatrix}$

$$S_Z = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} Z & Z & I & I & Z \\ I & I & Z & Z & Z \\ Z & Z & Z & Z & I \end{pmatrix} \end{array}$$

$$S_X = \begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} X & I & I & X & X \\ X & X & I & I & I \\ I & X & X & I & X \\ I & I & X & X & I \end{pmatrix} \end{array}$$

Stabilizer States from Graphic Matroids

Assume that we form a stabilizer state from a graphic matroid as follows: X-generators are formed from the cycle matroid of the graph



The stabilizer state is stabilized by $S = \begin{bmatrix} S_X \\ S_Z \end{bmatrix}$

$$S_Z = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} Z & Z & I & I & Z \\ I & I & Z & Z & Z \\ Z & Z & Z & Z & I \end{pmatrix} \end{array}$$

$$S_X = \begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{pmatrix} X & I & I & X & X \\ X & X & I & I & I \\ I & X & X & I & X \\ I & I & X & X & I \end{pmatrix} \end{array}$$

Stabilizer States from Graphs

Theorem

Let $|\psi\rangle$ be a CSS state induced by graph without loops/coloops and 2-cycles/2-cocycles. Then $LU(\psi) = LC(\psi)$.

Corollary

Given a matroid, we can efficiently test if the induced stabilizer state's LU and LC orbits are the same.

Stabilizer States from Graphs

Minimal elements are those whose support does not properly contain the support of any other element of the stabilizer

- Every generator induced by a graphic matroid and its dual is minimal.
- The stabilizer of the graphic stabilizer states is generated by its minimal elements.

Lemma (Van den Nest et al, Phys. Rev. A, 71(062323), 2005)

Let $|\psi\rangle$ be a stabilizer state with stabilizer $S(\psi)$. Let $M(\psi)$ be generated by its minimal elements. If X, Y, Z occur on every qubit of $M(\psi)$ then $LU(\psi) = LC(\psi)$.

Stabilizer States from Graphs

Minimal elements are those whose support does not properly contain the support of any other element of the stabilizer

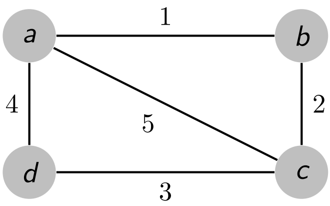
- Every generator induced by a graphic matroid and its dual is minimal.
- The stabilizer of the graphic stabilizer states is generated by its minimal elements.

Lemma (Van den Nest et al, Phys. Rev. A, 71(062323), 2005)

Let $|\psi\rangle$ be a stabilizer state with stabilizer $S(\psi)$. Let $M(\psi)$ be generated by its minimal elements. If X, Y, Z occur on every qubit of $M(\psi)$ then $LU(\psi) = LC(\psi)$.

Surface Code States

Can we go to a slightly larger class of matroids?



Qubits are on the edges.

$$A_v = \prod_{e \in \delta(v)} X_e$$

$$B_f = \prod_{e \in \partial(f)} Z_e,$$

$\delta(v) :=$ edges incident on the vertex v

$\partial(f) :=$ edges in the the boundary of the face f.

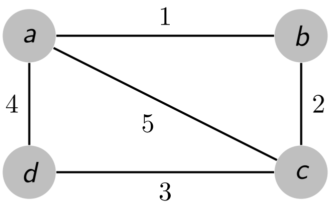
$S = \langle A_v, B_f \mid v \in V(\Gamma), f \in F(\Gamma) \rangle$

We call the states stabilized by S as the surface code states of Γ

Surface code states are CSS states.

Surface Code States

Can we go to a slightly larger class of matroids?



Qubits are on the edges.

$$A_v = \prod_{e \in \delta(v)} X_e$$

$$B_f = \prod_{e \in \partial(f)} Z_e,$$

$\delta(v) :=$ edges incident on the vertex v

$\partial(f) :=$ edges in the the boundary of the face f.

