Quantum Graph-State Synthesis with SAT

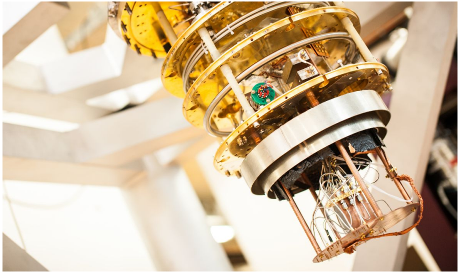
Sebastiaan Brand   Tim Coopmans   Alfons Laarman

Pragmatics of SAT, 4 July 2023

Universiteit Leiden
The Netherlands

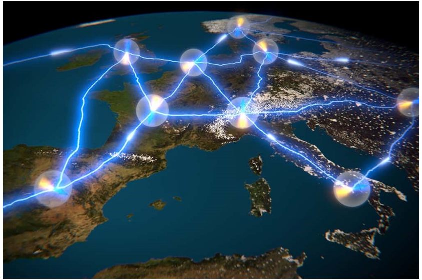
Quantum computing & networking

Quantum computing



- polynomial speedup to a lot of problems (including SAT)
- exponential speedup to some problems

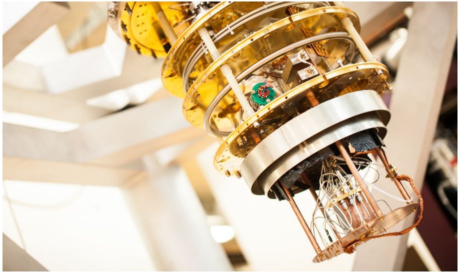
Quantum networking



- better cryptography
- connect quantum computers

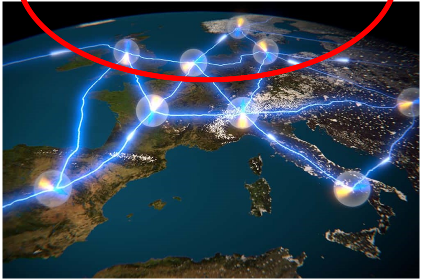
Quantum computing & networking

Quantum computing



- polynomial speedup to a lot of problems (including SAT)
- exponential speedup to some problems

Quantum networking



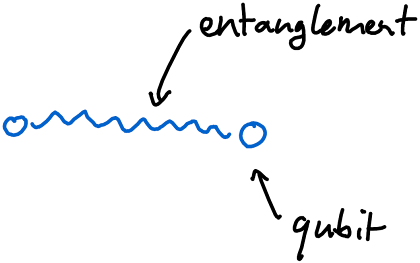
- better cryptography
- connect quantum computers

Graph states

Quantum states represented by (undirected) graphs

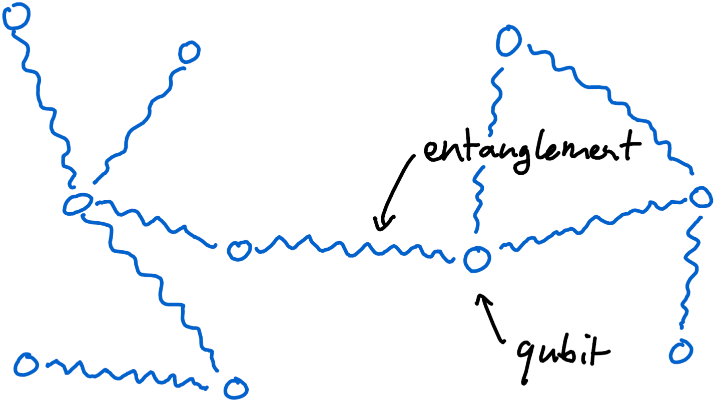
Graph states

Quantum states represented by (undirected) graphs

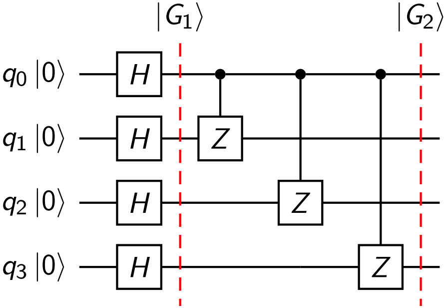


Graph states

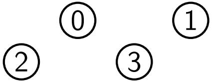
Quantum states represented by (undirected) graphs



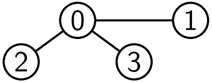
Graph states



(a) Circuit generating $|G_2\rangle$.

