Tensor network contraction

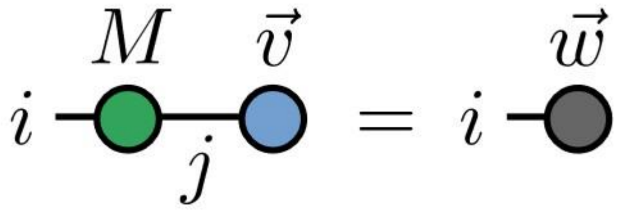
What is a tensor network?

"To-do" list of tensor multiplications – contractions

Examples

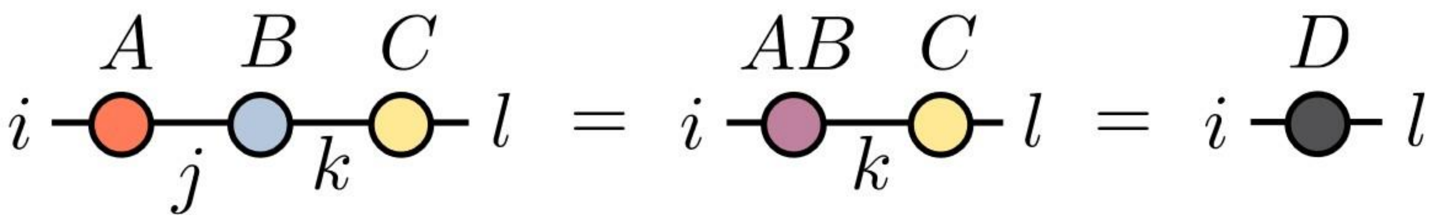
- matrix-vector multiplication:

$$\vec{w} = M\vec{v} \; : \; w_i = \sum_j M_{ij} v_j$$



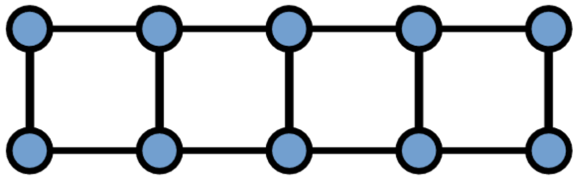
- matrix product:

$$D = ABC \; : \; D_{il} = \sum_{j,k} A_{ij} B_{jk} C_{kl}$$



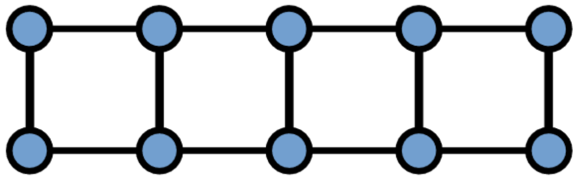
The <u>order</u> of contractions matters.

Example:



The order of contractions matters.

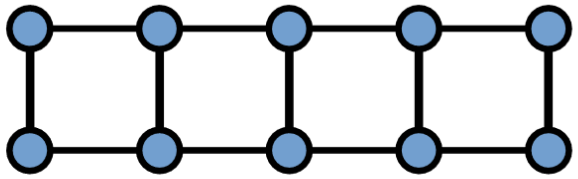
Example:



Schedule 1:

The <u>order</u> of contractions matters.

Example:



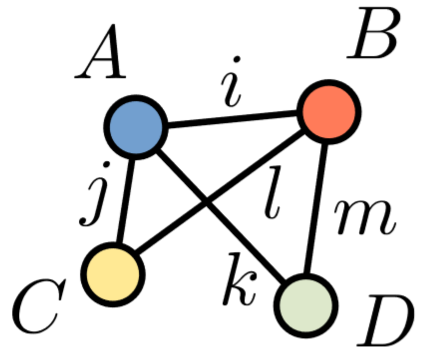
Schedule 1:

Schedule 2:

The <u>order</u> of contractions matters.

Example:

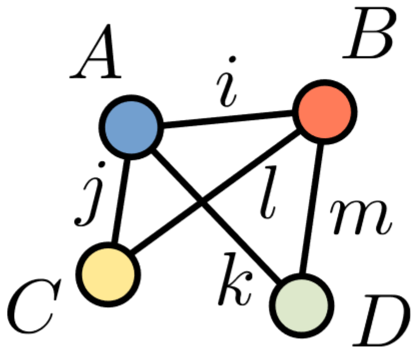
$$\sum_{ijklm} A_{ijk} B_{ilm} C_{jl} D_{km} :$$



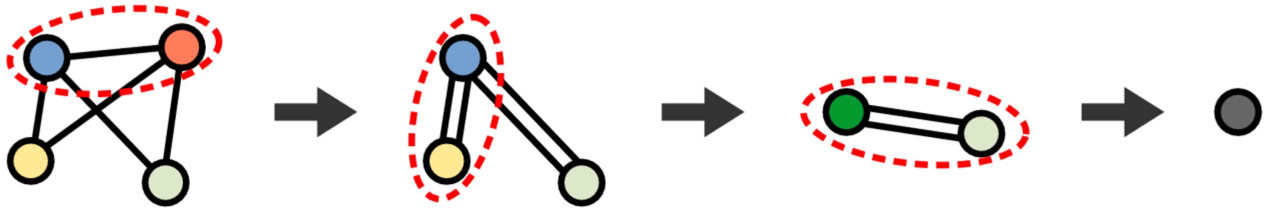
The order of contractions matters.

Example:

$$\sum_{ijklm} A_{ijk} B_{ilm} C_{jl} D_{km} :$$



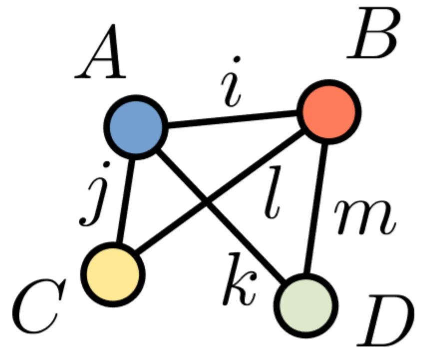
Schedule 1:



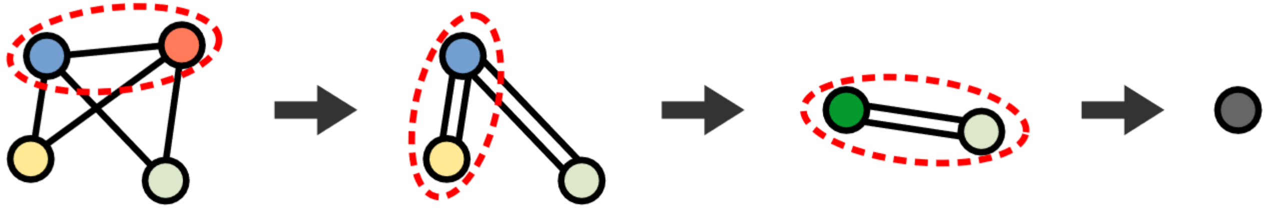
The order of contractions matters.

Example:

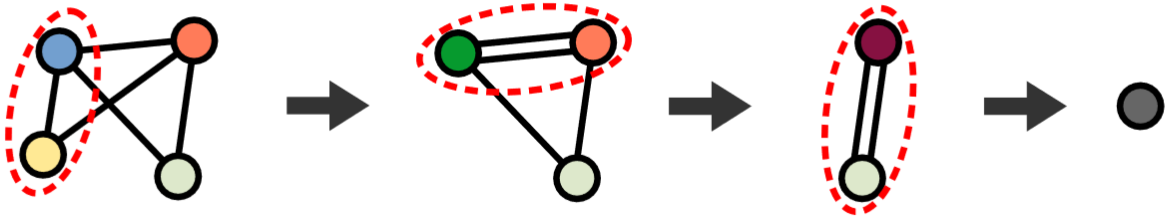
$$\sum_{ijklm} A_{ijk} B_{ilm} C_{jl} D_{km} :$$



Schedule 1:



Schedule 2:



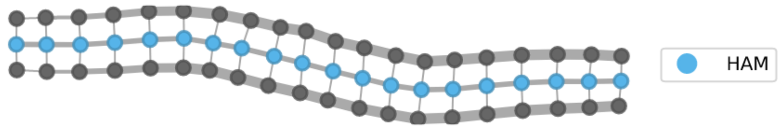
Contraction path finding

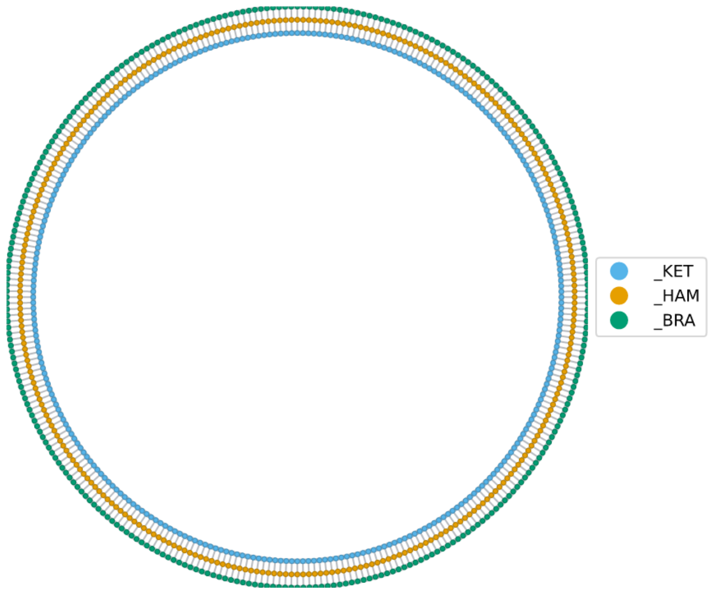
Easy cases:

Contraction path finding

Easy cases:

MPS algorithms; e.g., DMRG:



