Quantum Annealing

EE599-001 & EE699-010, Spring 2026

Hank Dietz

http://aggregate.org/hankd/

University of Kentucky

Quantum Annealing?

D-Wave has been the dominant player…

	Fujitsu [1]	D-Wave [3]	Toshiba [4]	Hitachi [6]	NEC [7,8]
Processor	Digital annealing unit, Multi-GPU	Quantum processing unit	FPGA, GPU	GPU	General-purpose vector processor
Basic search method/algorithm	MCMC ** Parallel tempering	Quantum annealing	Simulated bifurcation	Momentum annealing	Simulated annealing
Bit/Spin count	100,000 bits (1 million bits [2] by server parallelization)	Abt. 5,000 qubits	1 million bits	100,000 bits	100,000 bits
Degree of connection	Full connection	Sparse connection	Full connection	Full connection	Full connection
Coupling resolution	64-bit resolution	Analog (abt. 5-bit resolution)	2 to 32-bit resolution [5]	Undisclosed	Undisclosed
Constraint processing	**Practical constraint handling** Equality constraint (1way 1hot, 2way 1hot), Inequality constraint, Automatic coefficient adjustment	-	-	-	Equality constraint (1way 1hot)

**** MCMC: Markov-Chain Monte Carlo**

[1] https://www.fujitsu.com/jp/group/labs/en/about/resources/tech/techintro.html
[2] https://pr.fujitsu.com/jp/news/2020/11/9.html
[3] https://www.dwavesys.com/solutions-and-products/systems/
[4] https://www.global.toshiba/jp/products-solutions/ai-iot/sbm.html
[5] https://www.itmedia.co.jp/news/articles/1907/30/news030_3.html
[6] https://www.hitachi.co.jp/rd/news/topics/2019/0830.html
[7] https://jpn.nec.com/nec-vector-annealing-service/index.html?nid=jpntop211051
[8] Takano et al., IEICE Technical Report CAS 2019 -47, MSS 2019 -26

* According to survey by Fujitsu

https://global.fujitsu/-/media/Project/Fujitsu/Fujitsu-HQ/technology/research/quantum/Fujitsu_Quantum_20220803_En.pdf

Is a Quantum Annealer a Quantum Computer?

- Quantum computers use phenomena like entanglement and superposition to perform computations…

  which the D-Wave machines do.

- Quantum annealers are less generally programmable, but can inefficiently emulate quantum circuits, and vice versa

Outside a D-Wave

- 20’x20’x12’
- 12.5kW power
- 4.3 tons cooling

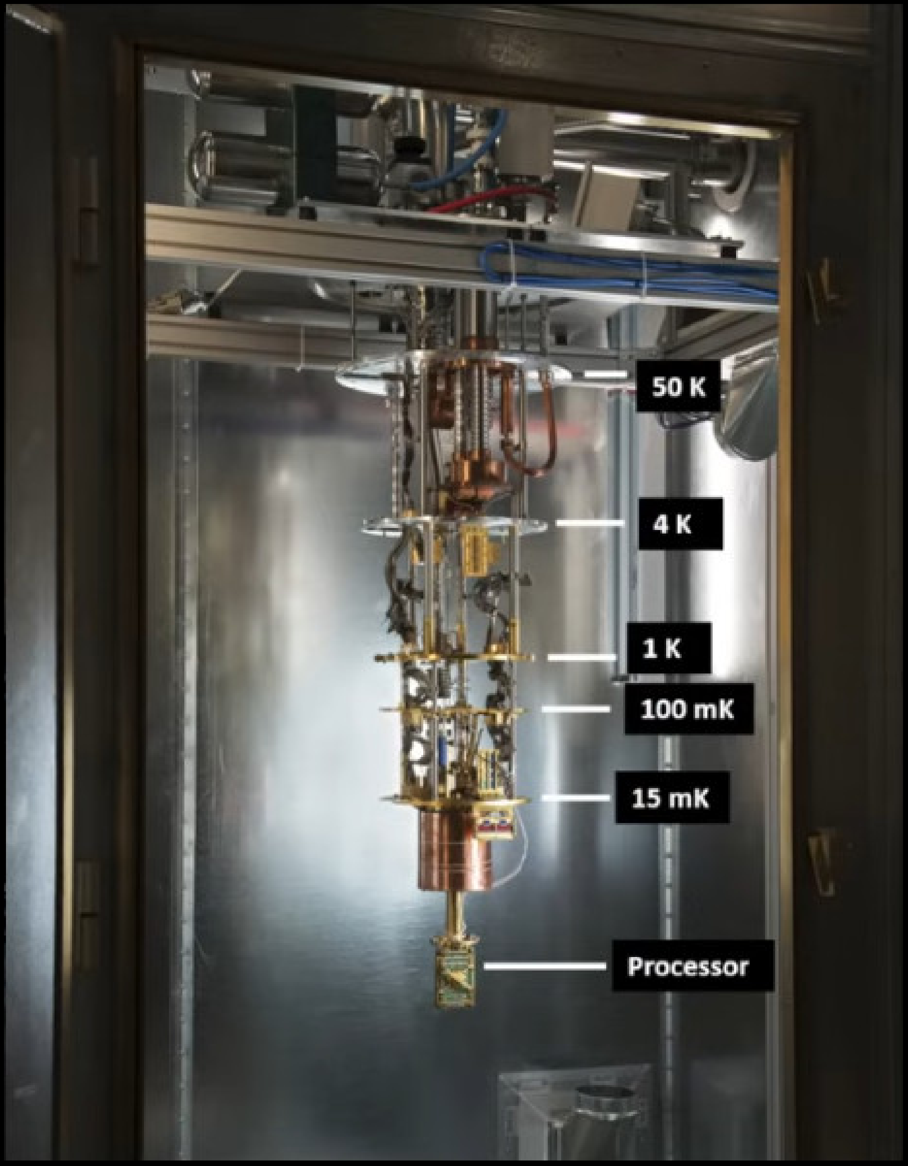
108a Marksbury could power and cool 7 of these!