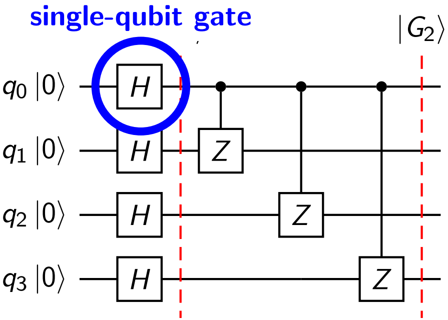


(b) Graph G_1.

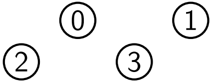


(c) Graph G_2.

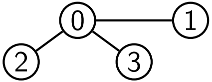
Graph states



(a) Circuit generating $|G_2\rangle$.

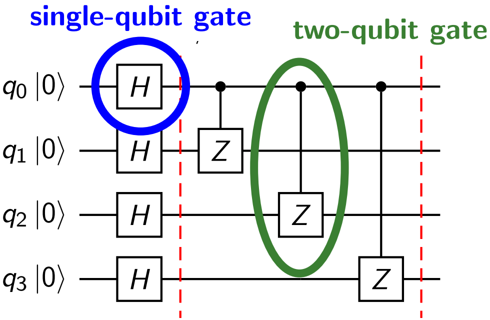


(b) Graph G_1.

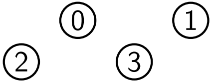


(c) Graph G_2.

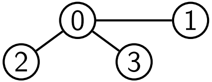
Graph states



(a) Circuit generating $|G_2\rangle$.

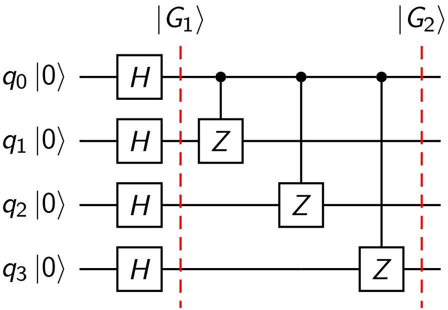


(b) Graph G_1.

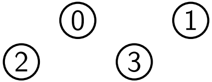


(c) Graph G_2.

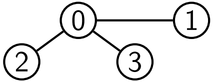
Graph states



(a) Circuit generating $|G_2\rangle$.



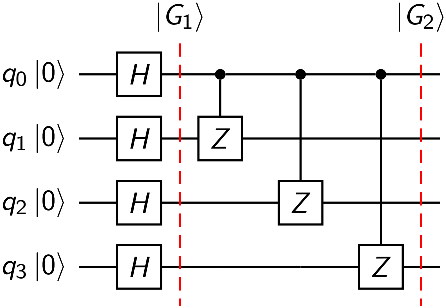
(b) Graph G_1.



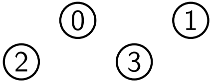
(c) Graph G_2.

$$|G_1\rangle = \frac{1}{4}(|0000\rangle + |0001\rangle + \cdots + |1011\rangle + |1100\rangle + |1101\rangle + |1110\rangle + |1111\rangle)$$

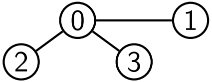
Graph states



(a) Circuit generating $|G_2\rangle$.



(b) Graph G_1.



(c) Graph G_2.