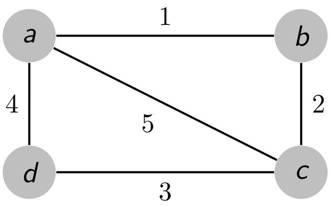
$S = \langle A_v, B_f \mid v \in V(\Gamma), f \in F(\Gamma) \rangle$

We call the states stabilized by S as the surface code states of Γ

Surface code states are CSS states.

Surface Code States

Can we go to a slightly larger class of matroids?



Qubits are on the edges.

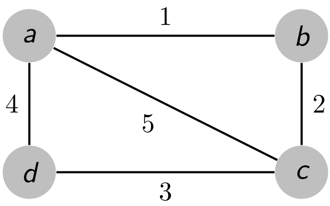
$$A_v = \prod_{e \in \delta(v)} X_e$$

$$B_f = \prod_{e \in \partial(f)} Z_e,$$

$\delta(v) :=$ edges incident on the vertex v
$\partial(f) :=$ edges in the the boundary of the face f.
$S = \langle A_v, B_f \mid v \in V(\Gamma), f \in F(\Gamma) \rangle$
We call the states stabilized by S as the surface code states of Γ
Surface code states are CSS states.

Surface Code States

Can we go to a slightly larger class of matroids?



Qubits are on the edges.

$$A_v = \prod_{e \in \delta(v)} X_e$$

$$B_f = \prod_{e \in \partial(f)} Z_e,$$

$\delta(v) :=$ edges incident on the vertex v

$\partial(f) :=$ edges in the the boundary of the face f.

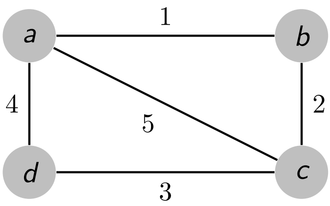
$S = \langle A_v, B_f \mid v \in V(\Gamma), f \in F(\Gamma) \rangle$

We call the states stabilized by S as the surface code states of Γ

Surface code states are CSS states.

Surface Code States

Can we go to a slightly larger class of matroids?



Qubits are on the edges.

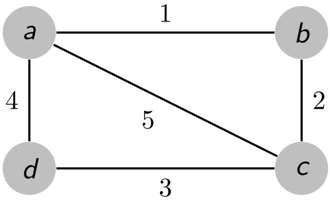
$$A_v = \prod_{e \in \delta(v)} X_e$$

$$B_f = \prod_{e \in \partial(f)} Z_e,$$

$\delta(v)$:= edges incident on the vertex v
$\partial(f)$:= edges in the the boundary of the face f.
$S = \langle A_v, B_f \mid v \in V(\Gamma), f \in F(\Gamma) \rangle$
We call the states stabilized by S as the surface code states of Γ
Surface code states are CSS states.

Surface Code States

Can we go to a slightly larger class of matroids?



Qubits are on the edges.

$$A_v = \prod_{e \in \delta(v)} X_e$$

$$B_f = \prod_{e \in \partial(f)} Z_e,$$

$\delta(v) :=$ edges incident on the vertex v
$\partial(f) :=$ edges in the the boundary of the face f.
$S = \langle A_v, B_f \mid v \in V(\Gamma), f \in F(\Gamma) \rangle$
We call the states stabilized by S as the surface code states of Γ
Surface code states are CSS states.

Dual Graphs

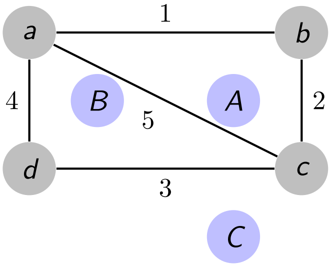
For every graph we can define a dual graph Γ^*.

- Each face becomes a node
- Two faces are conneced if they share an edge

Dual Graphs

For every graph we can define a dual graph Γ^*.