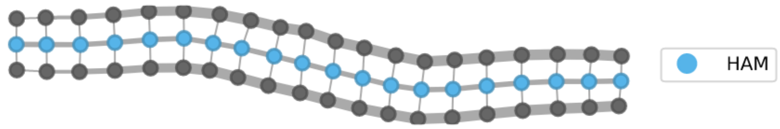


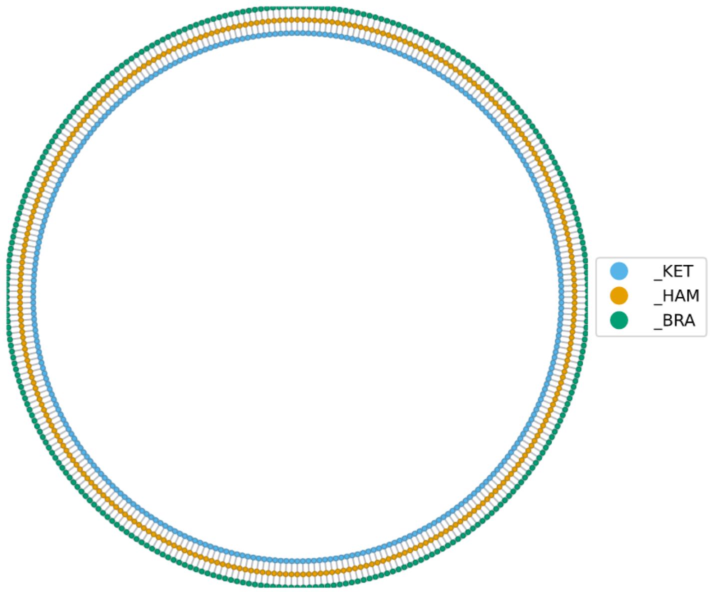
Johnnie Gray, *quimb* library
https://github.com/jcmgray/quimb

Contraction path finding

Easy cases:

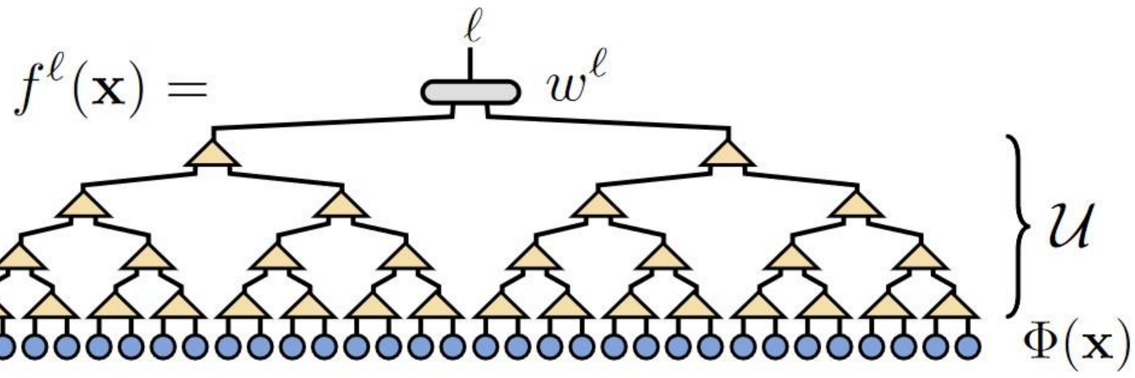
MPS algorithms; e.g., DMRG:





Johnnie Gray, *quimb* library
https://github.com/jcmgray/quimb

Tree TNs:



Stoudenmire (2018)

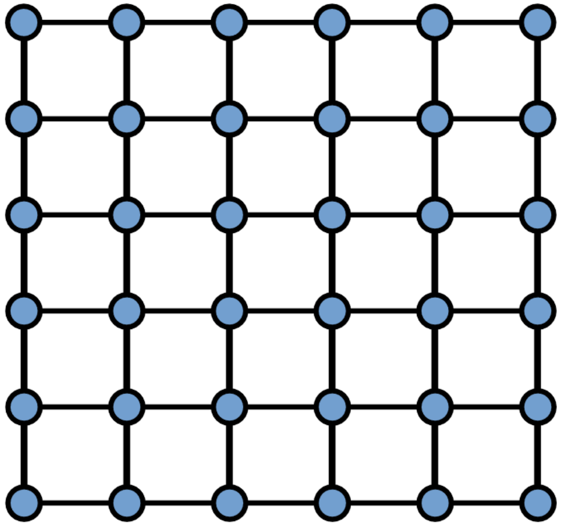
Contraction path finding

Less obvious: higher dimensions

Contraction path finding

Less obvious: higher dimensions

Coarse graining:

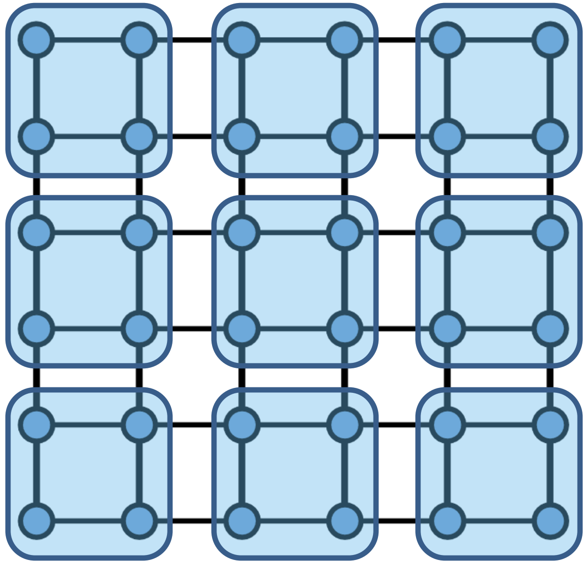


Levin & Nave (2007)

Contraction path finding

Less obvious: higher dimensions

Coarse graining:

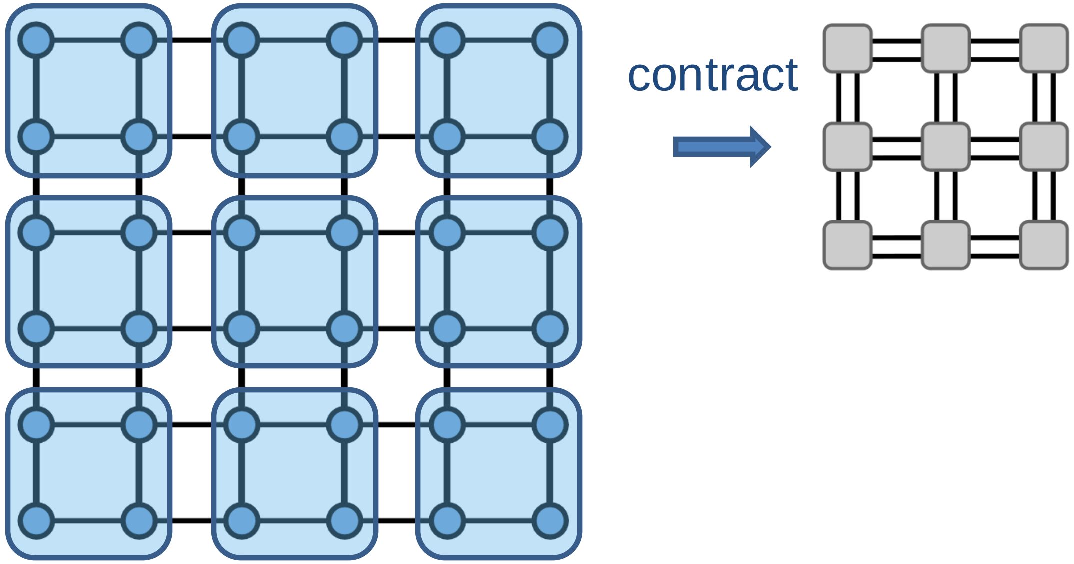


Levin & Nave (2007)

Contraction path finding

Less obvious: higher dimensions

Coarse graining:

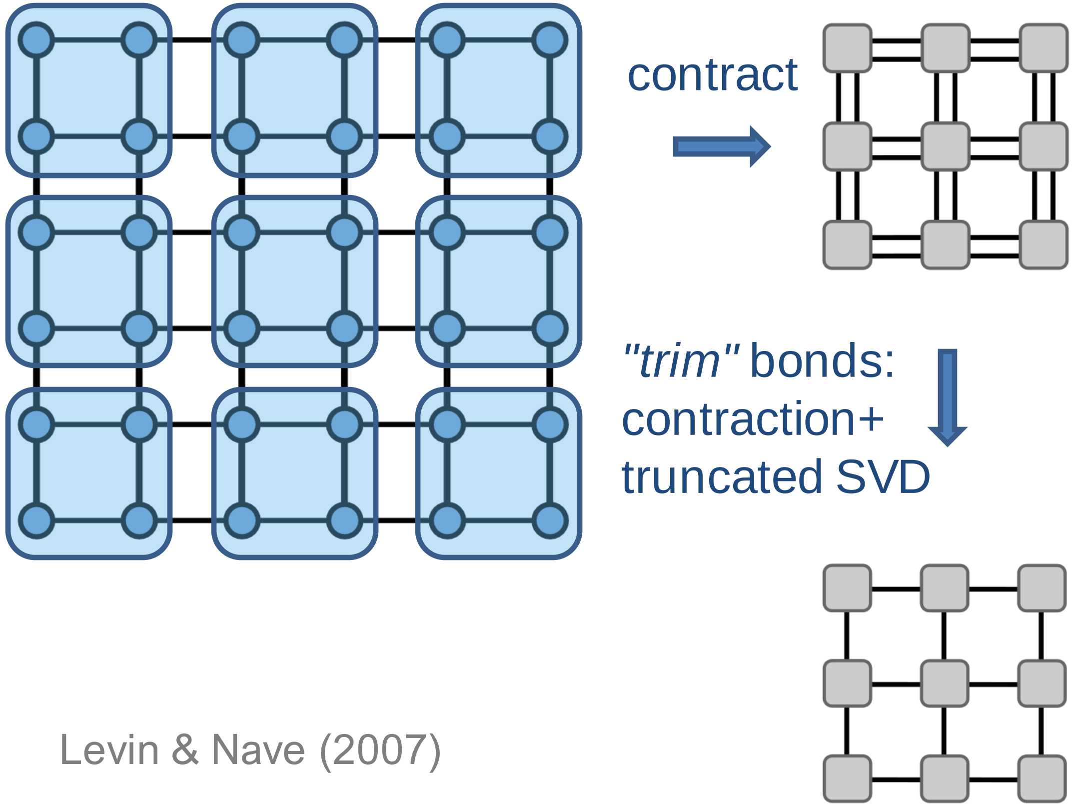


Levin & Nave (2007)

Contraction path finding

Less obvious: higher dimensions

Coarse graining:

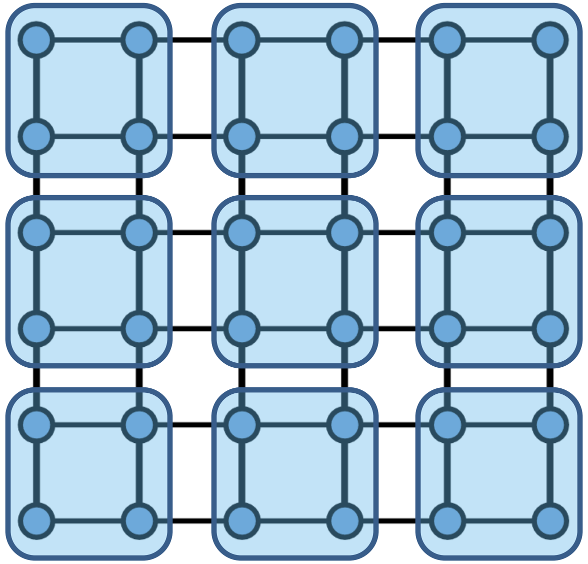


Levin & Nave (2007)

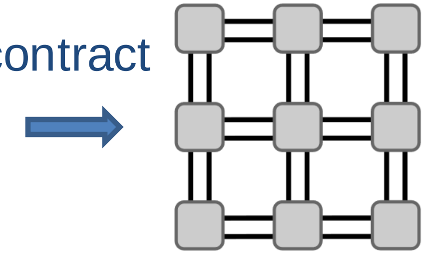
Contraction path finding

Less obvious: higher dimensions

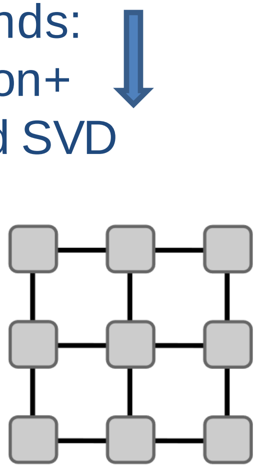
Coarse graining:



contract

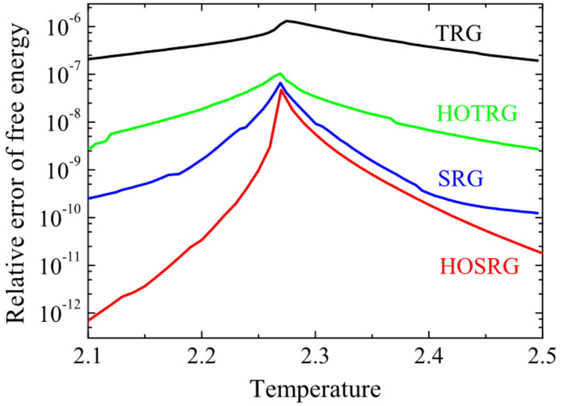


"trim" bonds:
contraction+
truncated SVD



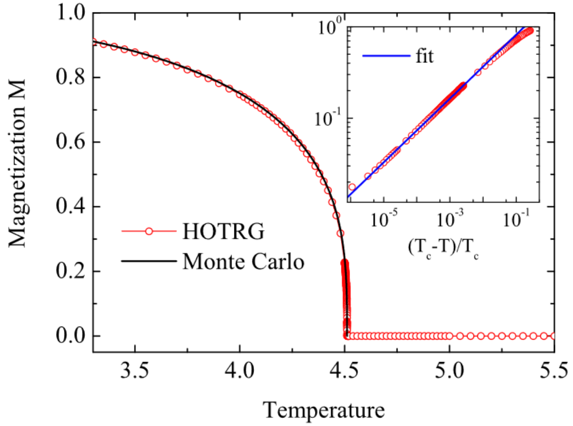
Levin & Nave (2007)

2D Ising:

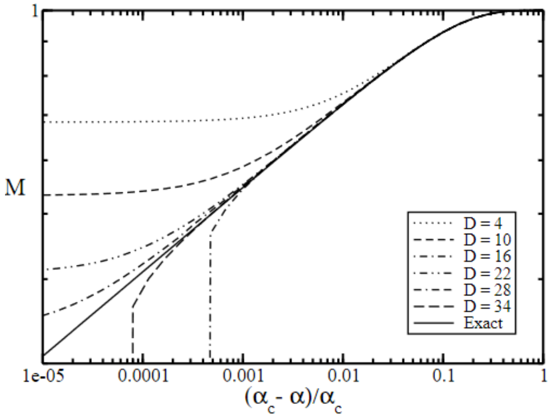


3D Ising:

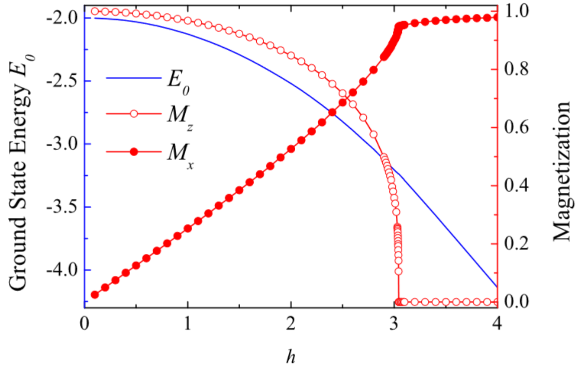
Xie *et al*. (2012)



triangular Ising close to T_c:



2D quantum Ising (2+1D):

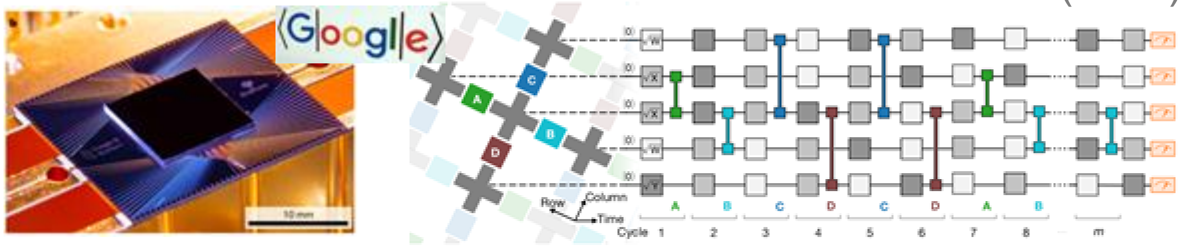


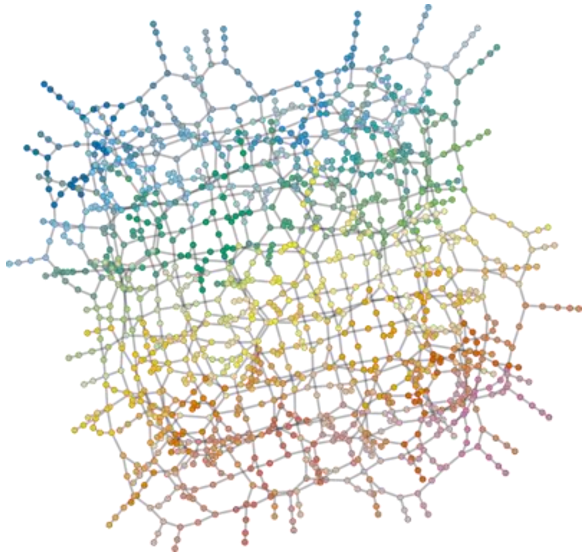
Contraction path finding

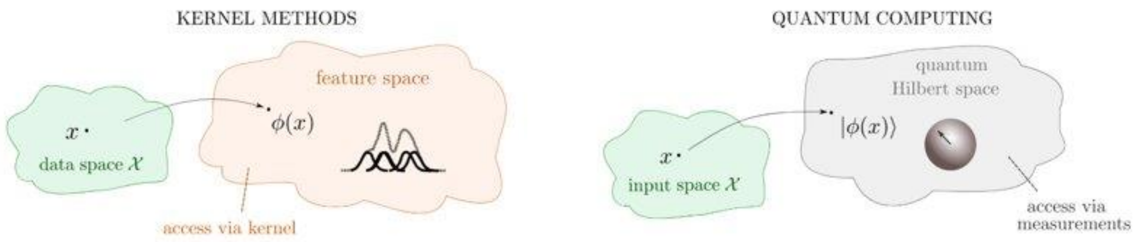
Not at all obvious: unstructured TNs

Quantum supremacy using a programmable superconducting processor

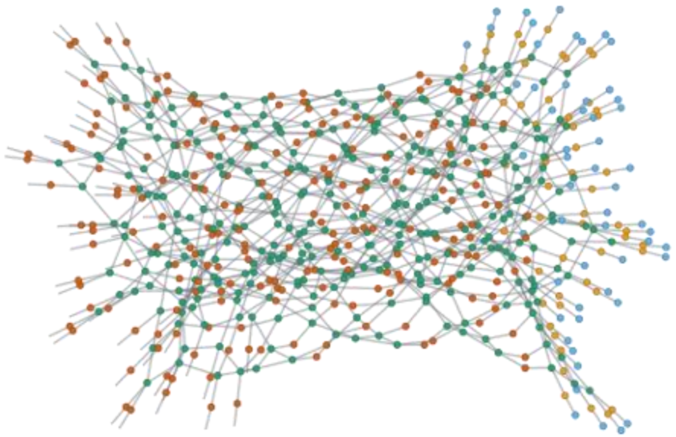
Arute *et al*. (2019)







Schuld (2021)



Coarse-graining irregular networks

Methods:

- Exhaustive search
- "Bubbling"
- Tree decomposition of line graph

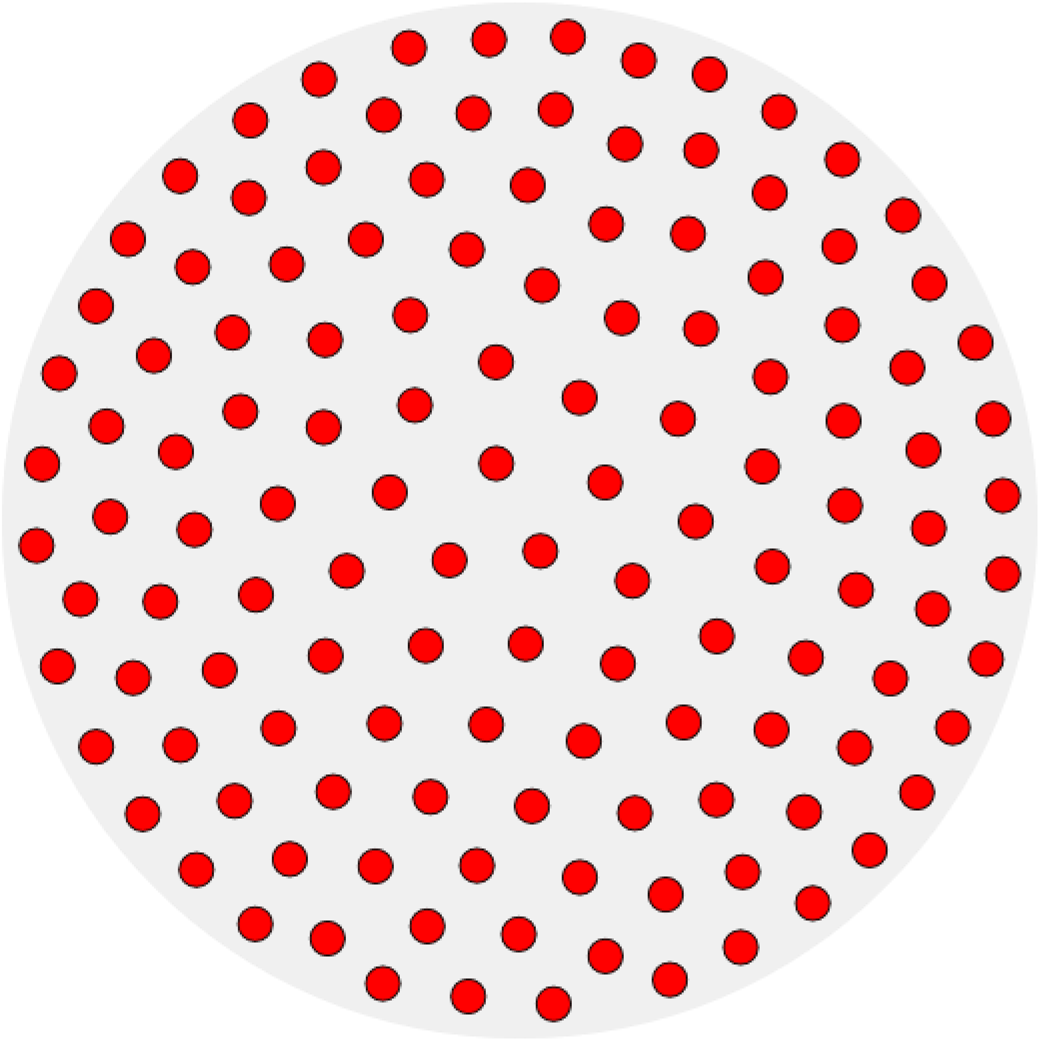
Coarse-graining irregular networks

Methods:

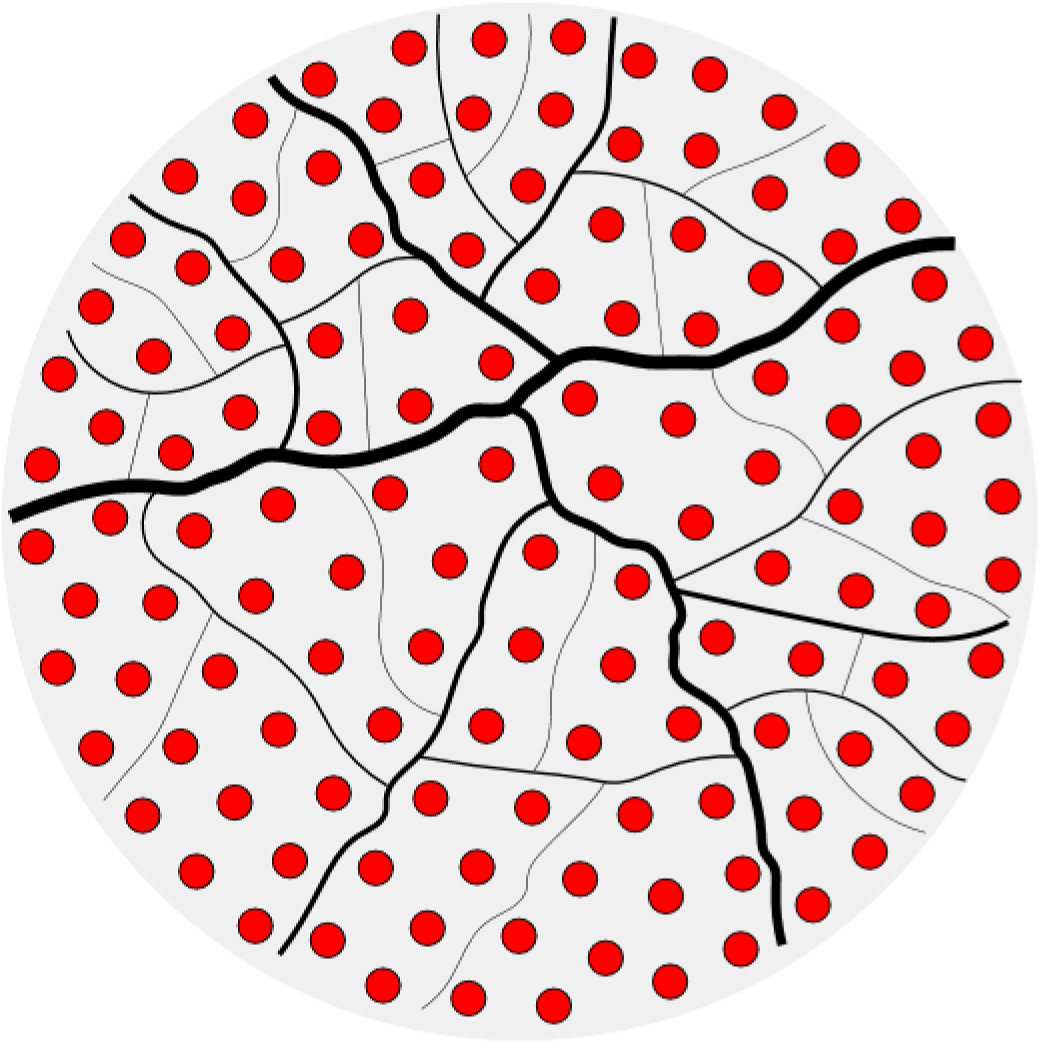
- Exhaustive search
- "Bubbling"
- Tree decomposition of line graph