Inside a D-Wave

- Photo shows inside of cabinet with temps

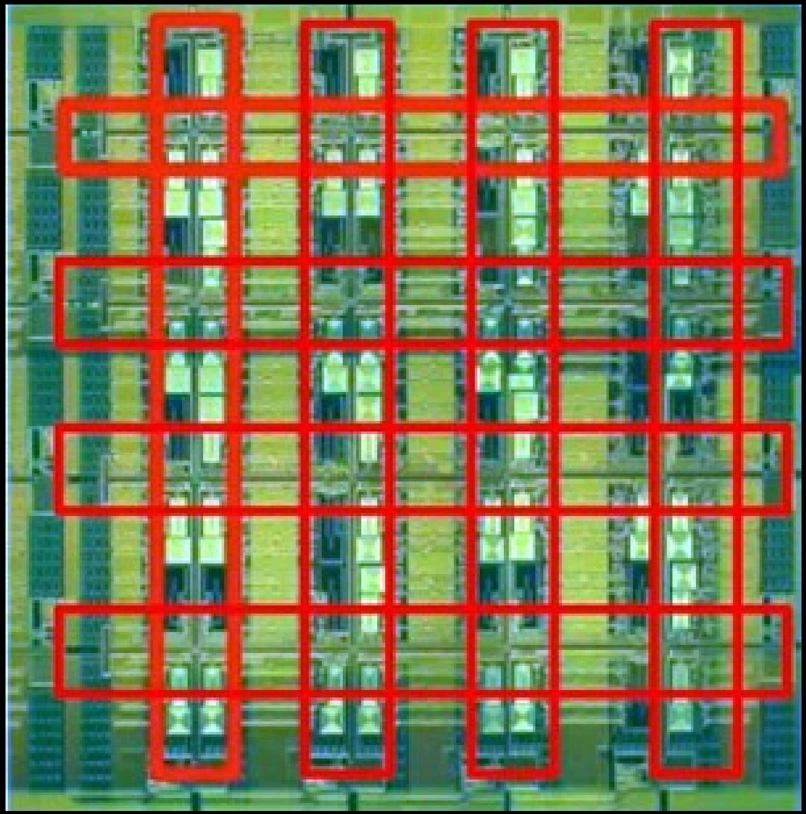
- 175x colder than space
- 50Kx < Earth mag field
- 10Gx < 1 atm pressure
- Low vibration floor

- 4,400+ Qubits
- 20-way connect



Deep Inside a D-Wave

- Photo shows a D-Wave QPU chip
- Each loop implements a qubit (spin up/down)
- Each intersection is a coupler