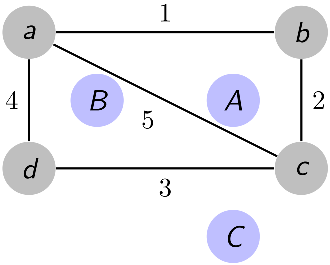
- Each face becomes a node
- Two faces are conneced if they share an edge

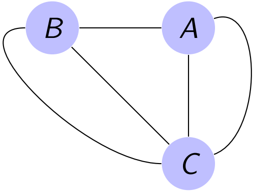


Dual Graphs

For every graph we can define a dual graph Γ^*.

- Each face becomes a node
- Two faces are conneced if they share an edge





Surface Code Matroids

The stabilizer of a surface code state is given by

- vertex operators
- face operators
- a subset of the cycles of Γ and Γ^*.

$\mathbb{I}_{V(\Gamma)}$:= vertex-edge incidence matrix of Γ
$\mathbb{I}_{C(\Gamma^*)}$:= cocycle-edge incidence matrix, where $C(\Gamma^*)$ is a subset of cycles of Γ^*.
We can write S_X i.e. the X-only operators in terms of these incidence matrices.

$$S_X = \begin{bmatrix} \mathbb{I}_{C(\Gamma^*)} \\ \mathbb{I}_{V(\Gamma)} \end{bmatrix},$$

The surface code matroid is defined as the vector matroid of S_X and we shall denote it as $\mathcal{M}(\psi_\Gamma)$.

Surface Code Matroids

The stabilizer of a surface code state is given by

- vertex operators
- face operators
- a subset of the cycles of Γ and Γ^*.

$\mathbb{I}_{V(\Gamma)}$:= vertex-edge incidence matrix of Γ
$\mathbb{I}_{C(\Gamma^*)}$:= cocycle-edge incidence matrix, where $C(\Gamma^*)$ is a subset of cycles of Γ^*.

We can write S_X i.e. the X-only operators in terms of these incidence matrices.

$$S_X = \left[\begin{array}{c} \mathbb{I}_{C(\Gamma^*)} \\ \mathbb{I}_{V(\Gamma)} \end{array} \right],$$

The surface code matroid is defined as the vector matroid of S_X and we shall denote it as $\mathcal{M}(\psi_\Gamma)$.

Surface Code Matroids

The stabilizer of a surface code state is given by

- vertex operators
- face operators
- a subset of the cycles of Γ and Γ^*.

$\mathbb{I}_{V(\Gamma)}$:= vertex-edge incidence matrix of Γ
$\mathbb{I}_{C(\Gamma^*)}$:= cocycle-edge incidence matrix, where $C(\Gamma^*)$ is a subset of cycles of Γ^*.
We can write S_X i.e. the X-only operators in terms of these incidence matrices.

$$S_X = \begin{bmatrix} \mathbb{I}_{C(\Gamma^*)} \\ \mathbb{I}_{V(\Gamma)} \end{bmatrix},$$

The surface code matroid is defined as the vector matroid of S_X and we shall denote it as $\mathcal{M}(\psi_\Gamma)$.

Surface Code Matroids

Let $|\psi_\Gamma\rangle$ be a surface code associated to a graph Γ. If Γ has no loops/coloops and 2-cycles/2-cocycles, then $\mathrm{LU}(\psi) = \mathrm{LC}(\psi)$.

Let $\mathcal{M}(\psi_\Gamma)$ be a surface code matroid with ground set $E(\Gamma)$. Then any minor of $\mathcal{M}(\psi_\Gamma)$ is also a surface code matroid. Furthermore,

a. $\mathcal{M}(\psi_\Gamma) \setminus e = \mathcal{M}(\psi'_{\Gamma\setminus e})$

b. $\mathcal{M}(\psi_\Gamma)/e = \mathcal{M}(\psi''_{\Gamma/e})$

where $\left|\psi'_{\Gamma\setminus e}\right\rangle$ and $\left|\psi''_{\Gamma/e}\right\rangle$ are some surface code states of $\Gamma \setminus e$ and Γ/e respectively.

Surface code matroids form a minor closed class of matroids.

Apparently these are called lift matroids and known already.

Surface Code Matroids

Let $|\psi_\Gamma\rangle$ be a surface code associated to a graph Γ. If Γ has no loops/coloops and 2-cycles/2-cocycles, then $\mathrm{LU}(\psi) = \mathrm{LC}(\psi)$.