- *Community detection*
- *Hierarchical graph partitioning*

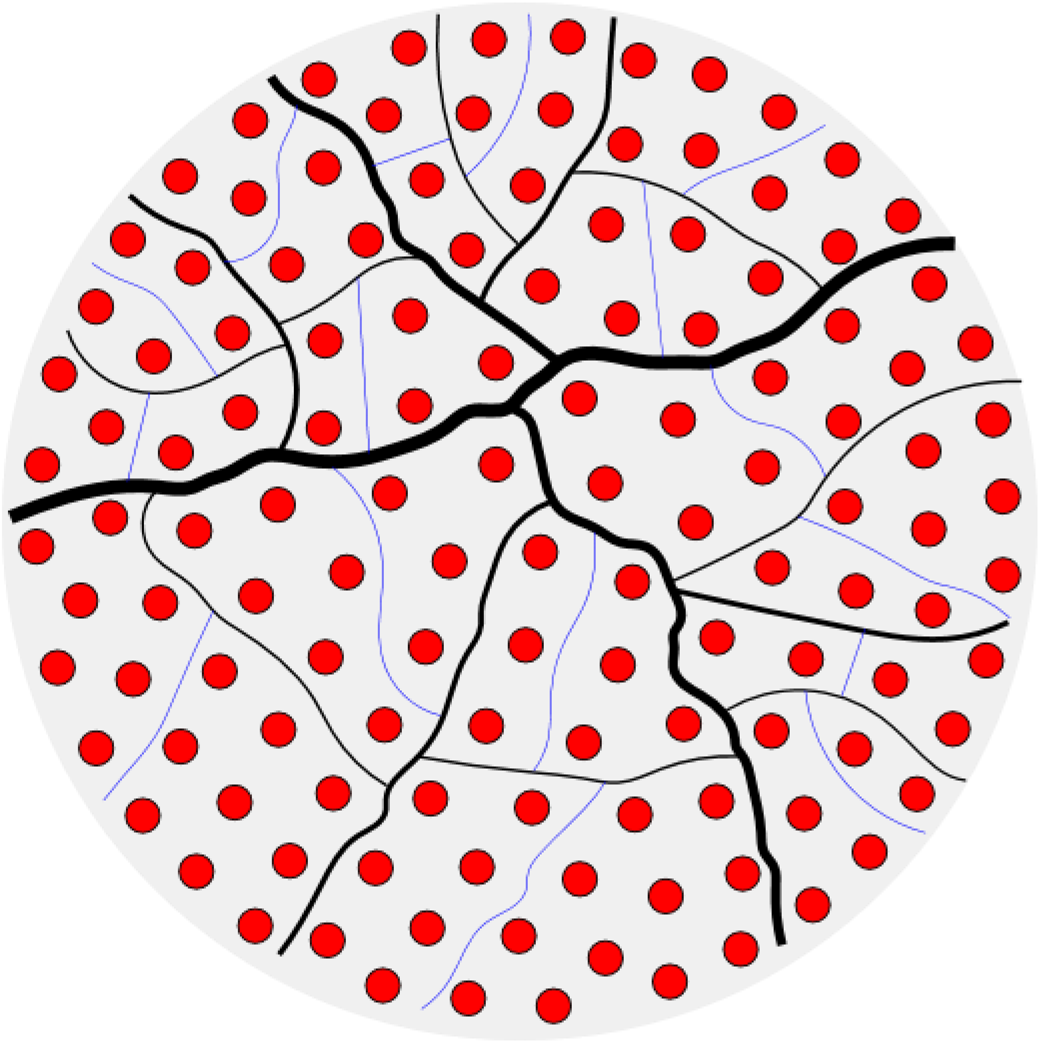
Hierarchical graph partitioning



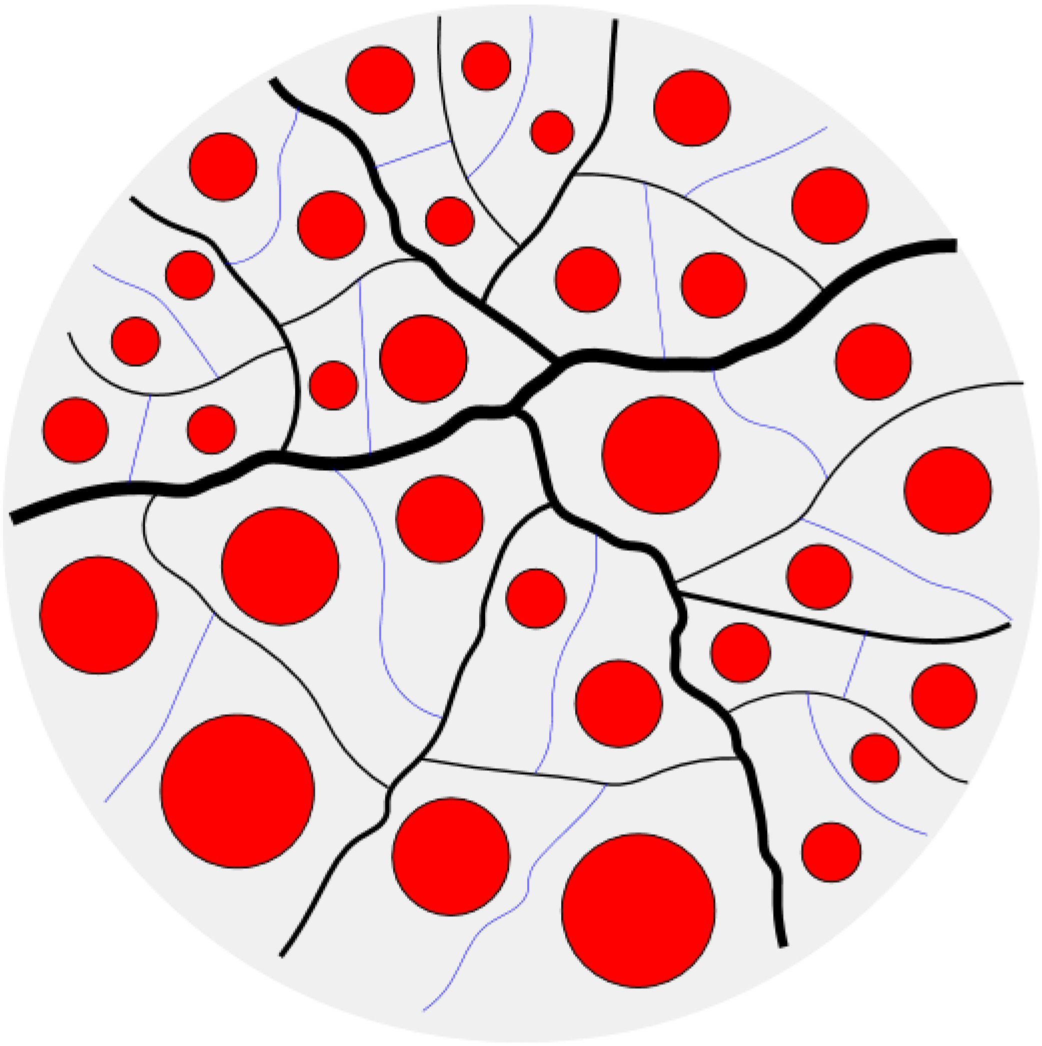
Hierarchical graph partitioning



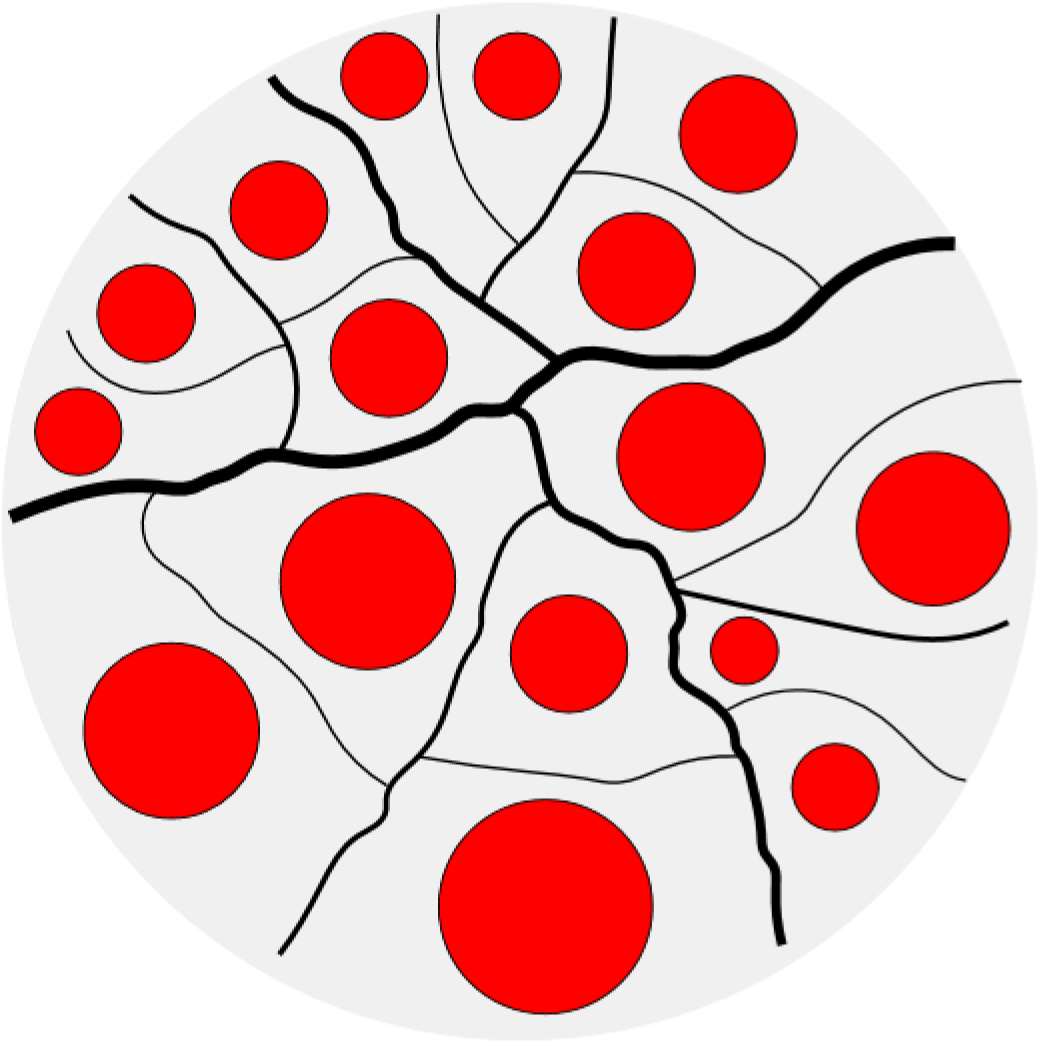
Hierarchical graph partitioning



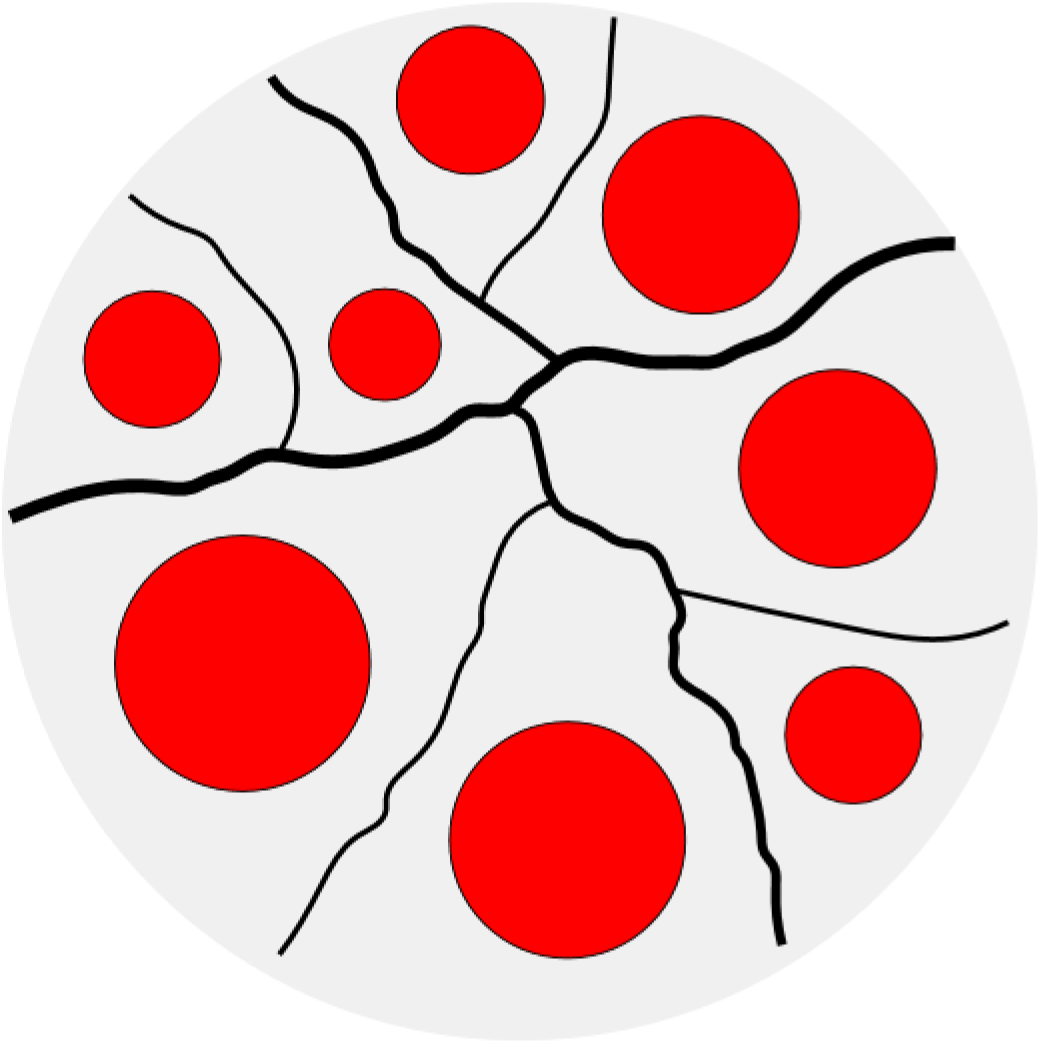
Hierarchical graph partitioning



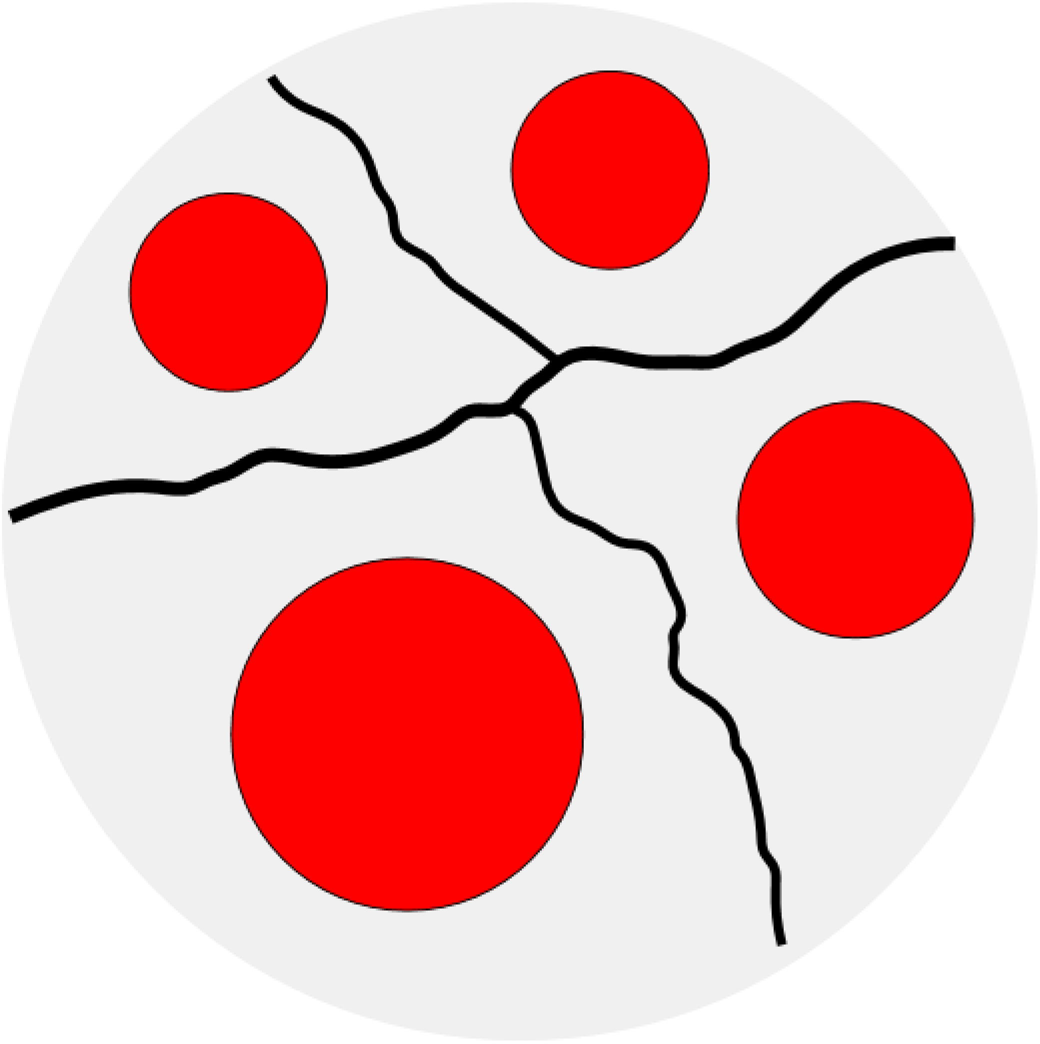
Hierarchical graph partitioning



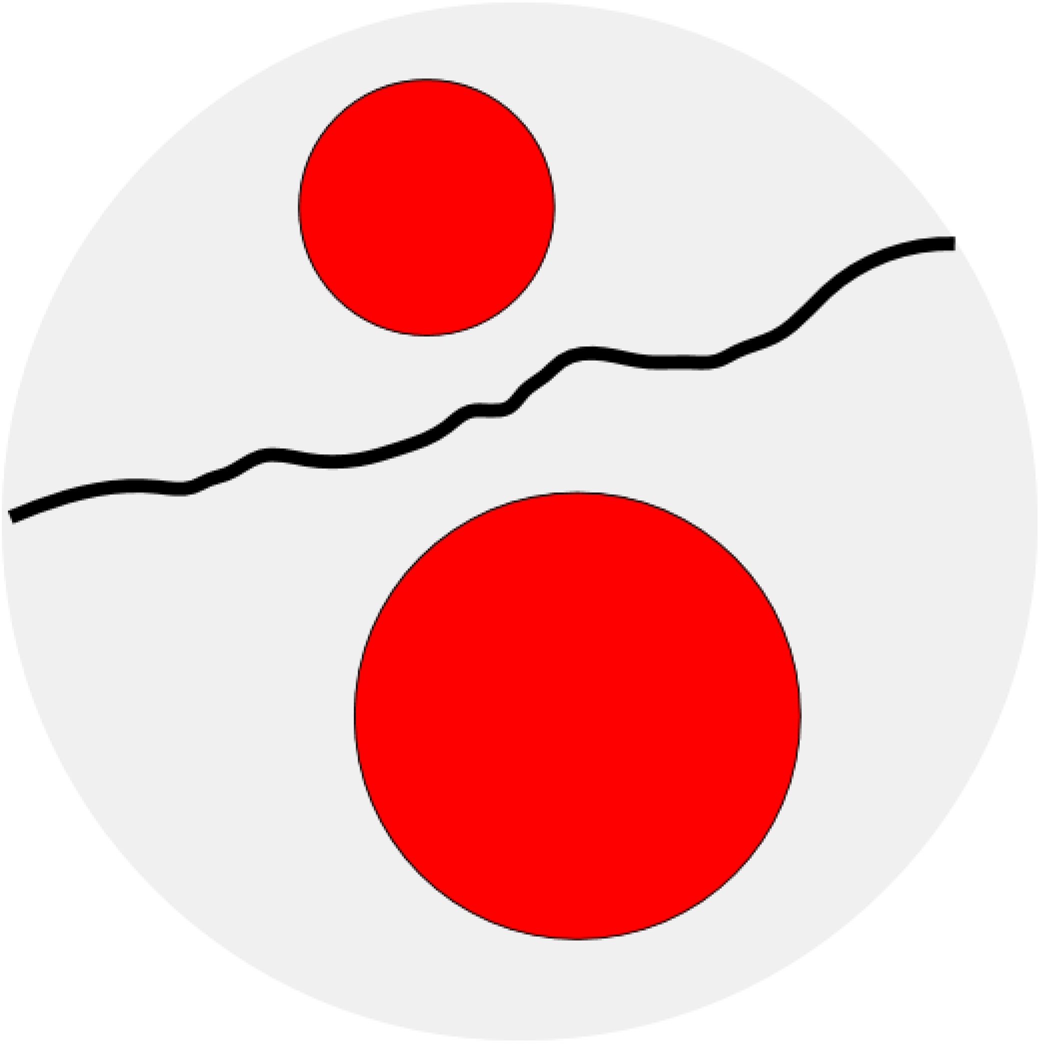
Hierarchical graph partitioning



Hierarchical graph partitioning



