A Practical Approach to Quantum Annealing

GOTO CHICAGO 2020

TRIFORK.
...think software

AGENDA

Practical Quantum Annealing - Part 2

- Constraint Satisfation Problems Revisited
- Networks and Network Algorithms
- Divide and Conquer - An Introduction

TRIFORK.
...think software

Objective function for optimisation - bias

Source: https://www.dwavesys.com/quantum-computing

Objective function for optimisation - bias

$$\mathbf{E}_{ising}(s) = \sum_{i=1}^{N} h_i s_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} J_{i,j} s_i s_j$$

OR

$$\mathbf{E}_{qubo}(a_i, b_{i,j}; q_i) = \sum_{i} a_i q_i + \sum_{i<j} b_{i,j} q_i q_j \,.$$

BINARY QUADRATIC MODEL

Binary Quadratic Model (BQM)

```
import dimod

bqm = dimod.BinaryQuadraticModel(
    {0: 1, 1: -1, 2: .5},          # Linear
    {(0, 1): .5, (1, 2): 1.5},     # Quadratic
    1.4,                           # Offset
    dimod.Vartype.SPIN)            # SPIN/BINARY
```

BINARY CSP

dwavebinarycsp

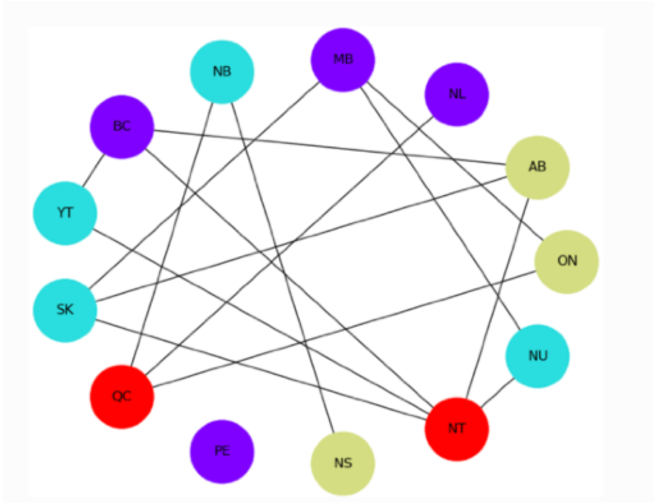
CONSTRAINT 1 $a = b$

CONSTRAINT 2 $b \neq c$

```
import dwavebinarycsp
import operator

csp=dwavebinarycsp.ConstraintSatisfactionProblem(‘BINARY')
csp.add_constraint(operator.eq, ['a', 'b'])
csp.add_constraint(operator.ne, ['b', 'c'])
result = csp.check({'a': 1, 'b': 1, 'c': 0}) # True
```

Formulating map colouring as a CSP



- View the map as a graph
  - Each area is a node
  - Each border is a vertex
- Area color as a binary constraint
  - **(qc_red, qc_yellow, qc_green, qc_blue)** can only be **(0,0,0,1), (0,0,1,0),(0,1,0,0),(1,0,0,0)**
- Two neighbouring area cannot have same color
  - **!(qc_red && on_red)**
  - **!(qc_green && on_green)**
  - ..
  - **!(qc_red && nb_red)**
  - ..

TRIFORK.
...think software

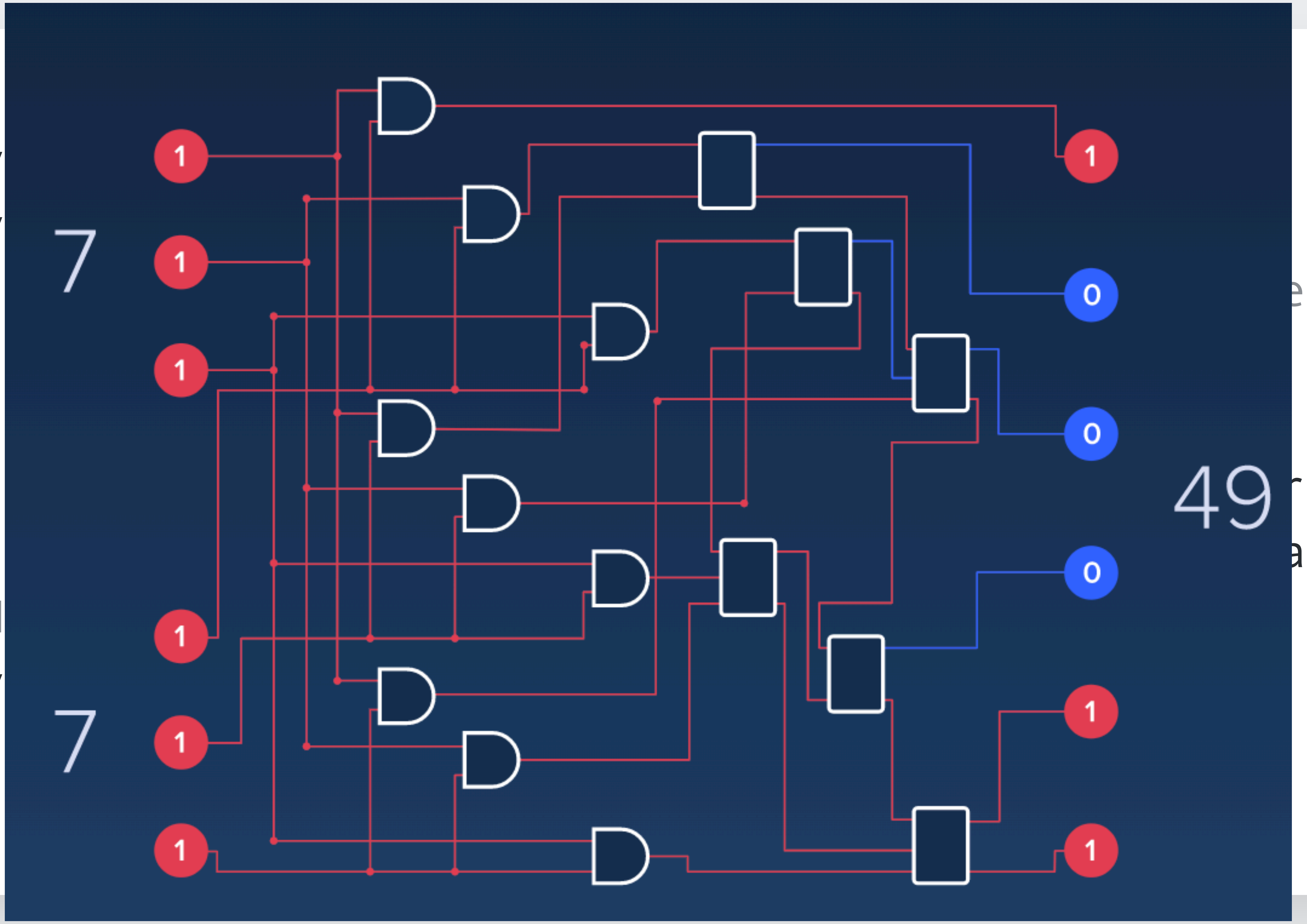
Demo

MAP COLORING

CONSTRAINT SATISFACTION

Example: Factoring

```
def factor(P)
    csp = dwav
    bqm = dwav
    # we know                                              e variables
    p_vars = [
    # convert
    fixed_vari                                              rmat(P)))
    fixed_vari                                              ables.items()}
    # fix prod
    for var, v
        bqm.fi
```