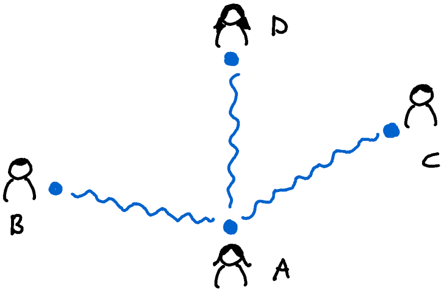
$$|G_2\rangle = \frac{1}{4}(|0000\rangle + |0001\rangle + \cdots + |1011\rangle - |1100\rangle + |1101\rangle + |1110\rangle - |1111\rangle)$$

Graph-state synthesis - example problem

Alice wants to run a quantum secret sharing protocol between herself, Bob, Charlie, and Diana

Graph-state synthesis - example problem

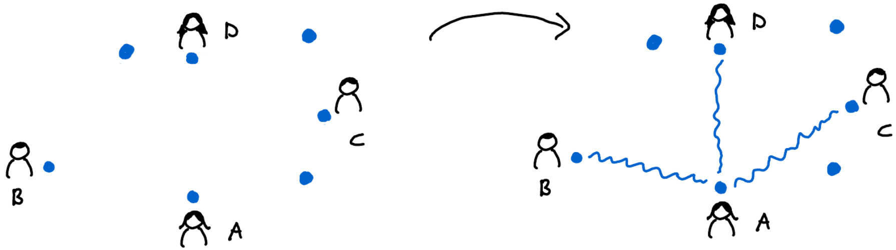
Alice wants to run a quantum secret sharing protocol between herself, Bob, Charlie, and Diana



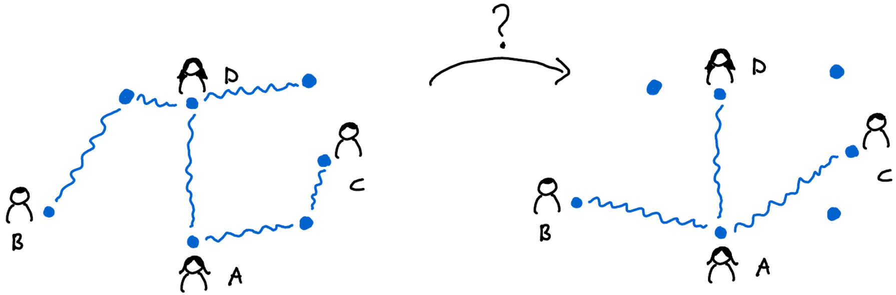
They need to share a specific quantum state (the GHZ state), described by this graph

$$|\mathrm{GHZ}_4\rangle = \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)$$

Graph-state synthesis - problem definition

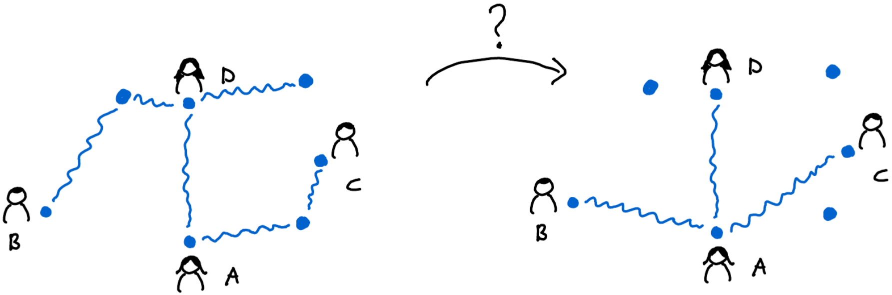


Graph-state synthesis - problem definition



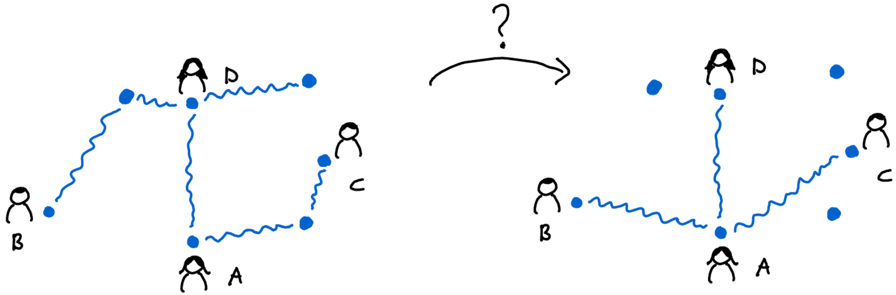
Graph-state synthesis - problem definition

Goal: generate a target graph, given some initial entanglement, **using only local operations**



Graph-state synthesis - problem definition

Goal: generate a target graph, given some initial entanglement, **using only local operations**



This problem has been shown to be NP-complete[1]

[1] Dahlberg, A. et al. (2020). How to transform graph states using single-qubit operations: computational complexity and algorithms.