Hierarchical graph partitioning

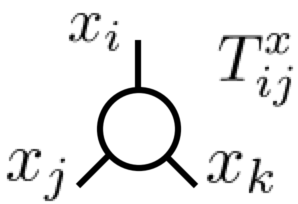


Hierarchical graph partitioning

42

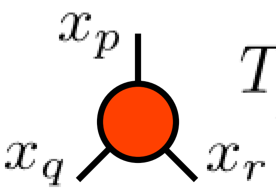
Model counting (#SAT)

variables: $x_i \in \{0, 1\},\ i = 1, \ldots, n_x$



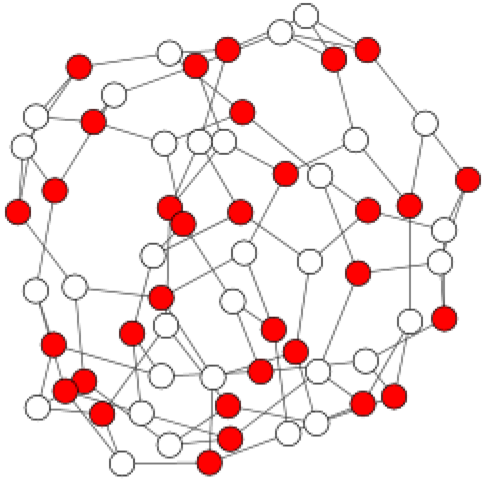
$T^x_{ijk} = \delta_{x_i x_j x_k}$

clauses: $c_m = x_p \vee x_q \vee x_r$



$T^c_{pqr} = x_p \vee x_q \vee x_r$

formula: $\phi = \bigwedge_{m=1}^{n_c} c_m$

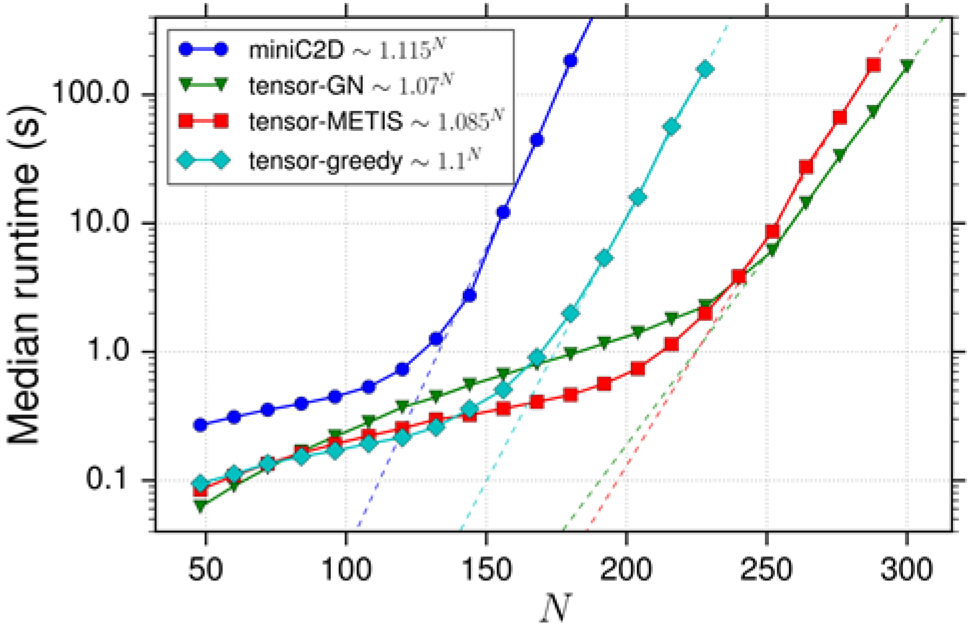


How many assignments $\vec{x}$ satisfy ϕ?