TRIFORK.
...think software

AGENDA

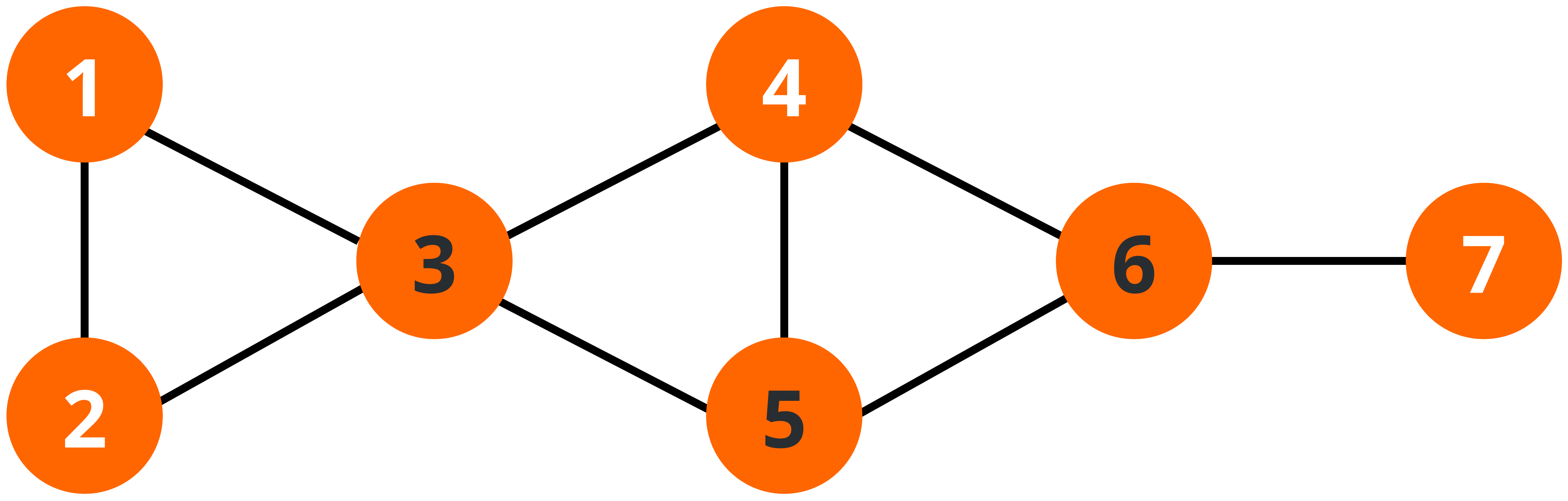
Practical Quantum Annealing - Part 2

Constraint Satisfation Problems Revisited

Networks and Network Algorithms

Divide and Conquer - An Introduction

Example: Vertex cover

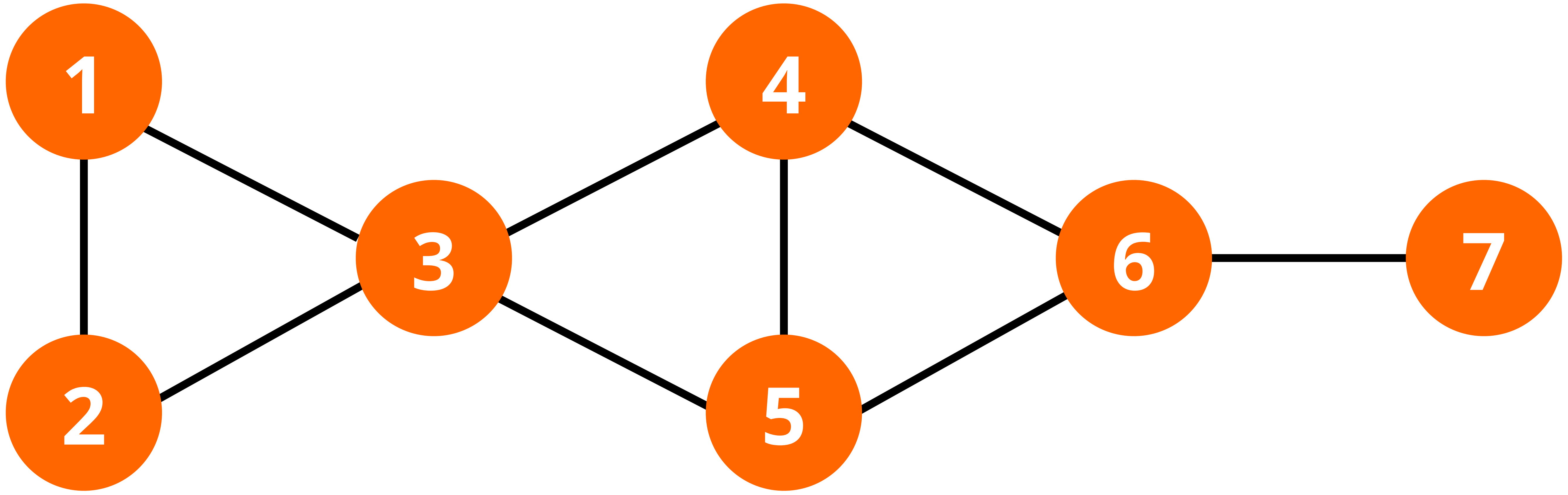


Objective function for optimisation

- Construct a function with bias that favour an optimal solution
- Nodes
  - Favor nodes with more connectors
- Connectors
  - As few as possible

7

Objective function for optimisation



LINEAR

$$-x_1 - x_2 - 3x_3 - 2x_4 - 2x_5 - 2x_6 + 0x_7 +$$

QUADRATIC

$$x_1x_2 + x_1x_3 + x_2x_3 + x_3x_4 + x_3x_5 +$$
$$x_4x_5 + x_4x_6 + x_5x_6 + x_6x_7$$

Objective function for optimisation

$$\begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -2 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