Graph-state synthesis - local operations

local quantum operations

corresponding graph operations

Graph-state synthesis - local operations

local quantum operations

corresponding graph operations

single-qubit quantum gates

Graph-state synthesis - local operations

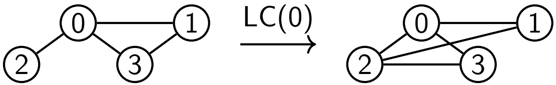
local quantum operations | **corresponding graph operations**

single-qubit quantum gates

single-qubit measurements

Graph-state synthesis - local operations

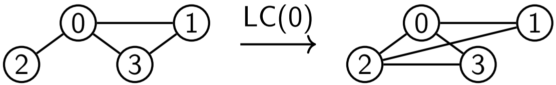
local quantum operations	**corresponding graph operations**
single-qubit quantum gates	**local complementations**
single-qubit measurements	



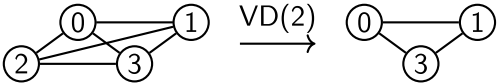
Graph-state synthesis - local operations

local quantum operations	**corresponding graph operations**
single-qubit quantum gates	**local complementations**
single-qubit measurements	**vertex deletions**

local complementations



vertex deletions



SAT encoding

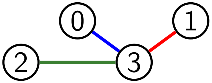
SAT encoding

Use a Boolean variable x_{uv} for each edge $(u, v) \in \mathbb{U} = \{(u, v) \in V \times V \mid u < v\}$

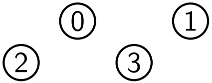
SAT encoding

Use a Boolean variable x_{uv} for each edge $(u, v) \in \mathbb{U} = \{(u, v) \in V \times V \mid u < v\}$

Examples:



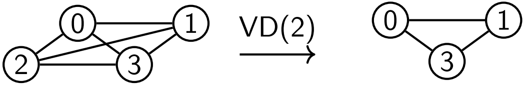
encoded as $\neg x_{01} \wedge \neg x_{02} \wedge \mathbf{x_{03}} \wedge \neg x_{12} \wedge \mathbf{x_{13}} \wedge \mathbf{x_{23}}$



encoded as $\neg x_{01} \wedge \neg x_{02} \wedge \neg x_{03} \wedge \neg x_{12} \wedge \neg x_{13} \wedge \neg x_{23}$

SAT encoding

vertex deletion

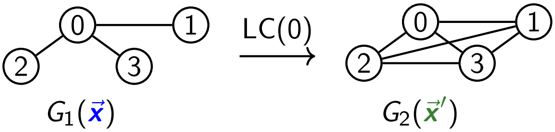


$G_1(\vec{x})$ $\quad$ $G_2(\vec{x}')$

$$VD_k = \bigwedge_{(u,v)\in\mathbb{U}} \begin{cases} \neg \mathbf{x}'_{\mathbf{uv}} & \text{if } u = k \text{ or } v = k \\ \mathbf{x}'_{\mathbf{uv}} \leftrightarrow \mathbf{x}_{\mathbf{uv}} & \text{otherwise.} \end{cases}$$

SAT encoding

local complementation



$$LC_k = \bigwedge_{(u,v)\in\mathbb{U}} \begin{cases} \mathbf{x}'_{\mathbf{uv}} \leftrightarrow \neg((\mathbf{x}_{\mathbf{uk}} \wedge \mathbf{x}_{\mathbf{vk}}) \oplus \neg \mathbf{x}_{\mathbf{uv}}) & \text{if } u \neq k \text{ and } v \neq k \\ \mathbf{x}'_{\mathbf{uv}} \leftrightarrow \mathbf{x}_{\mathbf{uv}} & \text{otherwise.} \end{cases}$$

SAT encoding

$$\left.\begin{array}{l} LC_0, LC_1, \ldots, LC_{n-1} \\ VD_0, VD_1, \ldots, VD_{n-1} \end{array}\right\}$$ combine into single global transition relation $R(\vec{x}, \vec{x}')$

$R(\vec{x}, \vec{x}')$ encodes all 1-step transformations and has $n^2 + \log n$ variables and $3.5n^3 + O(n^2 \log n)$ clauses.