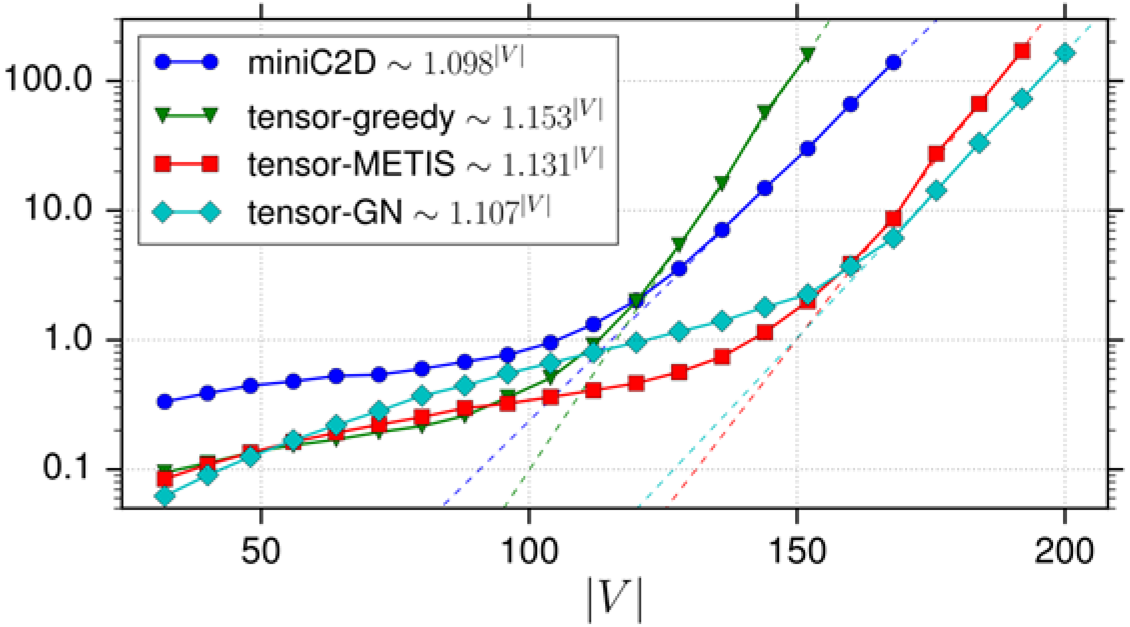
Model counting with TNs

Kourtis *et al*. (2018)

#1-in-3SAT:



#vertex covers on cubic graphs



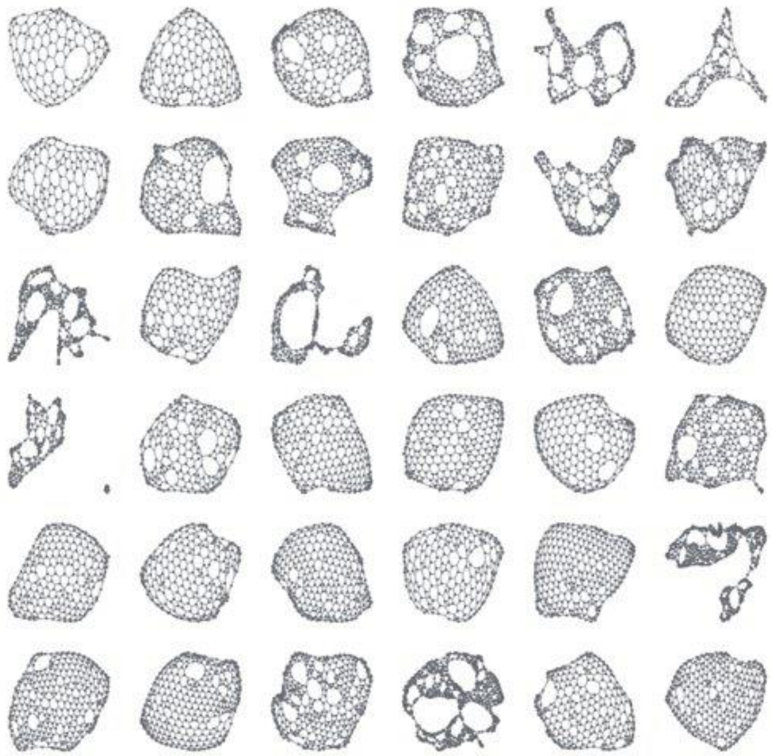
See also: Dudek, Dueñas-Osorio, Vardi (2019);
Dudek & Vardi (2020)

Weighted model counting with TNs

Gray & Kourtis (2021)

MC 2020 (Model Counting 2020)

The *1st International Competition on Model Counting (MC 2020)* is a competition to deepen the relationship between latest theoretical and practical development on the various model counting problems and their practical applications. It targets the problem of counting the number of models of a Boolean formula.

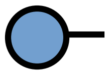


2020 champion: 69/100 instances solved within timeout
TN solver: 99/100 instances solved within timeout

Quantum computation simulation

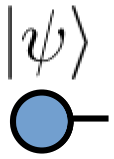
qubit: $|\psi\rangle = a|0\rangle + b|1\rangle = \begin{pmatrix} a \\ b \end{pmatrix}$:

$|\psi\rangle$



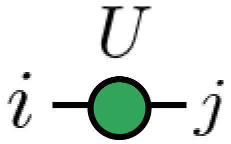
$i = 0, 1 \rightarrow |i\rangle$

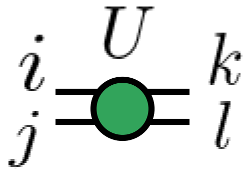
Quantum computation simulation

qubit: $|\psi\rangle = a|0\rangle + b|1\rangle = \begin{pmatrix} a \\ b \end{pmatrix}$: 

$i = 0, 1 \to |i\rangle$

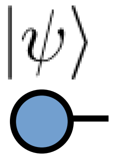
quantum gate: $|\psi'\rangle = U|\psi\rangle$

1-qubit: 

2-qubit: 

Quantum computation simulation

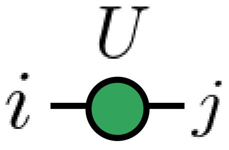
qubit: $|\psi\rangle = a|0\rangle + b|1\rangle = \begin{pmatrix} a \\ b \end{pmatrix}$:



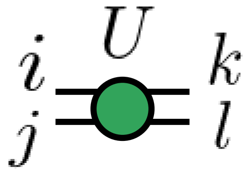
$i = 0, 1 \to |i\rangle$

quantum gate: $|\psi'\rangle = U|\psi\rangle$

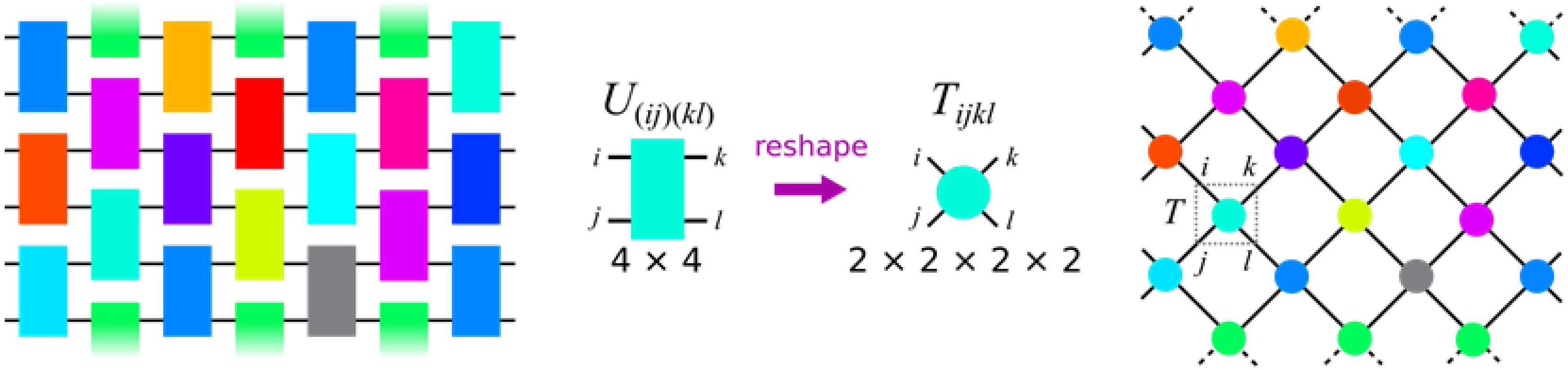
1-qubit:



2-qubit:



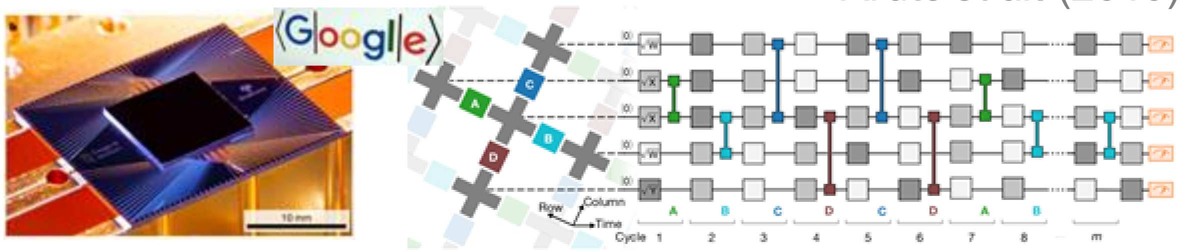
quantum circuit:

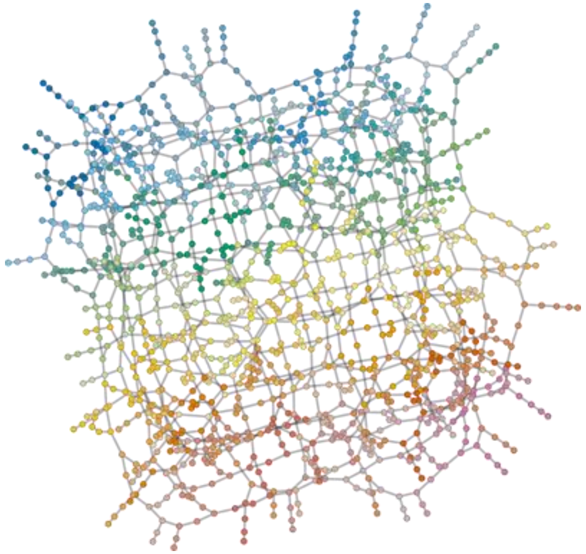


Random quantum circuit TN simulation

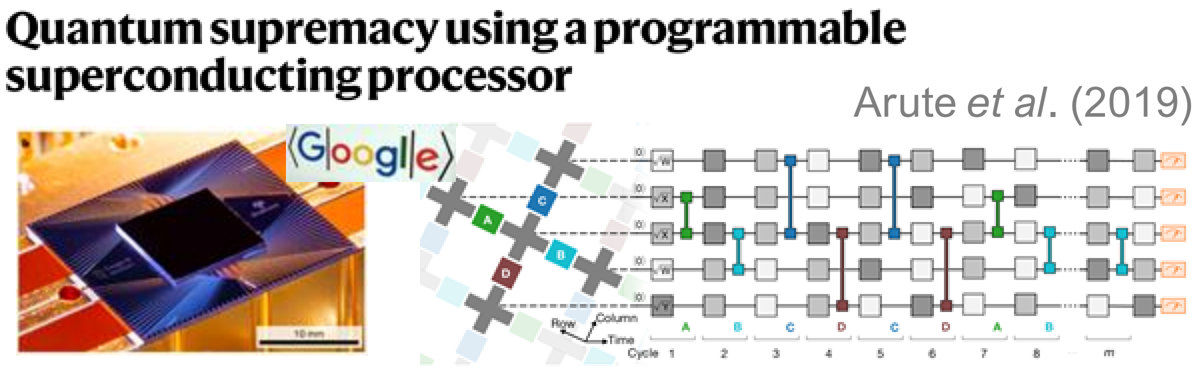
Quantum supremacy using a programmable superconducting processor

