Circuit Extraction for ZX-diagrams can be #**P**-hard

Niel de Beaudrap — University of Sussex
Aleks Kissinger — University of Oxford
John van de Wetering — Radboud University & Oxford

QPL – June 27, 2022

The circuit extraction problem

Input: A ZX-diagram with promise it is unitary.
Output: Description of quantum circuit of same unitary.

Theorem

The circuit extraction problem is **#P**-hard.

The result in brief

The circuit extraction problem

Input: A ZX-diagram with promise it is unitary.
Output: Description of quantum circuit of same unitary.

Theorem

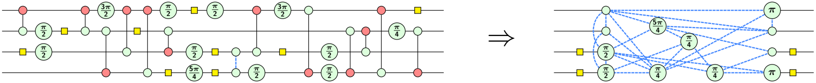
The circuit extraction problem is **#P**-hard.

Note: Strong quantum circ simulation is **#P**-complete
$\rightarrow$ circ extraction at least as hard as computing properties of diagrams directly.

Why do we need circuit extraction?

Suppose we want to use ZX for circuit optimisation.

- First write circuit as ZX-diagram.
- Simplify diagram with favourite strategy.

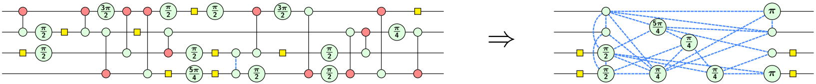


- Now need to transform diagram back into circuit.

Why do we need circuit extraction?

Suppose we want to use ZX for circuit optimisation.

- First write circuit as ZX-diagram.
- Simplify diagram with favourite strategy.



- Now need to transform diagram back into circuit.

More generally: if you want to run a ZX-diagram on a quantum computer, it needs to be a circuit.

Known extraction techniques

- *Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus*
  When diagram has *gflow*, can extract circuit.
- *There and back again: A circuit extraction tale*
  When diagram has *extended gflow*, can extract circuit.
- *Relating Measurement Patterns to Circuits via Pauli Flow*
  When diagram has *Pauli flow*, can extract circuit.

Known extraction techniques

- *Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus*
  When diagram has *gflow*, can extract circuit.
- *There and back again: A circuit extraction tale*
  When diagram has *extended gflow*, can extract circuit.
- *Relating Measurement Patterns to Circuits via Pauli Flow*
  When diagram has *Pauli flow*, can extract circuit.

They all require promise on 'local structure' of the diagram.
This prevents the diagrams from becoming 'too wild'.

Formal definition

CircuitExtraction
Input: A ZX-diagram D with n inputs and outputs and at most k wires and/or spiders, and a set $\mathcal{G}$ of unitary gates (each acting on at most $O(1)$ qubits).

Formal definition

CircuitExtraction
Input: A ZX-diagram D with n inputs and outputs and at most k wires and/or spiders, and a set $\mathcal{G}$ of unitary gates (each acting on at most $O(1)$ qubits).

Promise: D is proportional to a unitary.

Formal definition

CircuitExtraction
Input: A ZX-diagram D with n inputs and outputs and at most k wires and/or spiders, and a set $\mathcal{G}$ of unitary gates (each acting on at most $O(1)$ qubits).

Promise: D is proportional to a unitary.

Output: Either
(a) a poly(n, k)-size circuit C, given as a sequence of gates from $\mathcal{G}$ and expressing an n-qubit unitary that is proportional to D, if such a circuit exists;
or **(b)** a message no such circuit exists, if that is the case.

Some complexity theory

- An **NP**-complete problem is **SAT**, where we ask whether a Boolean formula has a satisfying assignment.

Some complexity theory

- An **NP**-complete problem is **SAT**, where we ask whether a Boolean formula has a satisfying assignment.
- **#P** is the 'counting version' of **NP**: I.e., output not whether there is *a* solution, but *how many* solutions there are.
- A **#P**-complete problems is **#SAT**, where we ask for the number of solutions of a Boolean formula.

