On the CNOT-complexity of CNOT-PHASE circuits

Matthew Amy [1,2] Parsiad Azimzadeh [2] Michele Mosca [1,2]

[1]Institute for Quantum Computing, Waterloo, Canada

[2]University of Waterloo, Waterloo, Canada

Theory of Quantum Computation, Communication and Cryptography
July 18th, 2018

CNOT

Sea knot???

Source: Wikipedia

CNOT/CZ optimization problems

Gate set	Complexity	State-of-the-art
CNOT	???	Asymptotically optimal synthesis[1]
CZ-PHASE	Polynomial	Optimal synthesis
CNOT-PHASE	???	Re-write rules
Clifford	???	Re-write rules
Clifford+T	???	Re-write rules

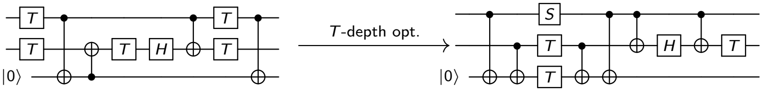
Assuming completely connected topology...

CNOT-PHASE: Circuits over CNOT and $R_Z(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i\theta} \end{pmatrix}$

[1] Patel, Markov and Hayes, *Optimal synthesis of linear reversible circuits*

Why CNOT-PHASE?

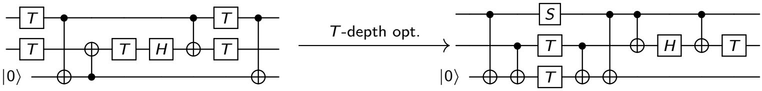
Phase folding/T-par uses T-depth optimal CNOT-PHASE synthesis as a sub-routine



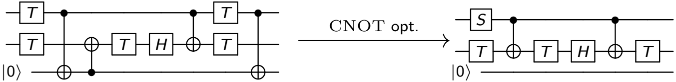
Amy, Maslov and Mosca, *Polynomial-time T-depth Optimization of Clifford+T circuits via Matroid Partitioning*

Why CNOT-PHASE?

Phase folding/T-par uses T-depth optimal CNOT-PHASE synthesis as a sub-routine



Idea: replace T-depth optimal with CNOT-optimal!



Amy, Maslov and Mosca, *Polynomial-time T-depth Optimization of Clifford+T circuits via Matroid Partitioning*

Overview

We...

- Show that in certain cases, minimizing the number of CNOT gates is equivalent to finding a minimal CNOT circuit cycling through a set of parities of the inputs
- Show that cycling through a set of parities is NP-hard if
  - all CNOT gates have the same target, or
  - the circuit inputs are not linearly independent
- Give a new heuristic optimization algorithm

The sum-over-paths form

Recall the basis state action of CNOT and Phase gates:

$$\begin{aligned} \mathrm{CNOT} :& |x\rangle|y\rangle \mapsto |x\rangle|x \oplus y\rangle \\ R_Z(\theta) :& |x\rangle \qquad \mapsto e^{2\pi i\theta x}|x\rangle \end{aligned}$$

We call this basis state action the **sum-over-paths (SOP) form**

The sum-over-paths form

Recall the basis state action of CNOT and Phase gates:

$$\begin{aligned} \mathrm{CNOT} : &|x\rangle|y\rangle \mapsto |x\rangle|x \oplus y\rangle \\ R_Z(\theta) : &|x\rangle \quad\;\; \mapsto e^{2\pi i\theta x}|x\rangle \end{aligned}$$

We call this basis state action the **sum-over-paths (SOP) form**

Definition

The SOP form of a CNOT-PHASE circuit C is a pair (f, A) where

- $f : \mathbb{F}_2^n \to \mathbb{R}$ is a pseudo-Boolean function given by

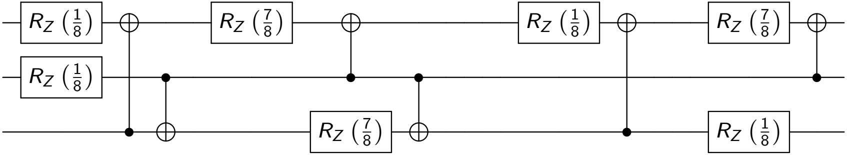
$$f(\boldsymbol{x}) = \sum_{\boldsymbol{y}\in\mathbb{F}_2^n} \widehat{f}(\boldsymbol{y})\chi_{\boldsymbol{y}}(\boldsymbol{x}), \qquad \chi_{\boldsymbol{y}}(\boldsymbol{x}) = x_1y_1 \oplus \cdots \oplus x_ny_n$$