SAT encoding

$$\left.\begin{array}{l} LC_0, LC_1, \ldots, LC_{n-1} \\ VD_0, VD_1, \ldots, VD_{n-1} \end{array}\right\} \text{combine into single global transition relation } R(\vec{x}, \vec{x}')$$

$R(\vec{x}, \vec{x}')$ encodes all 1-step transformations and has $n^2 + \log n$ variables and $3.5n^3 + O(n^2 \log n)$ clauses.

To encode multiple sequential transformation steps, we can use bounded model checking:

$$\overbrace{S(\vec{x}_1)}^{\text{starting graph}} \quad \wedge \quad \underbrace{R(\vec{x}_1, \vec{x}_2) \wedge R(\vec{x}_2, \vec{x}_3) \wedge \cdots \wedge R(\vec{x}_{d-1}, \vec{x}_d)}_{\text{any sequence of LC+VD of length } d} \quad \wedge \quad \overbrace{T(\vec{x}_d)}^{\text{target graph}}$$

Completeness threshold

Theorem

If a transformation from G_s to G_t exists using local complementations and vertex deletions, a transformation of length $\leq 2.5n$ exists, where n is the number of vertices in G_s.

Proof (sketch)

Completeness threshold

Theorem

If a transformation from G_s to G_t exists using local complementations and vertex deletions, a transformation of length $\leq 2.5n$ exists, where n is the number of vertices in G_s.

Proof (sketch)

1. *All local complementations can be done before the vertex deletions*

Completeness threshold

Theorem

If a transformation from G_s to G_t exists using local complementations and vertex deletions, a transformation of length $\leq 2.5n$ exists, where n is the number of vertices in G_s.

Proof (sketch)

1. *All local complementations can be done before the vertex deletions*
2. *We know exactly which vertex deletions need to happen*

Completeness threshold

Theorem

If a transformation from G_s to G_t exists using local complementations and vertex deletions, a transformation of length $\leq 2.5n$ exists, where n is the number of vertices in G_s.

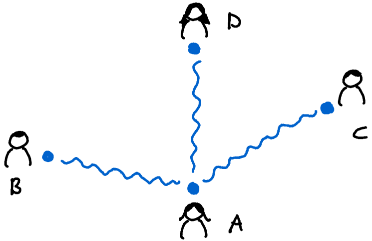
Proof (sketch)

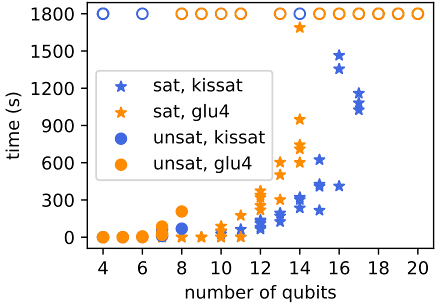
1. *All local complementations can be done before the vertex deletions*
2. *We know exactly which vertex deletions need to happen*
3. *We can bound the number of local complementations*[2]

[2] Bouchet, A. (1991). An efficient algorithm to recognize locally equivalent graphs.

Results

Synthesize the 4-qubit GHZ state from a random graph of n qubits





For comparison, various graph-state properties have been explored numerically up to 12 qubits[3]

[3] Cabello, A. et al. (2011). Optimal preparation of graph states.