Some complexity theory

- An **NP**-complete problem is **SAT**, where we ask whether a Boolean formula has a satisfying assignment.
- **#P** is the 'counting version' of **NP**: I.e., output not whether there is *a* solution, but *how many* solutions there are.
- A **#P**-complete problems is **#SAT**, where we ask for the number of solutions of a Boolean formula.

Toda's theorem

The polynomial hierarchy is contained in $\mathbf{P}^{\#\mathbf{P}}$.

Main result

Theorem

#SAT Cook reduces to **CircuitExtraction**:
$\mathbf{\#SAT} \in \mathbf{FP}^{\mathbf{CircuitExtraction}}$.

Corollary

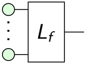
If there is a poly-time algorithm for **CircuitExtraction**, then the entire polynomial hierarchy collapses, and in particular $\mathbf{P} = \mathbf{NP}$.

Sketch of proof

- Suppose given Boolean formula $f : \{0,1\}^n \to \{0,1\}$.
- Construct ZX-diagram for linear map L_f which acts as $L_f|\vec{x}\rangle = |f(\vec{x})\rangle$:

Sketch of proof

- Suppose given Boolean formula $f : \{0,1\}^n \to \{0,1\}$.
- Construct ZX-diagram for linear map L_f which acts as $L_f|\vec{x}\rangle = |f(\vec{x})\rangle$:
- Now note:

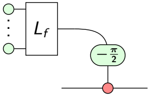


$$= \sum_x L_x|x\rangle = \sum_x |f(x)\rangle = \frac{N_0}{2^n}|0\rangle + \frac{N_1}{2^n}|1\rangle =: a_0|0\rangle + a_1|1\rangle$$

- Up to normalisation this state is $Y_\alpha|0\rangle = \cos(\frac{\alpha}{2})|0\rangle + \sin(\frac{\alpha}{2})|1\rangle$ where α is determined by number of solutions to f.

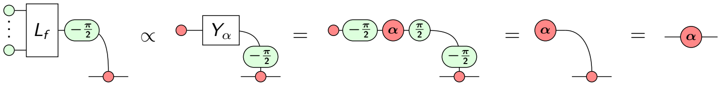
Sketch of Proof, part 2

- We use this state we prepared as the input to a controlled operation:



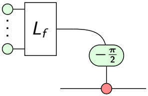
(1)

- This actually implements an X rotation:



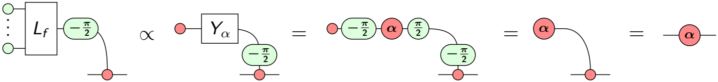
Sketch of Proof, part 2

- We use this state we prepared as the input to a controlled operation:



(1)

- This actually implements an X rotation:



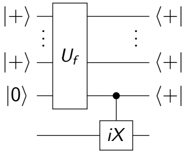
- So if we feed diagram (1) to **CircuitExtraction**, we get a 1-qubit circ equal to $X(\alpha)$.
- Multiply out the gates in the circ to actually calculate α. QED.

Why it works

- While the diagram contains poly(n) spiders, the circuit has exactly 1 qubit.
- Furthermore, circ only has poly(n) gates.
- So can multiply out all the gates in poly-time.
- Only need to know value of α up to poly(n) bits of precision.

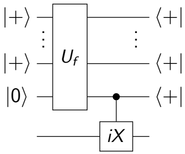
Relation to post-selection

Can write the diagram as post-selected circuit:



Relation to post-selection

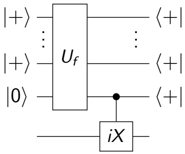
Can write the diagram as post-selected circuit:



- Note: 'correct' postselection prob is $\geq \frac{1}{4}$ independent of n.
- So: could implement the circ with high prob on real QC.

Relation to post-selection

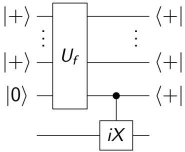
Can write the diagram as post-selected circuit:



- Note: ‘correct’ postselection prob is $\geq \frac{1}{4}$ independent of n.
- So: could implement the circ with high prob on real QC.
- However: adjacent possible values of α are exponentially close.
- So running on QC won’t work: need exponential samples.