- $A \in \mathrm{GL}(n, \mathbb{F}_2)$ is a linear permutation

such that $U_C : |\boldsymbol{x}\rangle \mapsto e^{2\pi i f(\boldsymbol{x})}|A\boldsymbol{x}\rangle$

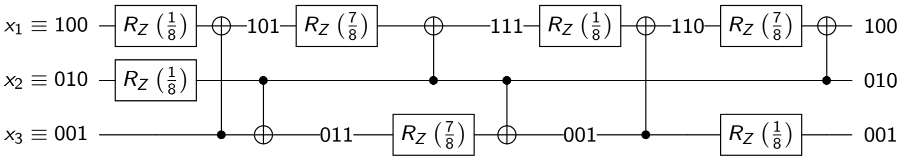
Computing the sum-over-paths

Consider an implementation of CCZ:



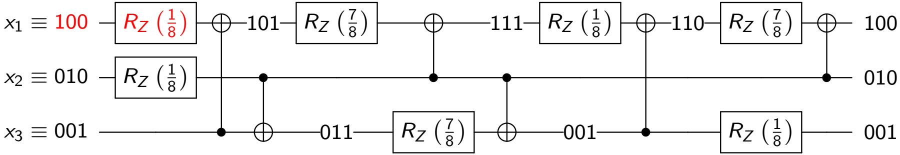
Computing the circuit sum-over-paths

First annotate...



Computing the circuit sum-over-paths

First annotate...

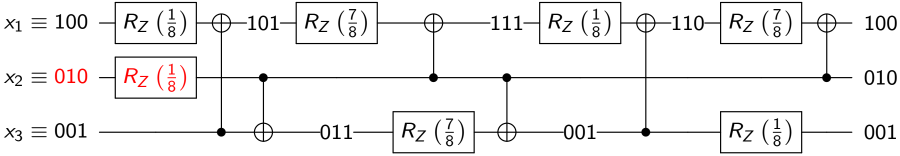


...Then add the phase factors

$$|\boldsymbol{x}\rangle \mapsto e^{\frac{2\pi i}{8}(x_1 \qquad)}|\boldsymbol{x}\rangle$$

Computing the circuit sum-over-paths

First annotate...

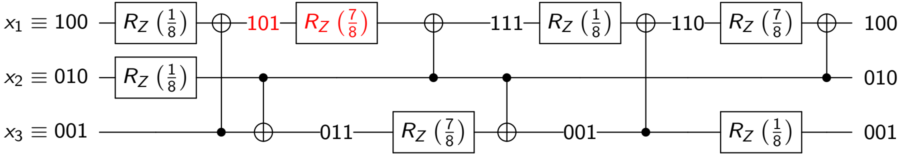


...Then add the phase factors

$$|\boldsymbol{x}\rangle \mapsto e^{\frac{2\pi i}{8}(x_1+x_2 \qquad)}|\boldsymbol{x}\rangle$$

Computing the circuit sum-over-paths

First annotate...

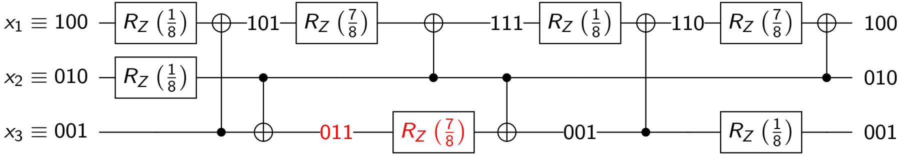


...Then add the phase factors

$$|\boldsymbol{x}\rangle \mapsto e^{\frac{2\pi i}{8}(x_1+x_2+7(x_1\oplus x_3) \qquad)}|\boldsymbol{x}\rangle$$

Computing the circuit sum-over-paths

First annotate...