Ising Model

- Ernst Ising description of energy of a system of magnetic spins. Each spin can be up/down:

$$S_i \in \{-1, +1\}$$

- Energy of the system is:

$$E = -\sum_i h_i S_i + \sum_{ij} J_{ij} S_i S_j$$

- Ising model convertible to QUBO:

$$q_i = \frac{1 + S_i}{2}$$

QUBO

- Quadratic Unconstrained Binary Optimization:
  - Weights a_i, Strengths $b_{i,j}$; no 3-factor values
  - No constraints on variables
  - Each variable is 0 or 1
  - Find variable values to minimize objective

$$Obj(c, a_i, b_{ij}; q_i) = c + \sum_{i} a_i q_i + \sum_{i<j} b_{ij} q_i q_j$$

- Also called:
  Binary Quadratic Model (BQM)

D-Wave Programming Model

QUBIT	q_i	**Quantum bit** which participates in annealing cycle and settles into one of two possible final states: $\{0,1\}$
COUPLER	$q_i q_j$	Physical device that allows one **qubit to influence another qubit**
WEIGHT	a_i	Real-valued **constant associated with each qubit**, which influences the qubit's tendency to collapse into its two possible final states; controlled by the programmer
STRENGTH	b_{ij}	Real-valued **constant associated with each coupler**, which controls the influence exerted by one qubit on another; controlled by the programmer
OBJECTIVE	Obj	Real-valued **function which is minimized** during the annealing cycle

The system samples from the q_i that minimize the objective

$$Obj(a_i, b_{ij}; q_i) = \sum_i a_i q_i + \sum_{ij} b_{ij} q_i q_j$$

What Can QUBO Solve?

- Ising/QUBO decisions are NP-complete, so ∃ polynomial mapping ∀ NP-complete problems
- Mappings for many NP-complete/NP-hard problems have been shown ≤ cubic cost
  https://arxiv.org/abs/1302.5843
- Quantum Annealing may be exponentially faster for QUBOs, but various counterexamples exist

What Can QUBO Solve?

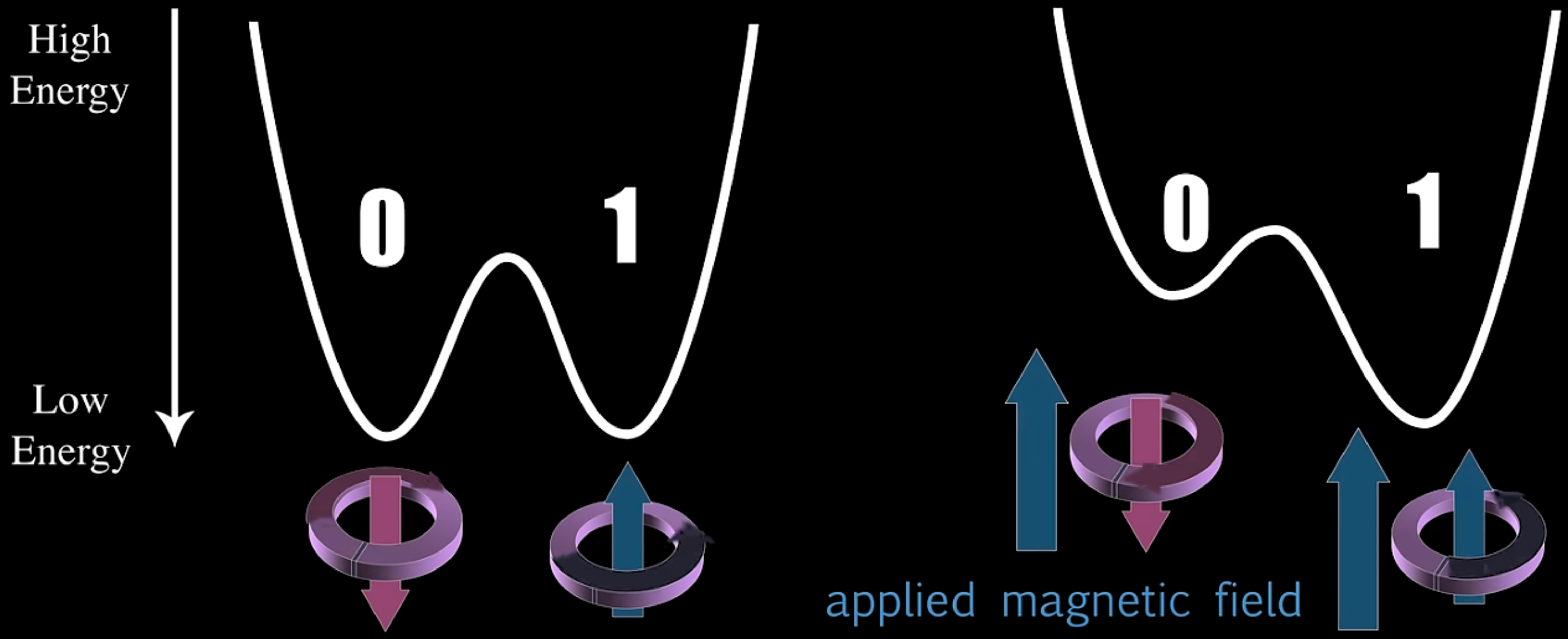
- Various partitioning problems
- Binary integer linear programming
- Various covering and packing problems
- Knapsack problem with integer weights
- Graph coloring
- Hamiltonian cycles and traveling salesman
- Minimal spanning trees
- Graph isomorphism
- Various digital logic gate circuit properties
- ...