Relation to post-selection

Can write the diagram as post-selected circuit:



- Note: 'correct' postselection prob is $\geq \frac{1}{4}$ independent of n.
- So: could implement the circ with high prob on real QC.
- However: adjacent possible values of α are exponentially close.
- So running on QC won't work: need exponential samples.

Corollary

Removing post-selections from post-selected circuit which is unitary, is #**P**-hard.

Circuit Extraction with auxiliary qubits

AuxCircuitExtraction
Input: Unitary ZX-diagram.
Output: Circuit with up to logarithmic number of ancillae, measurements and classical corrections.

Theorem
AuxCircuitExtraction is #**P**-hard.

Approximate Circuit Extraction

ApproxCircuitExtraction
Input: Unitary ZX-diagram D and a precision parameter $\varepsilon > 0$.
Output: A poly($n, k, \log(1/\varepsilon)$)-size circuit C expressing an n-qubit unitary which is an ε-approximation to D.

Theorem

ApproxCircuitExtraction is #**P**-hard.

Bypassing circuit extraction

- The motivation for extracting a circuit, is to run the diagram on a quantum computer.
- So really, we just want to *sample* from the unitary diagram.

Bypassing circuit extraction

- The motivation for extracting a circuit, is to run the diagram on a quantum computer.
- So really, we just want to *sample* from the unitary diagram.

UnitaryZXSampling
Input: A ZX-diagram D with n inputs and outputs.
Promise: D is proportional to some unitary U.
Output: A sample $\vec{x} \in \{0,1\}^n$ from a probability distribution, given by (or sufficiently close to) $|\langle \vec{x}|U|0\cdots 0\rangle|^2$.

Bypassing circuit extraction

- The motivation for extracting a circuit, is to run the diagram on a quantum computer.
- So really, we just want to *sample* from the unitary diagram.

UnitaryZXSampling
Input: A ZX-diagram D with n inputs and outputs.
Promise: D is proportional to some unitary U.
Output: A sample $\vec{x} \in \{0, 1\}^n$ from a probability distribution, given by (or sufficiently close to) $|\langle \vec{x}|U|0 \cdots 0\rangle|^2$.

Theorem

NP randomly polynomially reduces to **UnitaryZXSampling**.
That is: $\mathbf{NP} \subseteq \mathbf{BPP}^{\mathbf{UnitaryZXSampling}}$.

Bypassing circuit extraction

- The motivation for extracting a circuit, is to run the diagram on a quantum computer.
- So really, we just want to *sample* from the unitary diagram.

UnitaryZXSampling
Input: A ZX-diagram D with n inputs and outputs.
Promise: D is proportional to some unitary U.
Output: A sample $\vec{x} \in \{0, 1\}^n$ from a probability distribution, given by (or sufficiently close to) $|\langle \vec{x}|U|0\cdots 0\rangle|^2$.

Theorem

NP randomly polynomially reduces to **UnitaryZXSampling**.
That is: $\mathbf{NP} \subseteq \mathbf{BPP}^{\mathbf{UnitaryZXSampling}}$.

Corollary

If there were a procedure to run unitary ZX-diagrams on a quantum computer, then $\mathbf{NP} \subseteq \mathbf{BQP}$.

Conclusion

- Circuit extraction is hard.
- Allowing approximate extraction or some ancillae does not make it easier.
- In fact, any way to extract samples from a unitary ZX-diagram is at least **NP**-hard.

Conclusion

- Circuit extraction is hard.
- Allowing approximate extraction or some ancillae does not make it easier.
- In fact, any way to extract samples from a unitary ZX-diagram is at least **NP**-hard.

Future work:

- What is the exact complexity of **CircuitExtraction**? The best bound we have is $\mathbf{FNP}^{\mathbf{NP}^{\#\mathbf{P}}}$.
- Is Circuit Extraction for deterministic measurement patterns also hard?

Thank you for your attention!

Circuit Extraction for ZX-diagrams can be **#P***-hard*
Niel de Beaudrap, Aleks Kissinger & John van de Wetering
arXiv:2202.09194