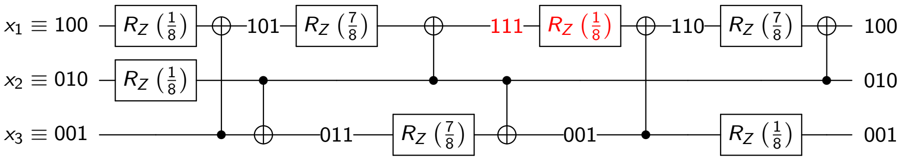


...Then add the phase factors

$$|\boldsymbol{x}\rangle \mapsto e^{\frac{2\pi i}{8}(x_1+x_2+7(x_1\oplus x_3)+7(x_2\oplus x_3)\qquad\qquad)}|\boldsymbol{x}\rangle$$

Computing the circuit sum-over-paths

First annotate...

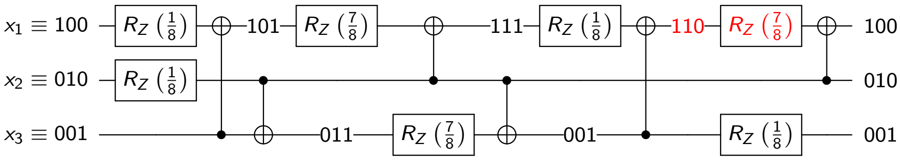


...Then add the phase factors

$$|\boldsymbol{x}\rangle \mapsto e^{\frac{2\pi i}{8}(x_1+x_2+7(x_1\oplus x_3)+7(x_2\oplus x_3)+(x_1\oplus x_2\oplus x_3) \qquad)}|\boldsymbol{x}\rangle$$

Computing the circuit sum-over-paths

First annotate...

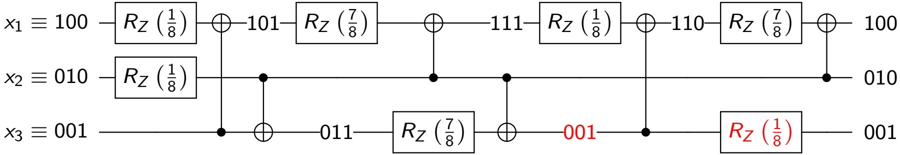


...Then add the phase factors

$$|\boldsymbol{x}\rangle \mapsto e^{\frac{2\pi i}{8}(x_1+x_2+7(x_1\oplus x_3)+7(x_2\oplus x_3)+(x_1\oplus x_2\oplus x_3)+7(x_1\oplus x_2) \qquad)}|\boldsymbol{x}\rangle$$

Computing the circuit sum-over-paths

First annotate...

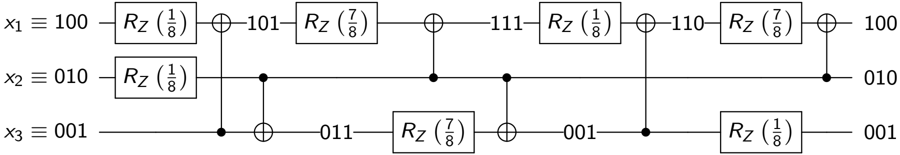


...Then add the phase factors

$$|\boldsymbol{x}\rangle \mapsto e^{\frac{2\pi i}{8}(x_1+x_2+7(x_1\oplus x_3)+7(x_2\oplus x_3)+(x_1\oplus x_2\oplus x_3)+7(x_1\oplus x_2)+x_3)}|\boldsymbol{x}\rangle$$

Computing the circuit sum-over-paths

First annotate...



...Then add the phase factors

$$|\boldsymbol{x}\rangle \mapsto e^{\frac{2\pi i}{8}(x_1+x_2+7(x_1\oplus x_3)+7(x_2\oplus x_3)+(x_1\oplus x_2\oplus x_3)+7(x_1\oplus x_2)+x_3)}|\boldsymbol{x}\rangle$$

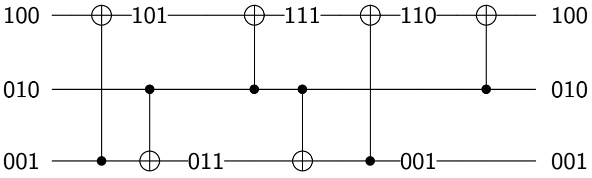
$$\mapsto e^{\frac{2\pi i}{2}x_1x_2x_3}|\boldsymbol{x}\rangle$$

An observation

Recall:

$$\begin{aligned} CS^\dagger : |x_1 x_2\rangle &\mapsto e^{\frac{2\pi i}{4} 3x_1x_2}|x_1x_2\rangle \\ &\mapsto e^{\frac{2\pi i}{8}(7x_1+7x_2+x_1\oplus x_2)}|x_1x_2\rangle \end{aligned}$$

Can use the same CNOT structure as CCZ to implement $CS^\dagger$!