How D-Wave Solves a QUBO

- Create the optimization problem matrix
- The a and b terms become voltages, currents, and/or magnetic fields set for the duration
- Qubits initialized to equiprobable superposition
- *Annealing is performed by applying decreasing quantum fluctuations until minimum energy classical state is reached in about 20μs*
- Repeat annealing many times (sampling)

- Reverse annealing: start with a solution
- Fast anneals: leave system in superposition

How D-Wave Anneals



- The applied field comes from a weights
- Couplers entangle, b favoring {00,11} or {01,10}

QUBO Model

- Quadratic Unconstrained Binary Optimization
  - Each variable can have a weight
  - Each pair of variables has a weight

$42x + 3$

$1.2 + 3.4x - 5.6y + 7.8xy$

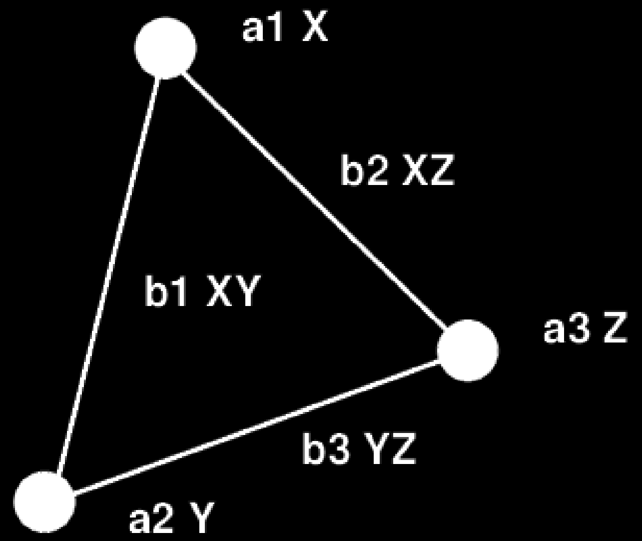
$xy + 2xz - 4yz$

xyz

QUBO Matrix Representation

- Generic 3-qubit example:

$a_1 X + a_2 Y + a_3 Z + b_1 XY + b_2 XZ + b_3 YZ$



a_1	b_1	b_2
	a_2	b_3
		a_3

QUBO Matrix Representation

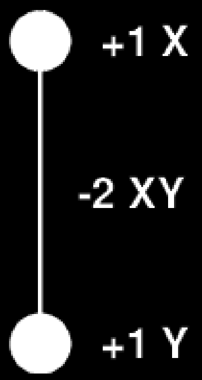
- Find X==Y, i.e., {00, 11}

$a_1 X + a_2 Y + b XY + c$

$1 = a_1 + c;\ 1 = a_2 + c;\ 0 = a_1 + a_2 + b + c;\ c = 0;$

$a_1 = 1;\ a_2 = 1;\ b = -2;\ c = 0;$

X	Y	Obj=a1X+a2Y+bXY + c
0	0	0 (*any* < 1)
0	1	1 (*any* > 0)
1	0	1 (*any* > 0)
1	1	0 (*any* < 1)