Let $\mathcal{M}(\psi_\Gamma)$ be a surface code matroid with ground set $E(\Gamma)$. Then any minor of $\mathcal{M}(\psi_\Gamma)$ is also a surface code matroid. Furthermore,

a. $\mathcal{M}(\psi_\Gamma) \setminus e = \mathcal{M}(\psi'_{\Gamma\setminus e})$

b. $\mathcal{M}(\psi_\Gamma)/e = \mathcal{M}(\psi''_{\Gamma/e})$

where $\left|\psi'_{\Gamma\setminus e}\right\rangle$ and $\left|\psi''_{\Gamma/e}\right\rangle$ are some surface code states of $\Gamma \setminus e$ and Γ/e respectively.

Surface code matroids form a minor closed class of matroids.

Apparently these are called lift matroids and known already.

Surface Code Matroids

Let $|\psi_\Gamma\rangle$ be a surface code associated to a graph Γ. If Γ has no loops/coloops and 2-cycles/2-cocycles, then $\mathsf{LU}(\psi) = \mathsf{LC}(\psi)$.

Let $\mathcal{M}(\psi_\Gamma)$ be a surface code matroid with ground set $E(\Gamma)$. Then any minor of $\mathcal{M}(\psi_\Gamma)$ is also a surface code matroid. Furthermore,
a. $\mathcal{M}(\psi_\Gamma) \setminus e = \mathcal{M}(\psi'_{\Gamma\setminus e})$
b. $\mathcal{M}(\psi_\Gamma)/e = \mathcal{M}(\psi''_{\Gamma/e})$
where $\left|\psi'_{\Gamma\setminus e}\right\rangle$ and $\left|\psi''_{\Gamma/e}\right\rangle$ are some surface code states of $\Gamma \setminus e$ and Γ/e respectively.

Surface code matroids form a minor closed class of matroids.

Apparently these are called lift matroids and known already.

Surface Code Matroids

Let $|\psi_\Gamma\rangle$ be a surface code associated to a graph Γ. If Γ has no loops/coloops and 2-cycles/2-cocycles, then LU(ψ) = LC(ψ).

Let $\mathcal{M}(\psi_\Gamma)$ be a surface code matroid with ground set $E(\Gamma)$. Then any minor of $\mathcal{M}(\psi_\Gamma)$ is also a surface code matroid. Furthermore,

a. $\mathcal{M}(\psi_\Gamma) \setminus e = \mathcal{M}(\psi'_{\Gamma\setminus e})$

b. $\mathcal{M}(\psi_\Gamma)/e = \mathcal{M}(\psi''_{\Gamma/e})$

where $\left|\psi'_{\Gamma\setminus e}\right\rangle$ and $\left|\psi''_{\Gamma/e}\right\rangle$ are some surface code states of $\Gamma \setminus e$ and Γ/e respectively.

Surface code matroids form a minor closed class of matroids.

Apparently these are called lift matroids and known already.

LU-LC Equivalence---Takeaway

- Our motivation was primarily to characterize stabilizer states with distinct LU and LC orbits.
- We show that stabilizer states arising from graphs cannot have distinct orbits.
- Such stabilizer states induce graphic matroids and they can be recognized efficiently by testing if the associated matroid is graphic or cographic.

- We also identified a larger class of stabilizer states which induce matroids that are minor closed.
- Characterize the excluded minors of these matroid closed family.

LU-LC Equivalence---Takeaway

- Our motivation was primarily to characterize stabilizer states with distinct LU and LC orbits.
- We show that stabilizer states arising from graphs cannot have distinct orbits.
- Such stabilizer states induce graphic matroids and they can be recognized efficiently by testing if the associated matroid is graphic or cographic.

- We also identified a larger class of stabilizer states which induce matroids that are minor closed.
- Characterize the excluded minors of these matroid closed family.