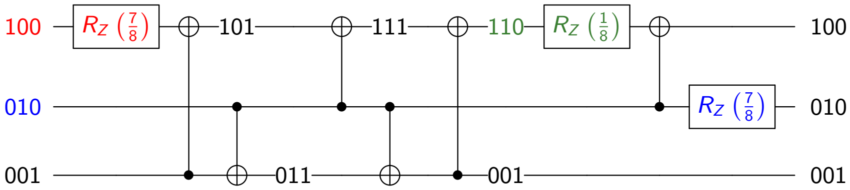


An observation

Recall:

$$CS^\dagger : |x_1 x_2\rangle \mapsto e^{\frac{2\pi i}{4} 3 x_1 x_2} |x_1 x_2\rangle$$
$$\mapsto e^{\frac{2\pi i}{8}(7x_1 + 7x_2 + x_1 \oplus x_2)} |x_1 x_2\rangle$$

Can use the same CNOT structure as CCZ to implement $CS^\dagger$!



Parity networks

Definition

A **parity network** for a set $S \subseteq \mathbb{F}_2^n$ is an n-qubit circuit C over CNOT gates where each $\boldsymbol{y} \in S$ appears in the annotated circuit.

A parity network is **pointed at** $A \in \mathrm{GL}(n, \mathbb{F}_2)$ if it implements the overall linear transformation A.

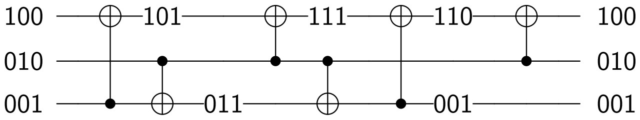
Parity networks

Definition

A **parity network** for a set $S \subseteq \mathbb{F}_2^n$ is an n-qubit circuit C over CNOT gates where each $\boldsymbol{y} \in S$ appears in the annotated circuit.

A parity network is **pointed at** $A \in \mathrm{GL}(n, \mathbb{F}_2)$ if it implements the overall linear transformation A.

E.g. the CNOT gates of CCZ,



is a parity network for $S = \{100, 010, 001, 110, 101, 011, 111\}$ pointed at $A = I$

CNOT-minimal synthesis and parity networks

A CNOT-minimal circuit with SOP form (f, A) **necessarily** gives a minimal parity network for $\mathrm{supp}(\widehat{f})$ pointed at A

CNOT-minimal synthesis and parity networks

However…

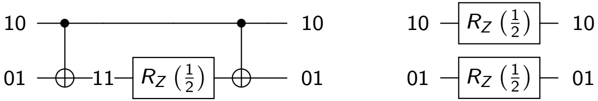
CNOT-minimal synthesis and parity networks

A minimal parity network for $\text{supp}(\widehat{f})$ may not give a CNOT-minimal circuit **across equivalent SOP forms**

CNOT-minimal synthesis and parity networks

A minimal parity network for $\mathrm{supp}(\widehat{f})$ may not give a CNOT-minimal circuit **across equivalent SOP forms**

E.g., $(\frac{1}{2}(x_1 \oplus x_2), I)$ and $(\frac{1}{2}x_1 + \frac{1}{2}x_2, I)$ give equivalent unitaries but have minimal parity network implementations



Main result

Theorem

CNOT *minimization of* CNOT-PHASE *circuits is at least as hard as synthesizing a minimal parity network*

Main result

Theorem

CNOT *minimization of* CNOT-PHASE *circuits is at least as hard as synthesizing a minimal parity network*

Intuition:

- If $(f, A) \sim (f', A')$, then $A = A'$ and $f' = f + k$ for $k : \mathbb{F}_2^n \to \mathbb{Z}$

Main result

Theorem

CNOT *minimization of* CNOT-PHASE *circuits is at least as hard as synthesizing a minimal parity network*

Intuition:

- If $(f, A) \sim (f', A')$, then $A = A'$ and $f' = f + k$ for $k : \mathbb{F}_2^n \to \mathbb{Z}$
- The Fourier coefficients of k have even order in $\mathbb{R}/\mathbb{Z}$

Main result

Theorem

CNOT *minimization of* CNOT-PHASE *circuits is at least as hard as synthesizing a minimal parity network*

Intuition:

- If $(f, A) \sim (f', A')$, then $A = A'$ and $f' = f + k$ for $k : \mathbb{F}_2^n \to \mathbb{Z}$
- The Fourier coefficients of k have even order in $\mathbb{R}/\mathbb{Z}$
- If no elements of $\widehat{f}$ have even order in $\mathbb{R}/\mathbb{Z}$, then

$$\mathrm{supp}(\widehat{f'}) \subseteq \mathrm{supp}(\widehat{f})$$

Minimal parity network synthesis is hard...?