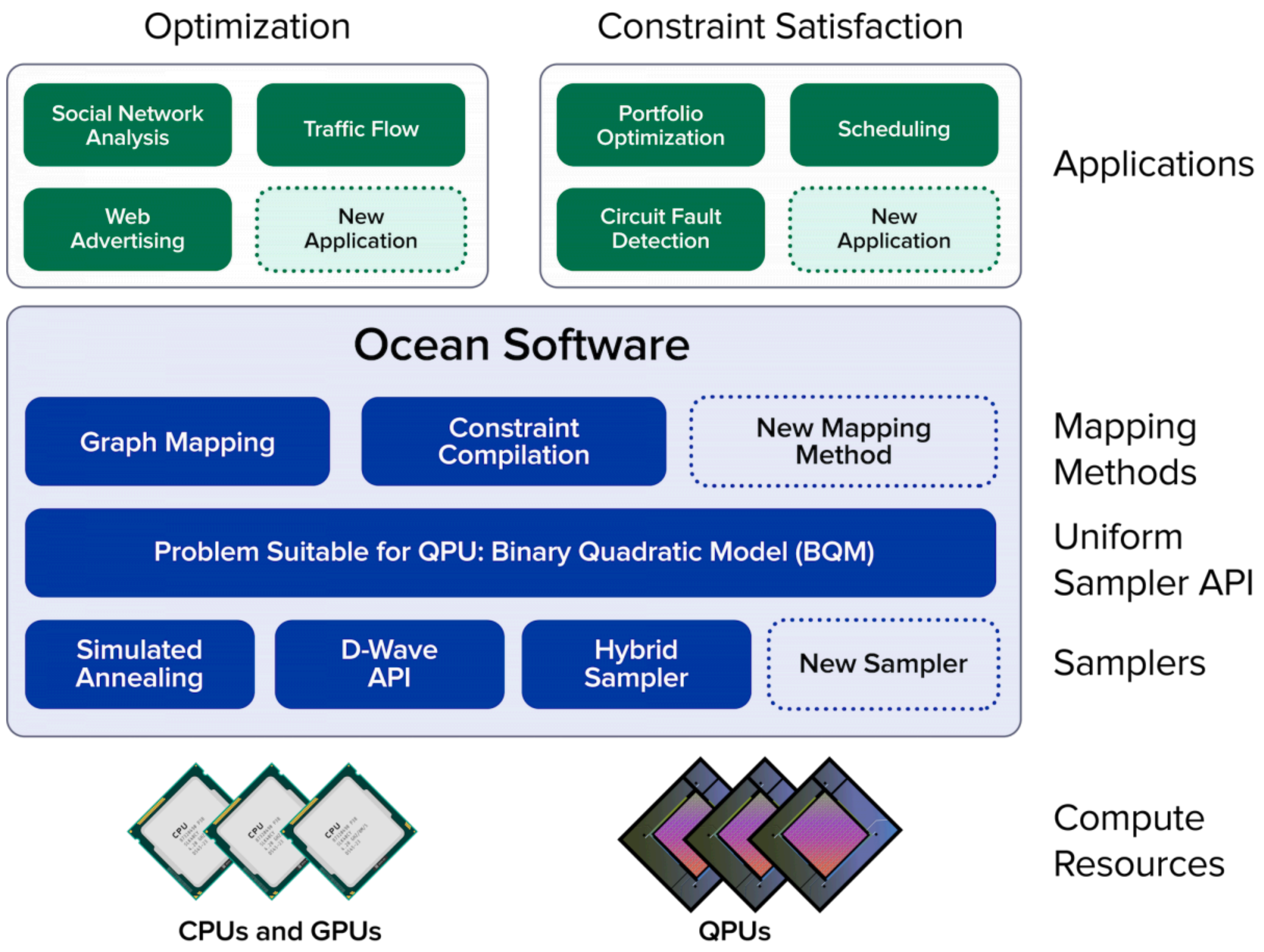
LINEAR

$$-x_1 - x_2 - 3x_3 - 2x_4 - 2x_5 - 2x_6 + 0x_7 +$$

QUADRATIC

$$\left\{ \begin{array}{l} x_1x_2 + x_1x_3 + x_2x_3 + x_3x_4 + x_3x_5 + \\ x_4x_5 + x_4x_6 + x_5x_6 + x_6x_7 \end{array} \right.$$