Quantum Secret Sharing (QSS)

Classical secret to be secured

- Secret is an element of a finite alphabet (usually a finite field $\mathbb{F}_q$)
- Encoded into q orthonormal quantum states

Quantum secret to be secured (quantum state sharing)

- Secret is chosen from a set of q pure states
- Encoded into a linear combination of q orthonormal states

Why quantum secret sharing?

- Enhanced security
- Increased efficiency for classical secrets
- We might require to share a quantum state

Quantum Secret Sharing (QSS)

Classical secret to be secured

- Secret is an element of a finite alphabet (usually a finite field $\mathbb{F}_q$)
- Encoded into q orthonormal quantum states

Quantum secret to be secured (quantum state sharing)

- Secret is chosen from a set of q pure states
- Encoded into a linear combination of q orthonormal states

Why quantum secret sharing?

- ◇ Enhanced security
- ◇ Increased efficiency for classical secrets
- ◇ We might require to share a quantum state

Quantum Secret Sharing and No Cloning

Using quantum states poses new set of problems.

No Cloning Theorem (Wootters, Zurek, Dieks 1982)

We cannot make copies of an unknown quantum state.

No cloning theorem puts restrictions on the permissible authorized sets equivalently, access structures.

- ◇ No two authorized sets are disjoint. [Cleve et al, 1999]
- ◇ The adversary structure contains its dual.

$$\mathcal{A}^* \subseteq \mathcal{A} \text{ where } \mathcal{A}^* = \{A \mid \overline{A} \notin \mathcal{A}\}$$

- The access structure Γ is self-orthogonal.

$$\Gamma \subseteq \Gamma^* \text{ where } \Gamma^* = \{A \mid \overline{A} \notin \Gamma\}.$$

Previous Work on Quantum Secret Sharing

[1] Quantum secret sharing, Hillery et al, Phys. Rev. A, 59, 1829, (1999).

Introduced quantum secret sharing.

[2] How to share a quantum secret, R. Cleve et al, Phys. Rev. Lett, 83, 648, (1999).

Systematic methods for a class of quantum secret sharing schemes and connected them to quantum codes.

[3] Theory of quantum secret sharing, D. Gottesman, Phys. Rev. A, 64, 042311, (2000).

Further developed the theory addressing general access structures and classical secrets.

[4] Quantum secret sharing for general access structures, A. Smith, quant-ph/001087, (2000).

Constructions for general access structures based on monotone span programs.

[5] A Quantum Information Theoretical Model for Quantum Secret Sharing Schemes, H. Imai et al, quant-ph/0311136, (2003).

Quantum secret sharing schemes analyzed in terms of von Neumann entropy.

[6] Graph states for quantum secret sharing, M. Damian and B. Sanders, Phys. Rev. A, 78, 042309, (2008).

A framework for secret sharing using labelled graph states.

The Present Work in Context

- Previous work by Gottesman and Smith has shown how to construct quantum secret sharing schemes for general access structures.
  - Based on the ideas of monotone span programs these schemes are not always efficient.
- No associations have been made with matroids unlike the classical case.

  Classically, the most efficient secret sharing schemes have been induced by matroids.

Present work

- Characterizes quantum secret sharing schemes using matroids.
- Develops efficient quantum secret sharing schemes.

The Present Work in Context

- Previous work by Gottesman and Smith has shown how to construct quantum secret sharing schemes for general access structures.
  - Based on the ideas of monotone span programs these schemes are not always efficient.
- No associations have been made with matroids unlike the classical case.
  - Classically, the most efficient secret sharing schemes have been induced by matroids.

Present work

- Characterizes quantum secret sharing schemes using matroids.
- Develops efficient quantum secret sharing schemes.

Secret Sharing Schemes from Matroids

Given a matroid $\mathcal{M}$ we can associate a secret sharing scheme to $\mathcal{M}$. Let $V = \{1, \ldots, n, n+1\}$

- Identify $i \in V$, as the dealer
- Consider all the circuits of $\mathcal{M}$ that contain i.

$$\mathcal{C}_i = \{C \in \mathcal{C} \mid i \in C\}$$

- Consider the access structure given by

$$\Gamma_i = \{A \setminus i \mid 2^V \supseteq A \supseteq C \text{ for some } C \in \mathcal{C}_i\}. \tag{1}$$

Fact

Every matroid $\mathcal{M}(V, \mathcal{C})$ induces an access structure Γ_i as defined in (1).