Goal:

Prove that the minimal parity network problem (MPNP) is NP-hard

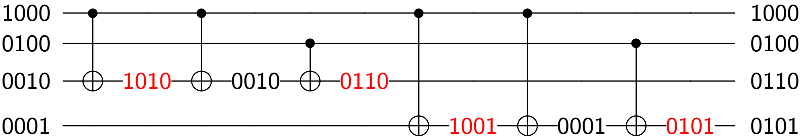
Obvious reductions don't work due to **shortcuts**

Minimal parity network synthesis is hard...?

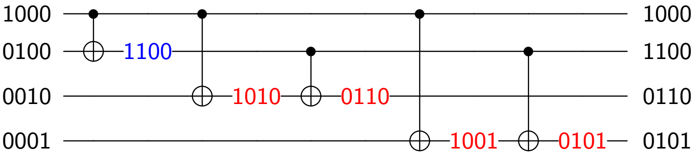
Goal:

> *Prove that the minimal parity network problem (MPNP) is NP-hard*

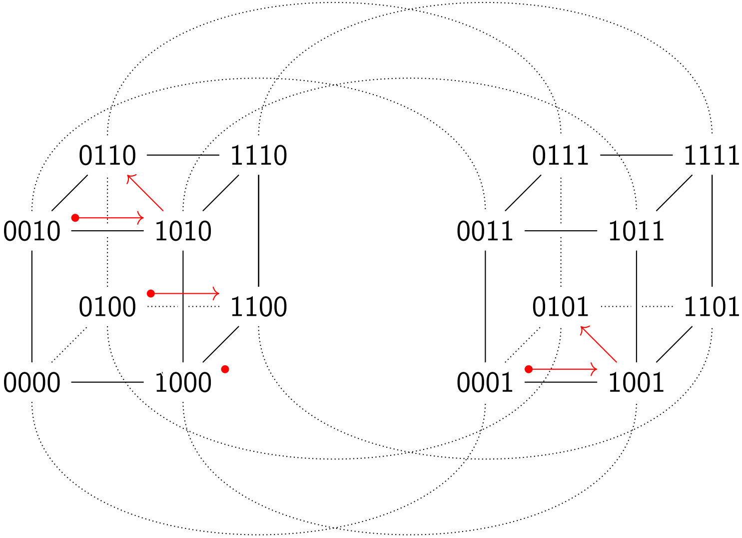
Obvious reductions don't work due to **shortcuts**



vs.



A graphical interpretation



Fixed-target minimal parity network

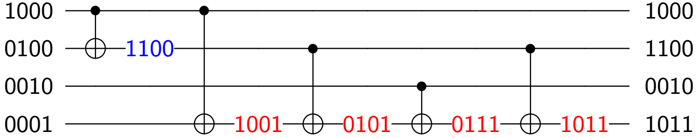
Conjecture

If for all $\boldsymbol{y} \in S$, $y_i = 1$, *then there exists a minimal parity network for* S *where each* CNOT *targets bit* i.

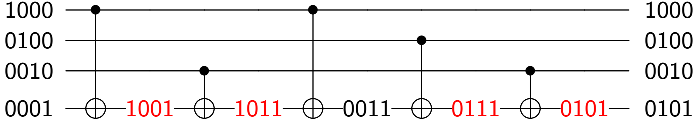
Fixed-target minimal parity network

Conjecture

If for all $\mathbf{y} \in S$, $y_i = 1$, then there exists a minimal parity network for S where each CNOT targets bit i.



vs.