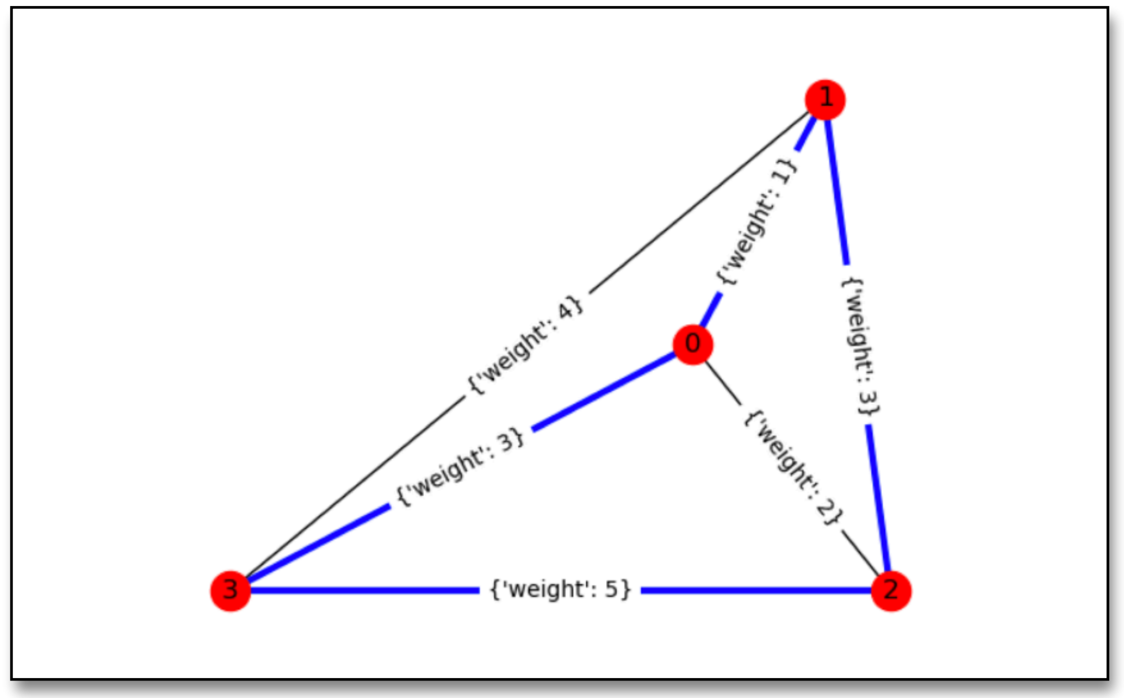
TRIFORK.
...think software

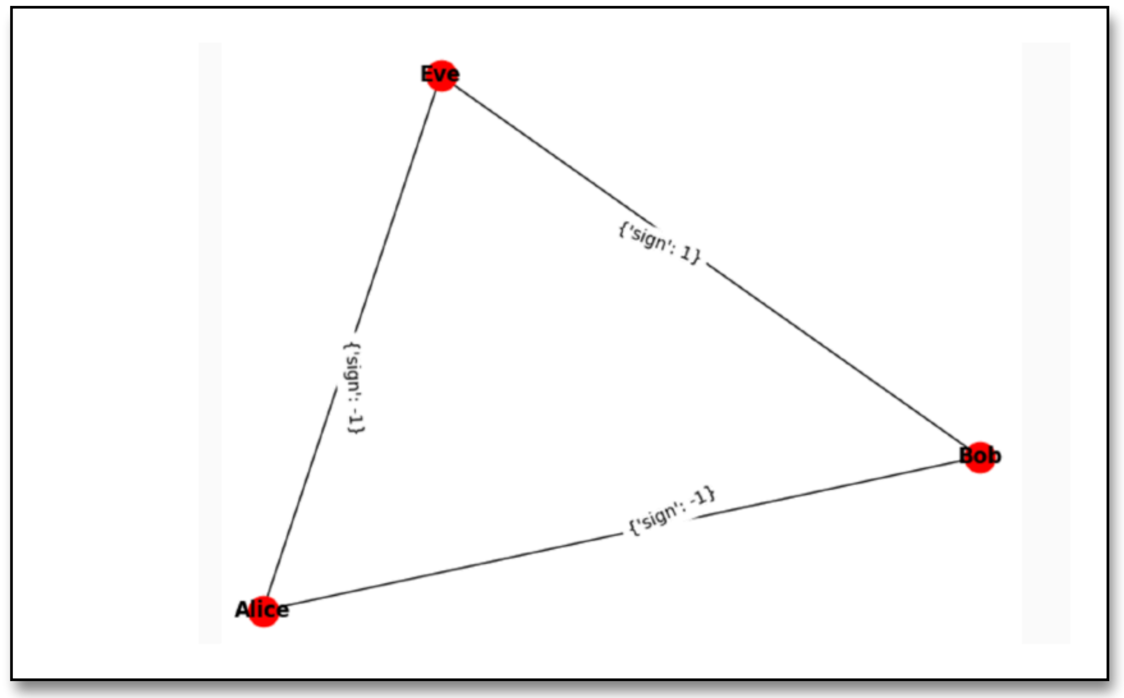


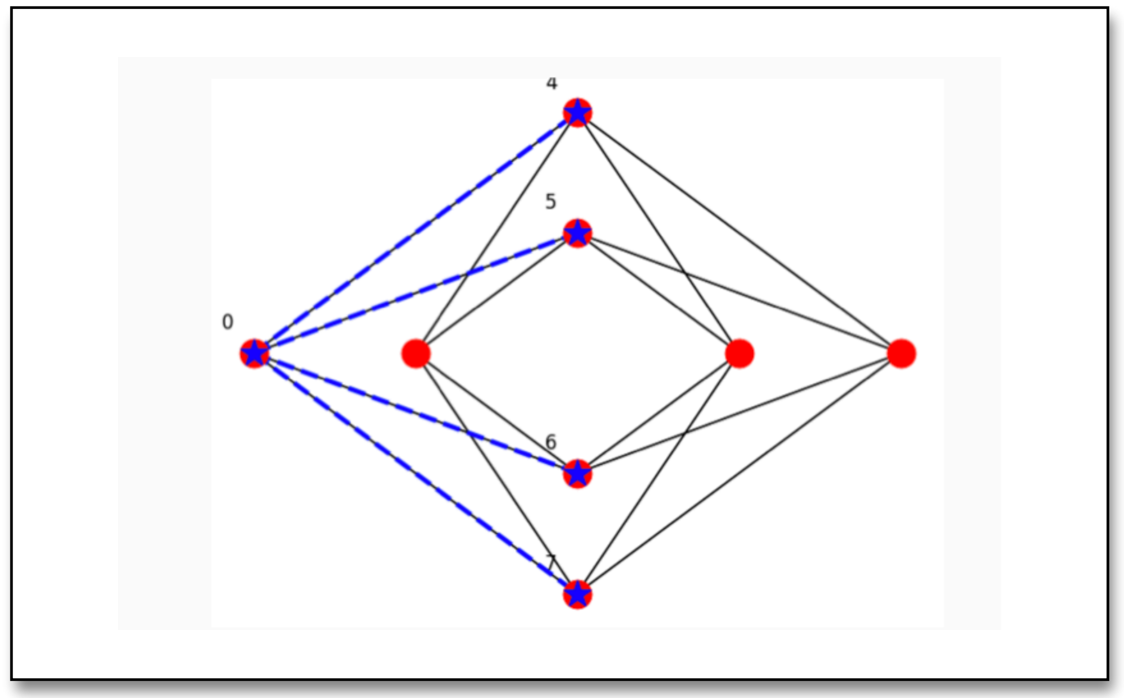
Optimization
Constraint Satisfaction
Social Network Analysis
Traffic Flow
Web Advertising
New Application
Portfolio Optimization
Scheduling
Circuit Fault Detection
New Application
Applications
Ocean Software
Graph Mapping
Constraint Compilation
New Mapping Method
Mapping Methods
Problem Suitable for QPU: Binary Quadratic Model (BQM)
Uniform Sampler API
Simulated Annealing
D-Wave API
Hybrid Sampler
New Sampler
Samplers
CPUs and GPUs
QPUs
Compute Resources

dwave-networkx

- Algorithms
  - Canonicalization
  - Clique
  - Coloring
  - Cover
  - Elimination Ordering
  - Markov Networks
  - Matching
  - Maximum Cut
  - Independent Set
  - Social
  - Traveling Salesperson
- Drawing
  - Chimera Graph Functions Graph Generators
  - D-Wave Systems
  - Other Graphs
- Utilities
  - Decorators
  - Graph Indexing
  - Exceptions
- Default sampler
  - Sampler API
  - Functions