Arute *et al*. (2019)

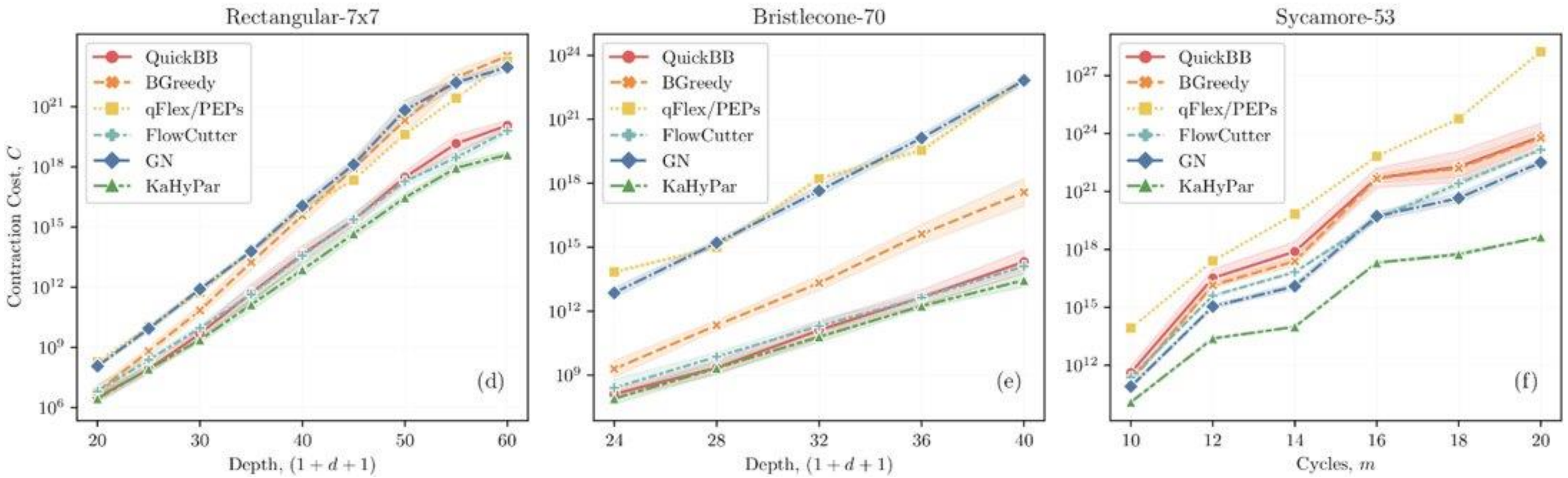


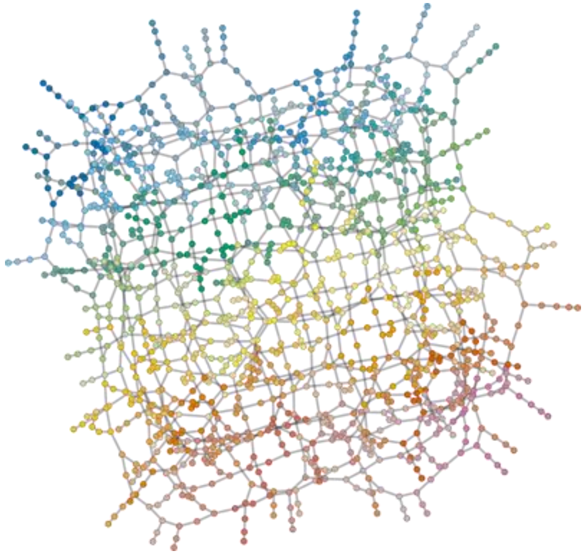


Random quantum circuit TN simulation



Arute *et al.* (2019)



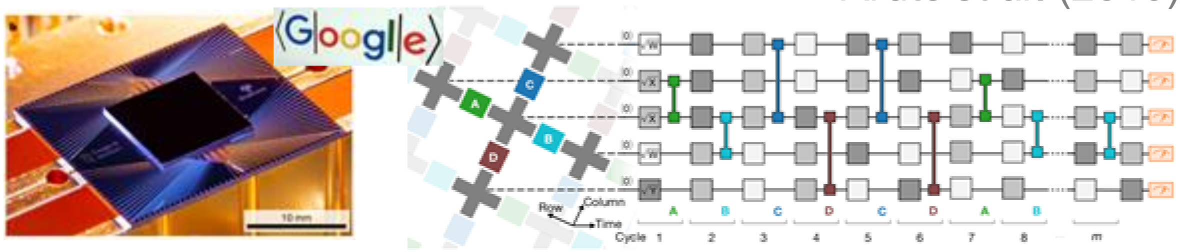


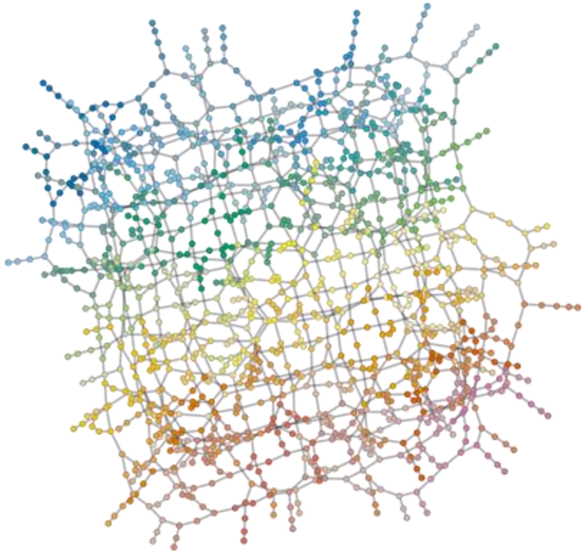
Gray & Kourtis (2021)

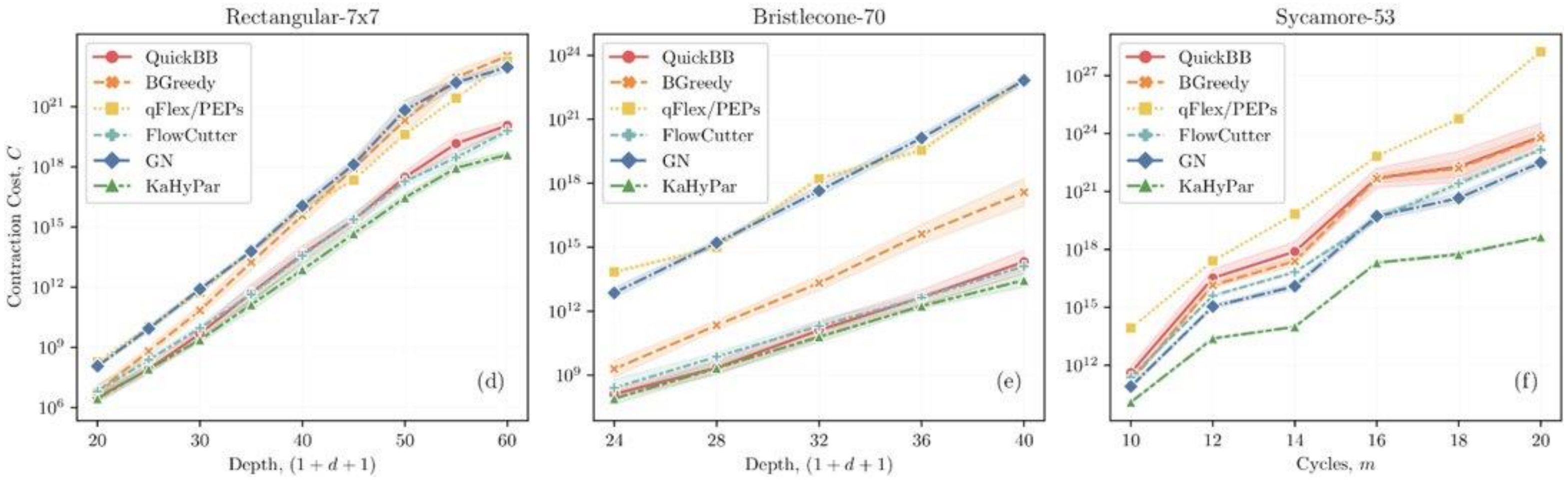
Random quantum circuit TN simulation

Quantum supremacy using a programmable superconducting processor

Arute *et al.* (2019)







Gray & Kourtis (2021)

Gray & Kourtis (2020) ~195 days @ 281 petaFLOPs (Summit) (*est.*)

Yong *et al.* (2021) ~300s @ 1.2 exaFLOPs (*est.*)

Pan, Chen, Zhang (2021) ~few dozen s @ exaFLOPs → 15 hours using 512 GPUs

Recap

Course outline

- Lecture 1
    a. Bird's-eye view: ~2.5 millennia of tensor networks
    b. Elementary theory of entanglement

- Lecture 2
    a. Matrix-product states & operators
    b. Eigenstate-finding; time evolution

- Lecture 3
    a. Logic as tensor networks
    b. Everything as tensor networks

- Lecture 4
    a. Tensor network contraction & examples
    b. Summary & outlook

Outlook

Future contractions

- Data science & ML
  - <u>many</u> new techniques
  - specialized hardware
- Quantum computation
  - tomography
  - hybrid quantum-classical algorithms
- Computational science
  - combinatorial optimization
  - approximate algorithms
- Physics
  - disordered systems (no symmetries)