Fixed-target minimal parity network

Theorem

The fixed-target minimal parity network problem is NP-complete

Proof:

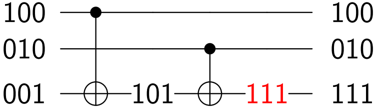
Reduction from traveling salesman on the hypercube [2]

[2] Ernvall, Katajainen, and Penttonen, *NP-completeness of the Hamming salesman problem*

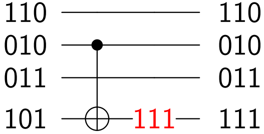
Minimal parity network with encoded inputs

If some inputs are linearly dependent, fewer gates may be needed to implement a parity network

E.g., $S = \{111\}$



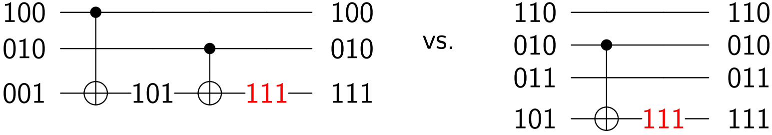
vs.



Minimal parity network with encoded inputs

If some inputs are linearly dependent, fewer gates may be needed to implement a parity network

E.g., $S = \{111\}$



Direct applications to phase folding with ancillas!

Minimal parity network with encoded inputs

Theorem

The encoded input minimal parity network problem is NP-complete

Proof:

Reduction from maximum-likelihood decoding[3]

[3] Berlekamp, McEliece, and van Tilborg, *On the inherent intractability of certain coding problems*

Problem statement

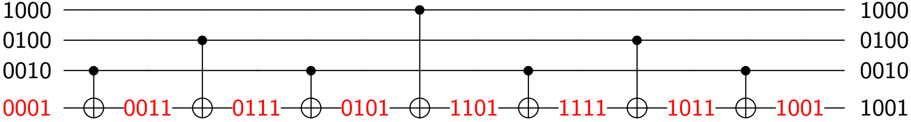
Given $S \subseteq \mathbb{F}_2^n$, synthesize an efficient parity network for S

Bases cases

For $S = \mathbb{F}_2^n \parallel \boldsymbol{x}$, $\boldsymbol{x} \in \mathbb{F}_2^m$, minimal parity network is the Gray code and can be computed greedily

E.g.,

$$S = \mathbb{F}_2^3 \parallel 1 = \{0001, 1001, 0101, 1101, 0011, 1011, 0111, 1111\}$$

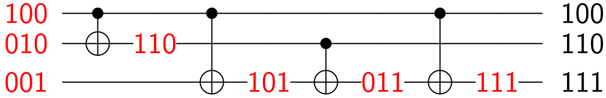


Bases cases

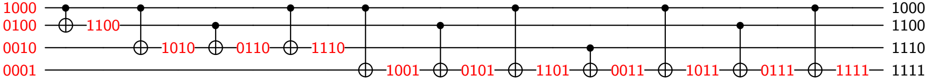
For $S = \mathbb{F}_2^n$, case is similar

E.g.,

$S = \mathbb{F}_2^3$



$S = \mathbb{F}_2^4$



The GRAY-SYNTH algorithm

Main idea:

> *Try to identify subsets S' of S which have the form $S' \simeq \mathbb{F}_2^n \,\|\, \boldsymbol{x}$, and synthesize those greedily*

The GRAY-SYNTH algorithm

1. Start with a singleton stack containing the set S
2. Pop a set S' off the stack
3. If $x_i \oplus x_j$ appears in every parity of S',
   - Apply a CNOT between bits i and j, and
   - Adjust all subsets remaining on the stack accordingly
4. Pick some row i maximizing the number of parities in S' which **either contain or do not contain** x_i
5. Set $S_b = \{\boldsymbol{x} \in S' \mid x_i = b\}$ and push S_1, S_0 onto the stack
6. Go to step 2

Invariant: remaining parities are expressed over the current basis

- Avoids "uncomputing" or backtracking

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$

$$\left\{ \boxed{\begin{matrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{matrix}} \right\}$$

$$\left\{ \begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{matrix} \right\}$$

$$\left\{ \begin{matrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{matrix} \right.$$

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$

$$\left\{ \begin{array}{c|ccccc} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{array} \right\}$$

- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$

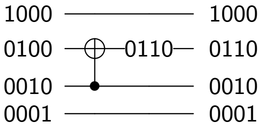
$$\left\{ \begin{array}{c|ccccc} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{array} \right\}$$

- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$

$$\left\{\begin{array}{c|ccccc} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{array}\right\} \rightarrow \left\{\begin{array}{c|ccccc} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ \color{red}{0} & 0 & 0 & \color{red}{0} & \color{red}{1} & \color{red}{1} \\ 0 & 0 & 1 & 0 & 1 & 0 \end{array}\right\}$$

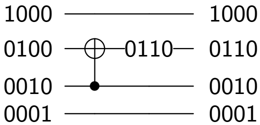


- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$

$$\left\{ \begin{array}{|c|cccccc} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{array} \right\}$$

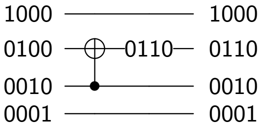


- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$

$$\left\{\begin{array}{cc|ccc} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{array}\right\}$$

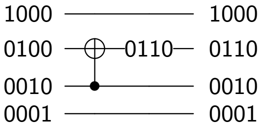


- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$

$$\left\{\begin{array}{cc|ccc} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{array}\right\}$$

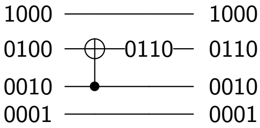


- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$

$$\left\{ \begin{array}{c|cccc} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{array} \right\}$$

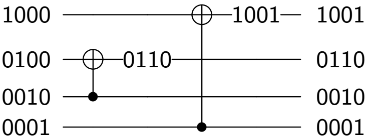


- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$

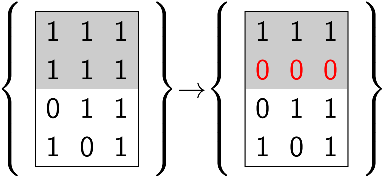
$$\left\{ \begin{array}{c|ccc} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{array} \right\} \rightarrow \left\{ \begin{array}{c|ccc} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ \color{red}{0} & \color{red}{1} & \color{red}{0} & \color{red}{1} \end{array} \right\}$$

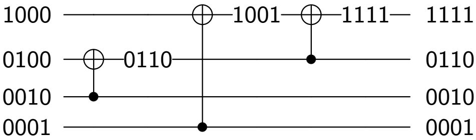


- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$



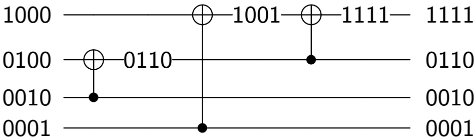


- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$

$$\left\{ \begin{array}{|c|cc} \hline 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ \hline \end{array} \right\}$$

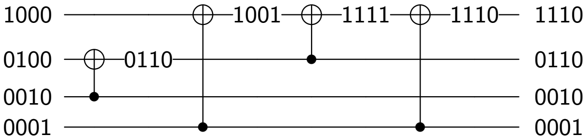


- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$

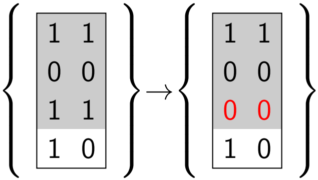
$$\left\{ \begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{matrix} \right\} \rightarrow \left\{ \begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{matrix} \right\}$$

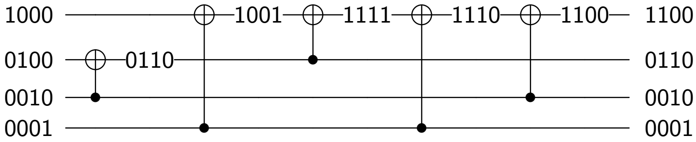


- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$



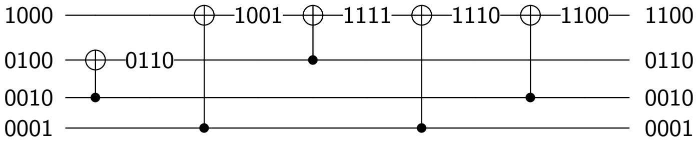


- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Example

Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$

$$\left\{ \begin{array}{c|c} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{array} \right\}$$