Secret Sharing Schemes from Matroids

Given a matroid $\mathcal{M}$ we can associate a secret sharing scheme to $\mathcal{M}$. Let $V = \{1, \ldots, n, n+1\}$

- Identify $i \in V$, as the dealer
- Consider all the circuits of $\mathcal{M}$ that contain i.

$$\mathcal{C}_i = \{C \in \mathcal{C} \mid i \in C\}$$

- Consider the access structure given by

$$\Gamma_i = \{A \setminus i \mid 2^V \supseteq A \supseteq C \text{ for some } C \in \mathcal{C}_i\}. \tag{1}$$

Fact

Every matroid $\mathcal{M}(V, \mathcal{C})$ induces an access structure Γ_i as defined in (1).

Matroidal QSS

Fact (Cramer et al, IEEE Trans. Inform. Theory, 2008)

Let Γ_i and Γ_i^d be the access structures induced by a matroid $\mathcal{M}(V, \mathcal{C})$ and its dual matroid $\mathcal{M}^*$ by treating the ith element as the dealer. Then we have

$$\Gamma_i^d = \Gamma_i^* \qquad (2)$$

Together with the observation that the quantum access structure is self-orthogonal:

Existence of matroidal QSS (almost free)

An identically self-dual matroid $\mathcal{M}$ induces a pure state quantum secret sharing scheme.

Matroidal QSS

Fact (Cramer et al, IEEE Trans. Inform. Theory, 2008)

Let Γ_i and Γ_i^d be the access structures induced by a matroid $\mathcal{M}(V, \mathcal{C})$ and its dual matroid $\mathcal{M}^*$ by treating the ith element as the dealer. Then we have

$$\Gamma_i^d = \Gamma_i^* \tag{2}$$

Together with the observation that the quantum access structure is self-orthogonal:

Existence of matroidal QSS (almost free)

An identically self-dual matroid $\mathcal{M}$ induces a pure state quantum secret sharing scheme.

Quantum Secret Sharing Schemes from Matroids

Let $\mathcal{M}(V, \mathcal{C})$ be an identically self-dual matroid representable over $\mathbb{F}_q$ and $C \subseteq \mathbb{F}_q^{n+1}$ such that its generator matrix is a representation of $\mathcal{M}$.

$$G_C = \left[\begin{array}{c|c} 1 & g \\ \mathbf{0} & G_{\sigma_0(C)} \end{array}\right] \text{ and } G_{\rho_0(C)} = \left[\begin{array}{c} g \\ G_{\sigma_0(C)} \end{array}\right]. \qquad (3)$$

Then there exists a quantum secret sharing scheme Σ on n parties whose access structure is determined the by $\mathcal{M}$ and the dealer is associated to the first coordinate. The encoding for Σ is determined by the stabilizer code with the stabilizer matrix given by

$$S = \left[\begin{array}{c|c} G_{\sigma_0(C)} & \mathbf{0} \\ \mathbf{0} & G_{\rho_0(C)^\perp} \end{array}\right]. \qquad (4)$$

The reconstruction procedure for an authorized set A of Σ is the transformation on S such that the encoded operators for the transformed stabilizer code are X_1 and Z_1.

Encoding

$$\mathcal{E} : |s\rangle \mapsto \sum_{x \in \sigma_0(C)} |s \cdot g + \sigma_0(C)\rangle$$

For an arbitrary state we use linearity of quantum mechanics:

$$\mathcal{E} : \sum_{s \in \mathbb{F}_q} \alpha_s |s\rangle \mapsto \sum_{s \in \mathbb{F}_q} \sum_{x \in \sigma_0(C)} \alpha_s |s \cdot g + \sigma_0(C)\rangle$$

Aside: These states are precisely, the codewords of an $[[n, 1, d]]_q$ quantum code.