AGENDA

Practical Quantum Annealing - Part 2

Constraint Satisfation Problems Revisited

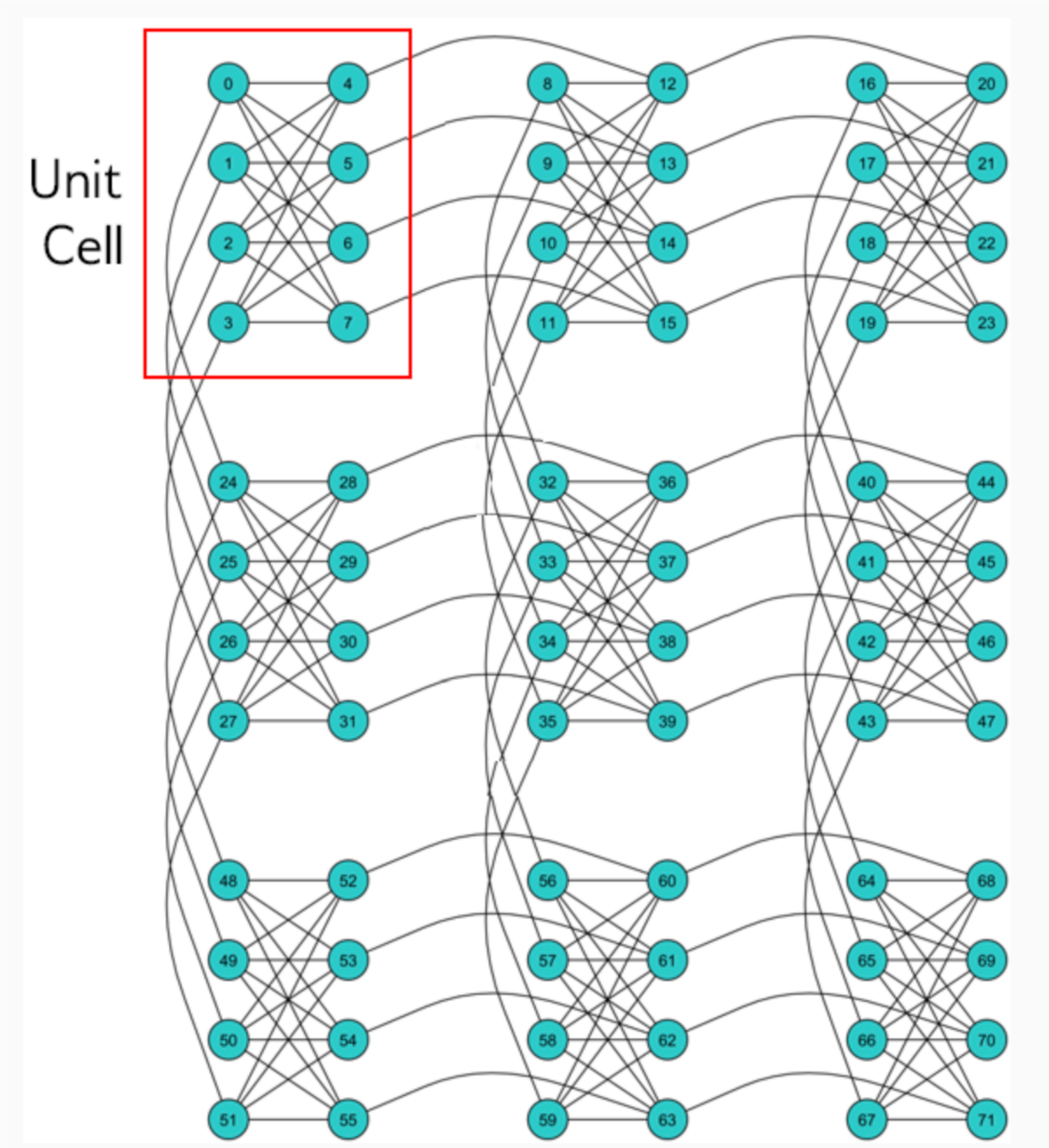
Networks and Network Algorithms

Divide and Conquer - An Introduction

Chimera Architecture

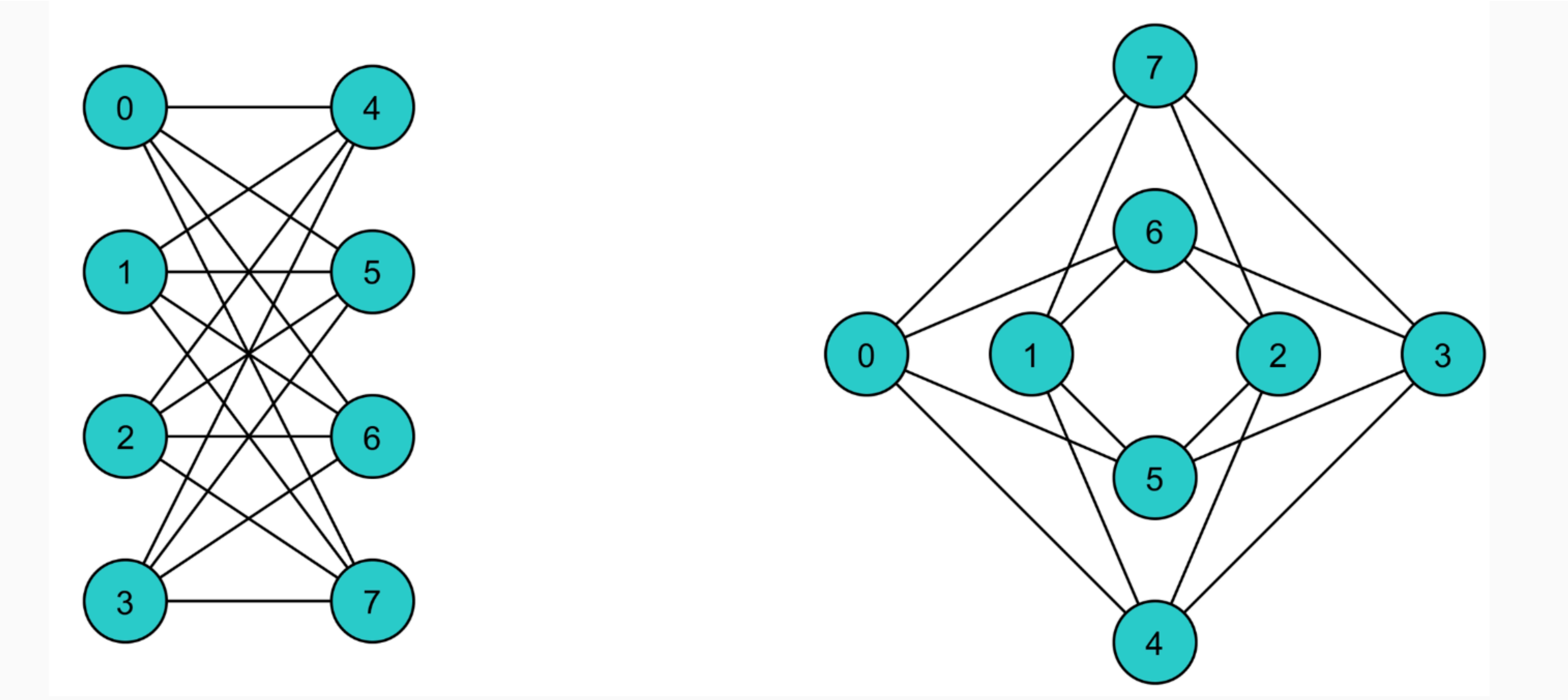
CHIMERA ARCHITECTURE

Unit Cells

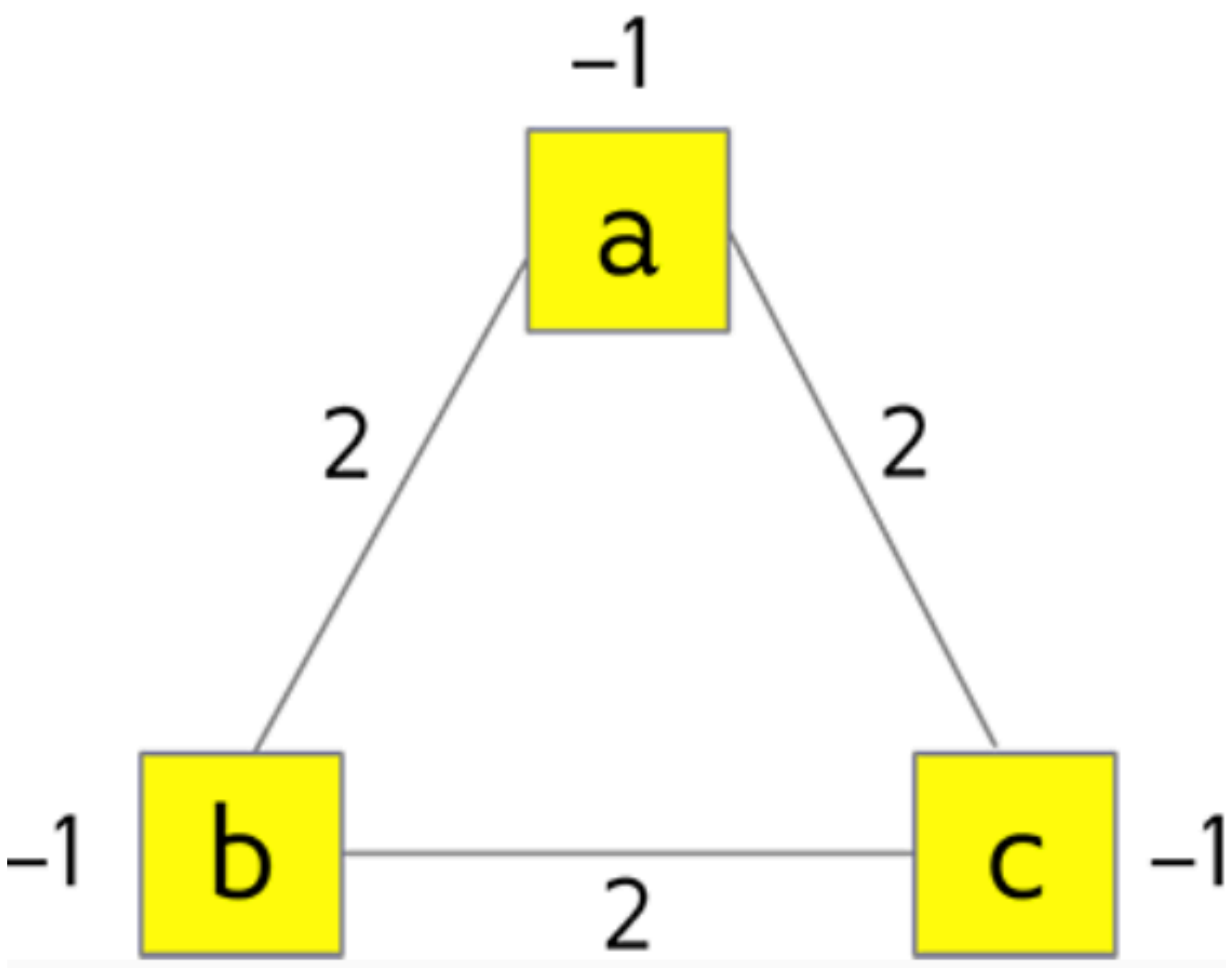