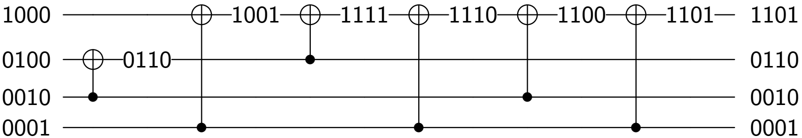


- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Example

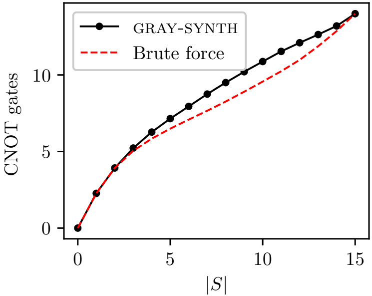
Parity network for $S = \{0110, 1000, 1001, 1110, 1101, 1100\}$

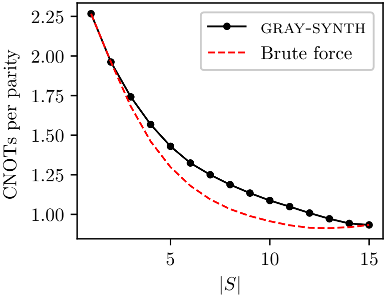
$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\} \to \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$



- Columns are remaining parities
- Box is current top of the stack
- White rows haven't been partitioned
- Grey rows have been partitioned

Performance vs. brute force





- Data collected across all sets of parities on 4 bits
- GRAY-SYNTH within 15% of optimal on average for $|S| = 8$

---

https://github.com/meamy/feynman

Benchmarks

Benchmark	n	Base	Nam *et al.* (L)		T-par (GRAY-SYNTH)		
		CNOT	Time	CNOT	Time	CNOT	% Red.
Grover_5	9	**336**	–	–	0.027	**210**	37.5
Mod 5_4	5	**32**	< 0.001	**28**	0.001	**26**	18.8
VBE-Adder_3	10	**80**	< 0.001	**50**	0.004	**42**	47.5
CSLA-MUX_3	15	**90**	< 0.001	**76**	0.073	**100**	-11.1
CSUM-MUX_9	30	**196**	< 0.001	**168**	0.095	**148**	24.5
QCLA-Com_7	24	**215**	0.001	**132**	0.097	**136**	36.7
QCLA-Mod_7	26	**441**	0.004	**302**	0.145	**356**	19.3
QCLA-Adder_10	36	**267**	0.002	**195**	0.112	**189**	29.2
Adder_8	24	**466**	0.004	**331**	0.165	**352**	24.5
RC-Adder_6	14	**104**	< 0.001	**73**	0.080	**71**	31.7
Mod-Red_21	11	**122**	< 0.001	**81**	0.091	**84**	31.1
Mod-Mult_55	9	**55**	< 0.001	**40**	0.004	**45**	18.2
Mod-Adder_1024	28	**2005**	–	–	0.739	**1376**	31.4
Cycle 17_3	35	**4532**	–	–	2.618	**2998**	36.8
GF(2^{32})-Mult	96	**7292**	1.834	**6299**	5.571	**6658**	8.7
GF(2^{64})-Mult	192	**28861**	58.341	**24765**	114.310	**25966**	10.0
Ham_15 (low)	17	**259**	–	–	0.043	**208**	19.7
Ham_15 (med)	17	**574**	–	–	0.089	**351**	43.0
Ham_15 (high)	20	**2489**	–	–	0.376	**1500**	40.0
HWB_6	7	**131**	–	–	0.006	**111**	15.3
HWB_8	12	**7508**	–	–	1.706	**6719**	10.5
QFT_4	5	**48**	–	–	0.005	**47**	2.1
$\Lambda_5(X)$	9	**49**	< 0.001	**30**	0.003	**30**	38.8
$\Lambda_5(X)$ (Barenco)	9	**84**	< 0.001	**60**	0.004	**54**	35.7
$\Lambda_{10}(X)$	19	**119**	< 0.001	**70**	0.071	**70**	41.2
$\Lambda_{10}(X)$ (Barenco)	19	**224**	0.001	**160**	0.029	**144**	35.7
Total							23.3

https://github.com/meamy/feynman

Conclusion

In this talk...

- Parity networks characterize the CNOT complexity of CNOT-PHASE circuits **for a particular phase function**
- CNOT minimization is at least as hard as synthesizing a minimal parity network
- Synthesizing a minimal parity network is NP-hard when targets are fixed or inputs are encoded
- A heuristic parity network synthesis algorithm & benchmarks

Future work

- Proof of hardness for the general problem
- Synthesis algorithm that combines parity network synthesis with an output linear permutation
- Adding topology constraints

Thank you!