Encoding

$$\mathcal{E} : |s\rangle \mapsto \sum_{x \in \sigma_0(C)} |s \cdot g + \sigma_0(C)\rangle$$

For an arbitrary state we use linearity of quantum mechanics:

$$\mathcal{E} : \sum_{s \in \mathbb{F}_q} \alpha_s |s\rangle \mapsto \sum_{s \in \mathbb{F}_q} \sum_{x \in \sigma_0(C)} \alpha_s |s \cdot g + \sigma_0(C)\rangle$$

Aside: These states are precisely, the codewords of an $[[n, 1, d]]_q$ quantum code.

Encoding

$$\mathcal{E} : |s\rangle \mapsto \sum_{x\in\sigma_0(C)} |s\cdot g + \sigma_0(C)\rangle$$

For an arbitrary state we use linearity of quantum mechanics:

$$\mathcal{E} : \sum_{s\in\mathbb{F}_q} \alpha_s |s\rangle \mapsto \sum_{s\in\mathbb{F}_q} \sum_{x\in\sigma_0(C)} \alpha_s |s\cdot g + \sigma_0(C)\rangle$$

Aside: These states are precisely, the codewords of an $[[n, 1, d]]_q$ quantum code.

Recovering the secret

- We must form a linear combination of the authorized shares into one of the share.

$$\mathcal{R}: \sum_{s\in\mathbb{F}_q} \sum_{x\in\sigma_0(C)} \alpha_s |s\cdot g + x\rangle \mapsto \sum_{s\in\mathbb{F}_q} \sum_{x\in\sigma_0(C)} \alpha_s |s\rangle\,|f(s)\rangle$$

- We must also make sure the state of the authorized sets is not entangled with the rest of the system.
- We must disentangle using only the authorized sets.

Recovering the secret

- We must form a linear combination of the authorized shares into one of the share.

$$\mathcal{R} : \sum_{s\in\mathbb{F}_q} \sum_{x\in\sigma_0(C)} \alpha_s \left|s\cdot g + x\right\rangle \mapsto \sum_{s\in\mathbb{F}_q} \sum_{x\in\sigma_0(C)} \alpha_s \left|s\right\rangle \left|f(s)\right\rangle$$

- We must also make sure the state of the authorized sets is not entangled with the rest of the system.
- We must disentangle using only the authorized sets.

Recovering the secret

- We must form a linear combination of the authorized shares into one of the share.

$$\mathcal{R} : \sum_{s\in\mathbb{F}_q} \sum_{x\in\sigma_0(C)} \alpha_s |s\cdot g + x\rangle \mapsto \sum_{s\in\mathbb{F}_q} \sum_{x\in\sigma_0(C)} \alpha_s |s\rangle\, |f(s)\rangle$$

- We must also make sure the state of the authorized sets is not entangled with the rest of the system.
- We must disentangle using only the authorized sets.

Recovering the Secret

- We must disentangle using only the authorized sets.
- The key to this relies on the fact that both C and $C^{\perp}$ induce the same matroid.
- So there exists a g' whose support is entirely in $\mathrm{supp}(A)$ such that

$$\mathcal{E} : |s\rangle \mapsto \sum_{x \in \sigma_0(C)} |s \cdot g + \sigma_0(C)\rangle = \sum_{x \in \sigma_0(C)} |s \cdot g' + \sigma_0(C)\rangle$$

- To disentangle the authorized set we transform g' to $(1, 0, \ldots, 0)$

$$\mathcal{R} : \sum_{s \in \mathbb{F}_q} \sum_{x \in \sigma_0(C)} \alpha_s |s \cdot g + x\rangle \mapsto \sum_{s \in \mathbb{F}_q} \sum_{x \in \sigma_0(C)} \alpha_s |s\rangle \, |h(x)\rangle$$

Recovering the Secret

- We must disentangle using only the authorized sets.
- The key to this relies on the fact that both C and $C^{\perp}$ induce the same matroid.
- So there exists a g' whose support is entirely in $\mathrm{supp}(A)$ such that

$$\mathcal{E} : |s\rangle \mapsto \sum_{x \in \sigma_0(C)} |s \cdot g + \sigma_0(C)\rangle = \sum_{x \in \sigma_0(C)} |s \cdot g' + \sigma_0(C)\rangle$$