Adding non-local operations

non-local quantum operations

corresponding graph operations

Adding non-local operations

non-local quantum operations

corresponding graph operations

two-
~~single~~-qubit quantum gates

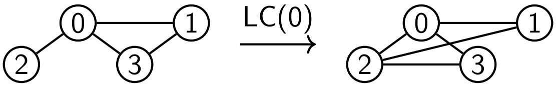
Adding non-local operations

non-local quantum operations

two-
~~single~~-qubit quantum gates

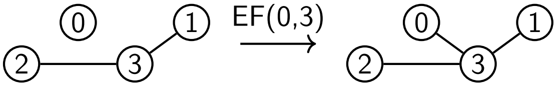
corresponding graph operations

local complementations



+

edge flips



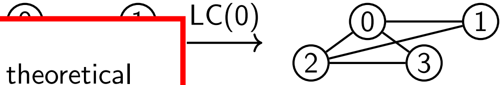
Adding non-local operations

non-local quantum operations

two-
~~single~~-qubit quantum gates

corresponding graph operations

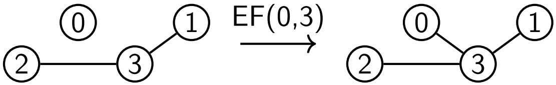
local complementations



significantly fewer theoretical results in this regime

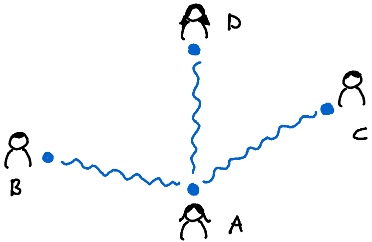
+

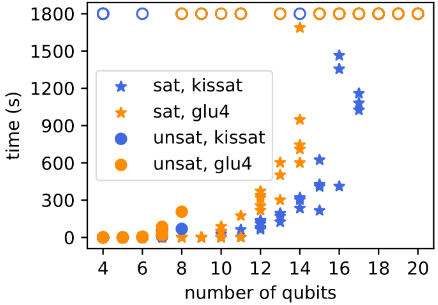
edge flips



Adding non-local operations

Synthesize the 4-qubit GHZ state from a random graph of n qubits, **allowing for two-qubit operations** on a limited subset of $V \times V$.

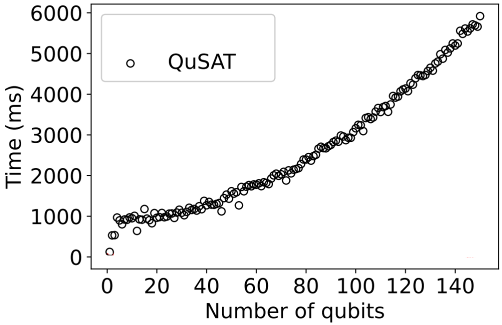




Not all quantum problems are hard

Not all quantum problems are hard

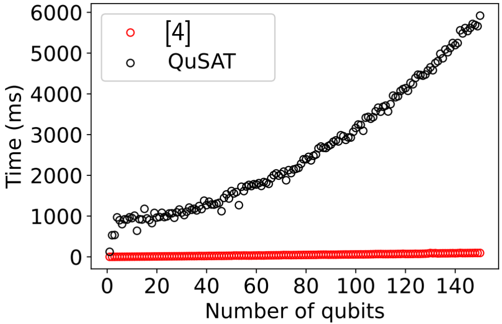
E.g. Clifford circuit equivalence checking:



[4] Berent, L., Burgholzer, L., Wille, R. (2022). Towards a SAT encoding for quantum circuits: A journey from classical circuits to Clifford circuits and beyond. arXiv preprint arXiv:2203.00698.

Not all quantum problems are hard

E.g. Clifford circuit equivalence checking:



[4] Thanos, D., Coopmans, T., Laarman, A. (2023) Fast equivalence checking of quantum circuits of Clifford gates. *To appear at ATVA 2023*

Summary

1. **Graph states** are an important subset of quantum states with many applications, e.g. in quantum networking
2. We want to synthesize graph states using **local operations** because these are easier to do
3. We translate this NP-complete problem to SAT and are able to find graph state transformations up to **17 qubits**
4. The method easily generalizes to **non-local operations**