CHIMERA ARCHITECTURE

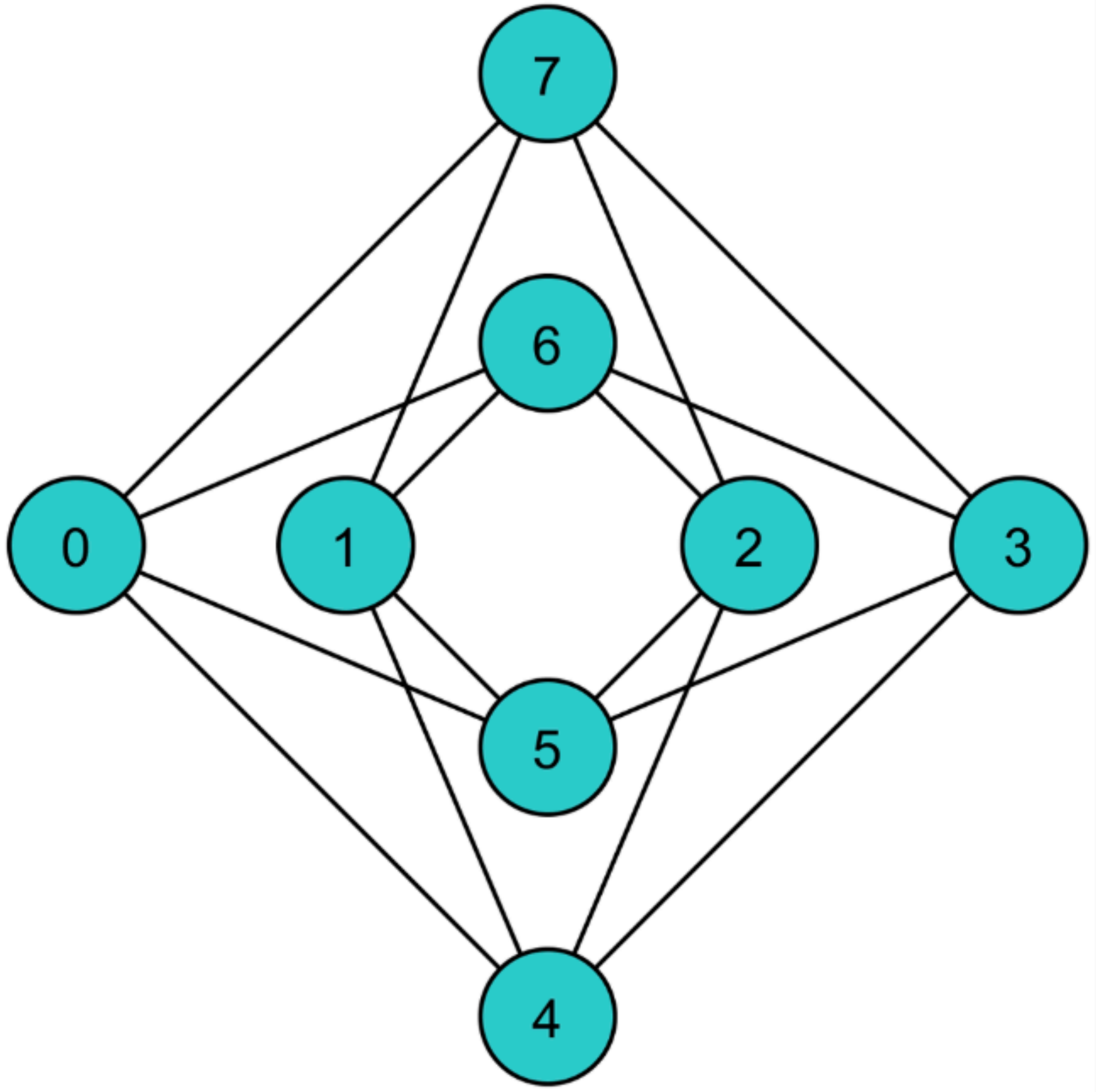
Unit Cells



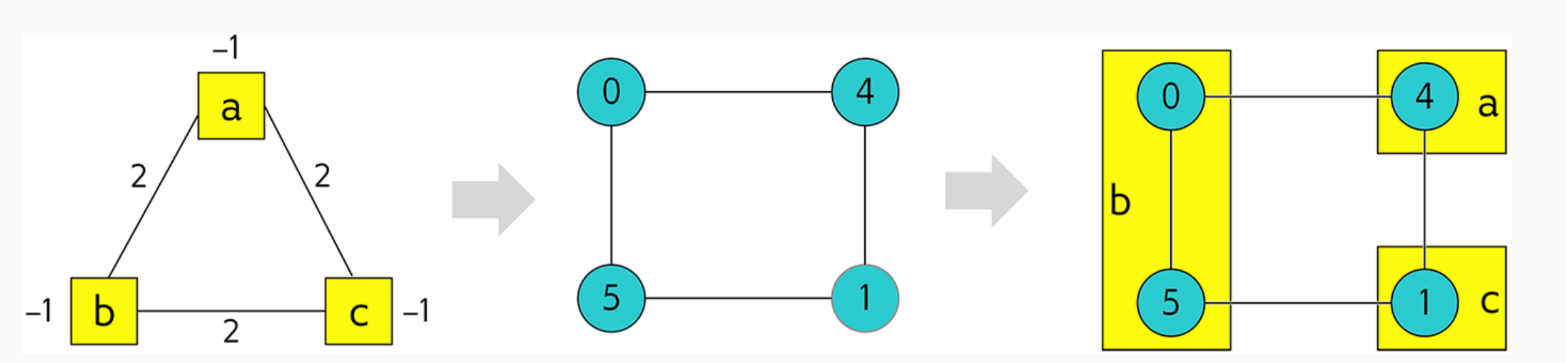
Minor Embedding



?



Minor Embedding



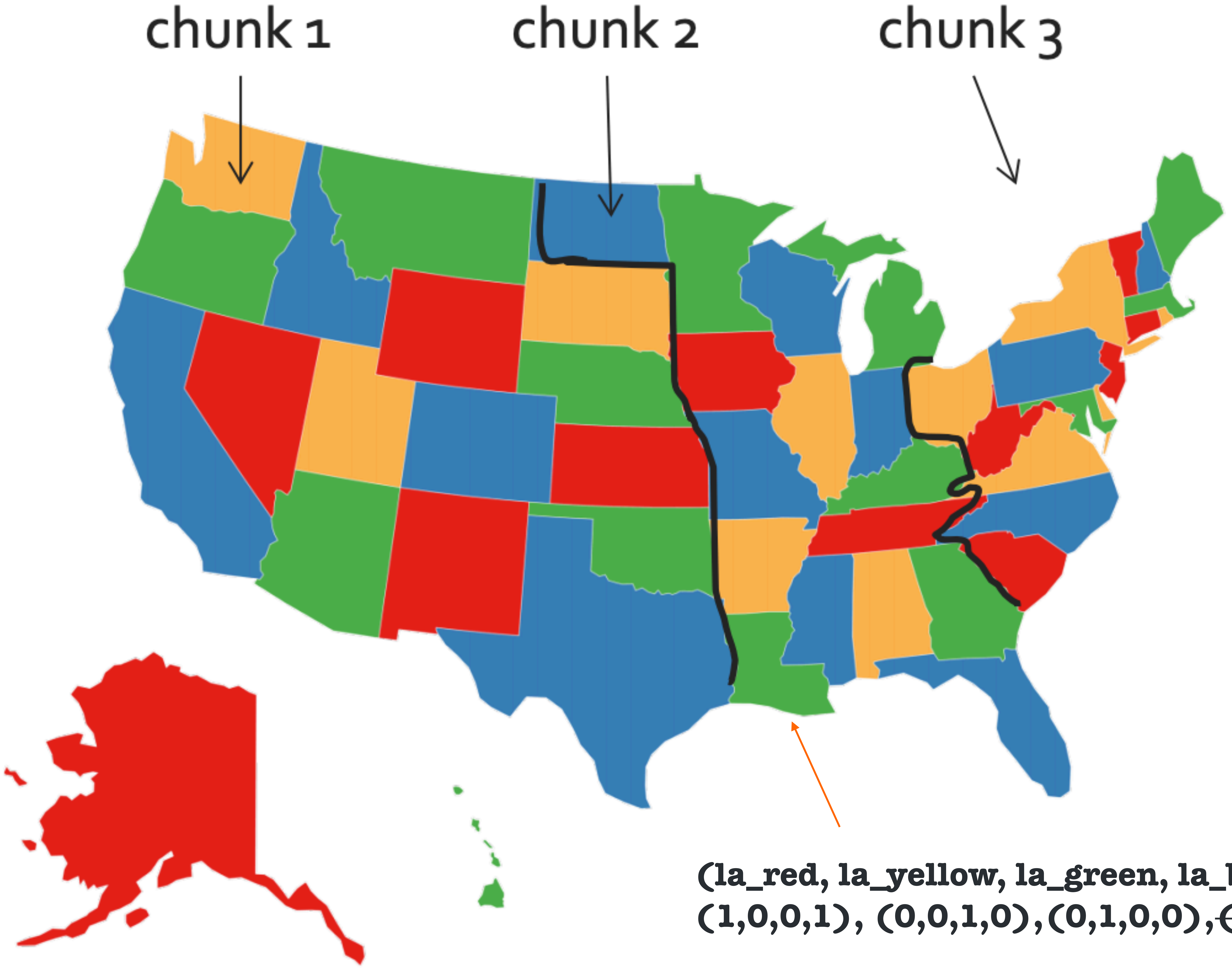
TRIFORK.
...think software

DIVIDING LARGE PROBLEMS

Divide and conquer

- Problems may be too large for the number of qubits and couplers available on the quantum processor
- This can be solved by **dividing** the problem into smaller sections, solving each section individually and stitching the results back together
- See the DWave Problem Solving Handbook for examples of this
    - https://docs.dwavesys.com/docs/latest/doc_handbook.html