- To disentangle the authorized set we transform g' to $(1, 0, \ldots, 0)$

$$\mathcal{R} : \sum_{s \in \mathbb{F}_q} \sum_{x \in \sigma_0(C)} \alpha_s |s \cdot g + x\rangle \mapsto \sum_{s \in \mathbb{F}_q} \sum_{x \in \sigma_0(C)} \alpha_s |s\rangle |h(x)\rangle$$

Recovering the Secret

- We must disentangle using only the authorized sets.
- The key to this relies on the fact that both C and $C^{\perp}$ induce the same matroid.
- So there exists a g' whose support is entirely in $\mathrm{supp}(A)$ such that

$$\mathcal{E} : |s\rangle \mapsto \sum_{x\in\sigma_0(C)} |s\cdot g + \sigma_0(C)\rangle = \sum_{x\in\sigma_0(C)} \left|s\cdot g' + \sigma_0(C)\right\rangle$$

- To disentangle the authorized set we transform g' to $(1, 0, \ldots, 0)$

$$\mathcal{R} : \sum_{s\in\mathbb{F}_q} \sum_{x\in\sigma_0(C)} \alpha_s |s\cdot g + x\rangle \mapsto \sum_{s\in\mathbb{F}_q} \sum_{x\in\sigma_0(C)} \alpha_s |s\rangle\, |h(x)\rangle$$

Recovering the Secret

- We must disentangle using only the authorized sets.
- The key to this relies on the fact that both C and $C^\perp$ induce the same matroid.
- So there exists a g' whose support is entirely in $\mathrm{supp}(A)$ such that

$$\mathcal{E} : |s\rangle \mapsto \sum_{x \in \sigma_0(C)} |s \cdot g + \sigma_0(C)\rangle = \sum_{x \in \sigma_0(C)} |s \cdot g' + \sigma_0(C)\rangle$$

- To disentangle the authorized set we transform g' to $(1, 0, \ldots, 0)$

$$\mathcal{R} : \sum_{s \in \mathbb{F}_q} \sum_{x \in \sigma_0(C)} \alpha_s |s \cdot g + x\rangle \mapsto \sum_{s \in \mathbb{F}_q} \sum_{x \in \sigma_0(C)} \alpha_s |s\rangle |h(x)\rangle$$

Quantum Secret Sharing---Takeaway

Quantum secret sharing is very different from classical secret sharing

- No-cloning theorem implies there cannot be disjoint authorized sets, equivalently the access structure is self-orthogonal, [Cleve et al, 1999, Smith 2000].
- If there exists a self-orthogonal access structure then there exists a QSS, [Smith 2000, Gottesman 2000].

- We show that representable identically self-dual matroids give rise to QSS with self-dual access structures.
- These schemes have information rate one and improve upon previous schemes.

Quantum Secret Sharing---Takeaway

Quantum secret sharing is very different from classical secret sharing

- ◇ No-cloning theorem implies there cannot be disjoint authorized sets, equivalently the access structure is self-orthogonal, [Cleve et al, 1999, Smith 2000].
- ◇ If there exists a self-orthogonal access structure then there exists a QSS, [Smith 2000, Gottesman 2000].

- ◇ We show that representable identically self-dual matroids give rise to QSS with self-dual access structures.
- ◇ These schemes have information rate one and improve upon previous schemes.

Some Questions

- Considering that there exist classical secret sharing schemes that arise from matroids that are not coordinatizable, are there ideal quantum secret sharing schemes that are induced by noncoordinatizable matroids?
- Which self-dual access structures cannot be realized as ideal quantum secret schemes?
- For matroid induced classical secret sharing schemes [Beimel and Livne, 2008] showed that

$$\text{rank}(A) \leq H(A)/H(S),$$

How are the von Neumann entropy of the sets related to the rank function of the matroid, when the scheme is matroidal?