TRIFORK.
...think software

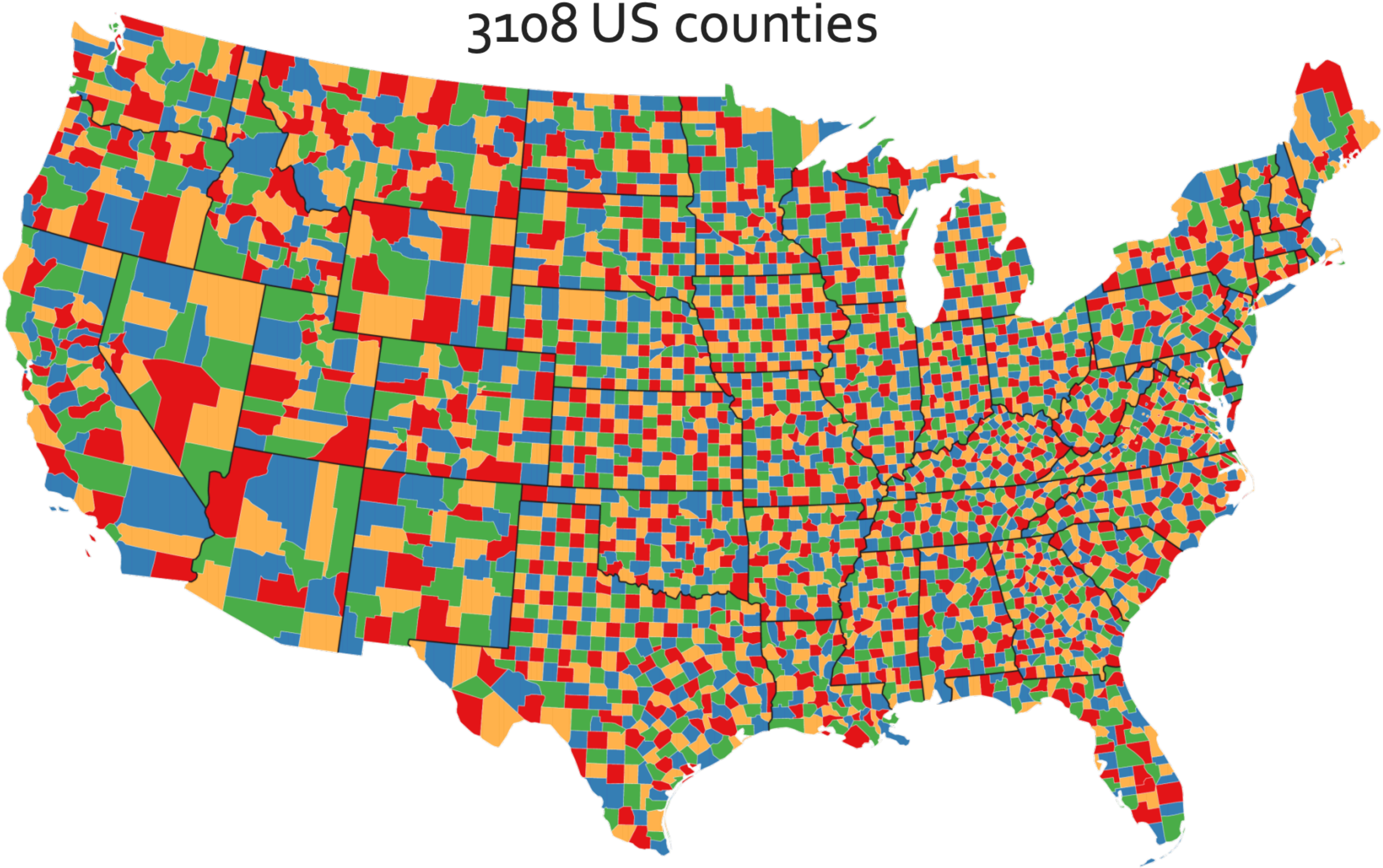


- Calculate chunk 1
- Use colorings as **bias** for calculating chunk 2
- .. repeat

(la_red, la_yellow, la_green, la_blue) can only be **(1,0,0,1), (0,0,1,0),(0,1,0,0),~~(0,0,0,1)~~**

TRIFORK. ...think software

3108 US counties

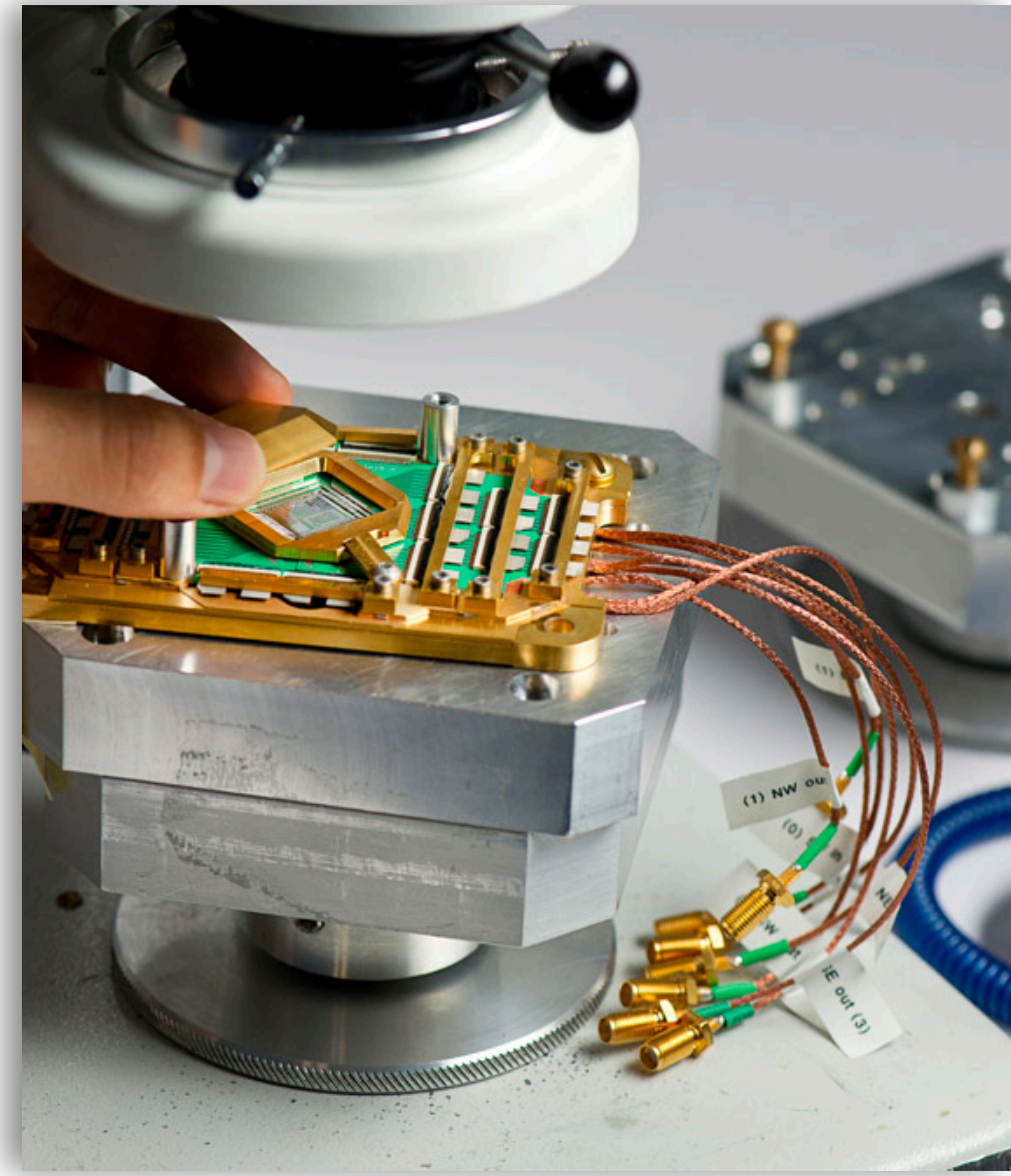


TRIFORK.
...think software

NEXT STEPS

Resources

- Take the Leap !
  - https://www.dwavesys.com/take-leap
- GOTO Masterclasses
  - https://gotocph.com/2020/pages/offseasonmasterclasses



Thank you

NIELS STHEN HANSEN
NSH@TRIFORK.COM

TRIFORK.
...think software