1	-2
	1

QUBO Matrix Representation

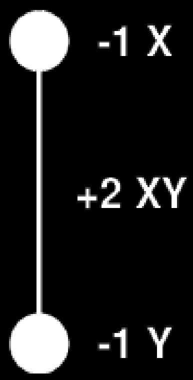
- Find X!=Y, i.e., {01, 10}

$a_1 X + a_2 Y + b XY + c$

$0 = a_1 + c$; $0 = a_2 + c$; $1 = a_1 + a_2 + b + c$; $c = 1$;

$a_1 = -1$; $a_2 = -1$; $b = 2$; $c = 1$;

X	Y	Obj=a1X+a2Y+bXY + c
0	0	1
0	1	0
1	0	0
1	1	1



-1	2
	-1

A 4-Qubit QUBO

- Find (X==Y) & (W==Z) & (Y!=Z), consider these separately:

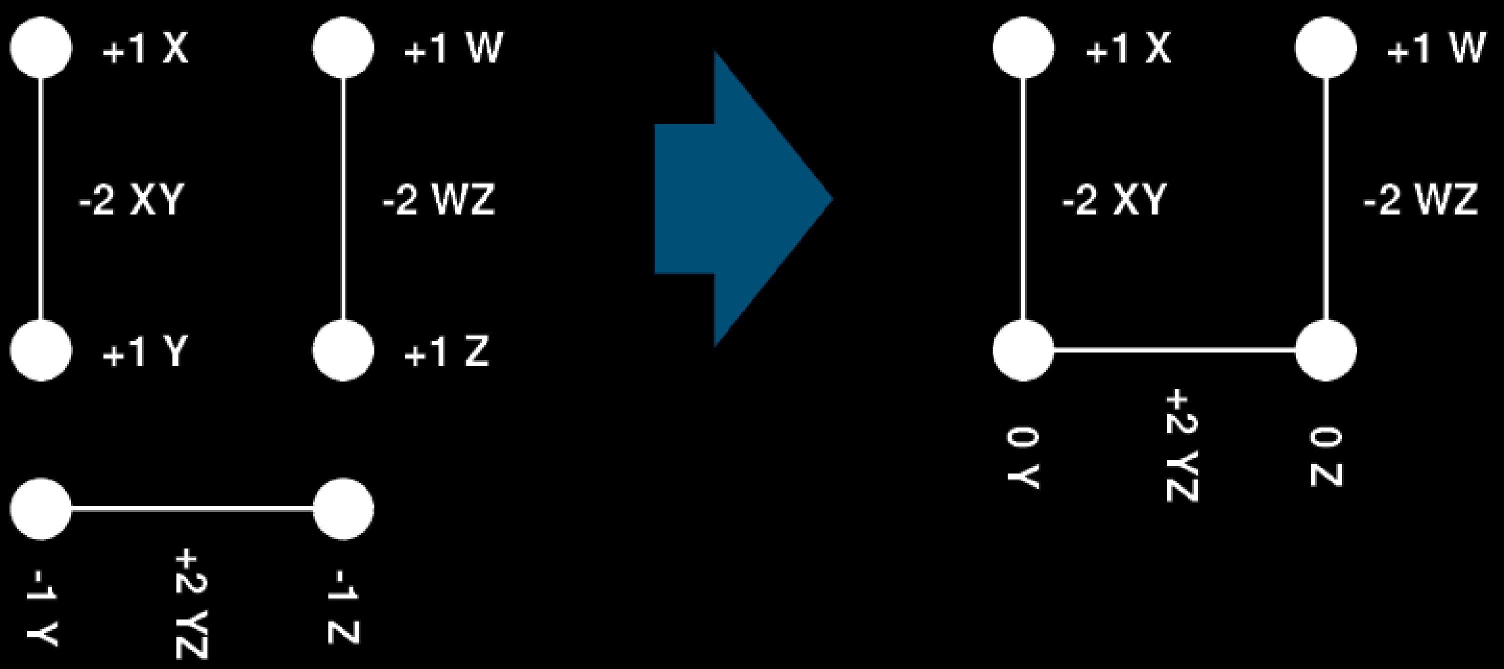
X	Y	Obj=a1X+a2Y+bXY + c
0	0	-1
0	1	0
1	0	0
1	1	-1

W	Z	Obj=a1W+a2Z+bWX + c
0	0	-1
0	1	0
1	0	0
1	1	-1

Y	Z	Obj=a1Y+a2Z+bYZ + c
0	0	1
0	1	0
1	0	0
1	1	1

A 4-Qubit QUBO

- Linear equations add, so:



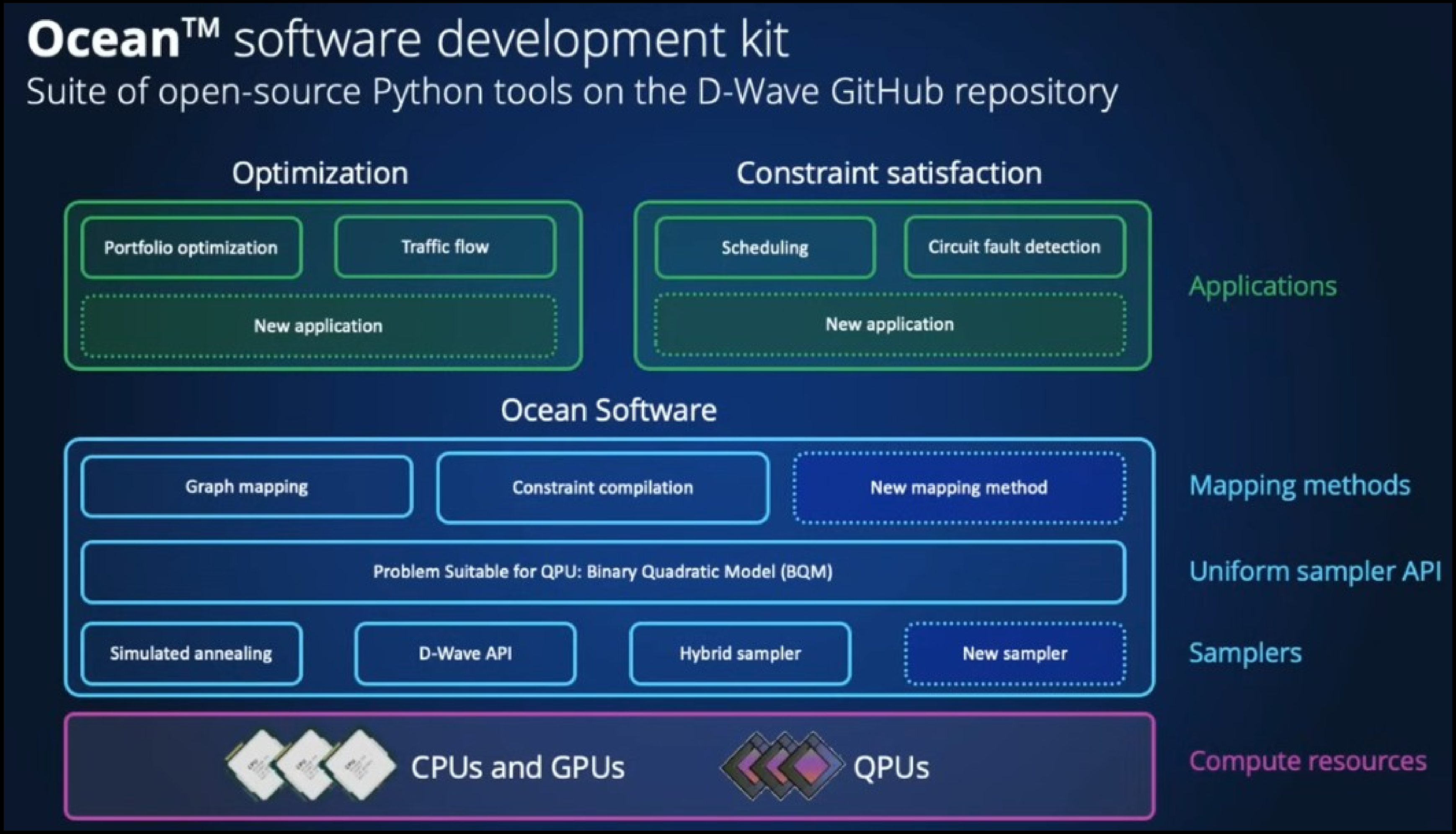
Objective = -2 + X + W – 2XY – 2WZ + 2YZ

A 4-Qubit QUBO Results

- Find (X==Y) & (W==Z) & (Y!=Z) gives two possible XYWZ solutions: 0011 and 1100
- Which one do I get?
  - Either is one is a bit less than 50% likely
  - Over many samplings, you'll see both
- One annealing run with sampling takes about 20μs, so 200 only take about 4ms

D-Wave Programming

- Yet another Python SDK: **Ocean**



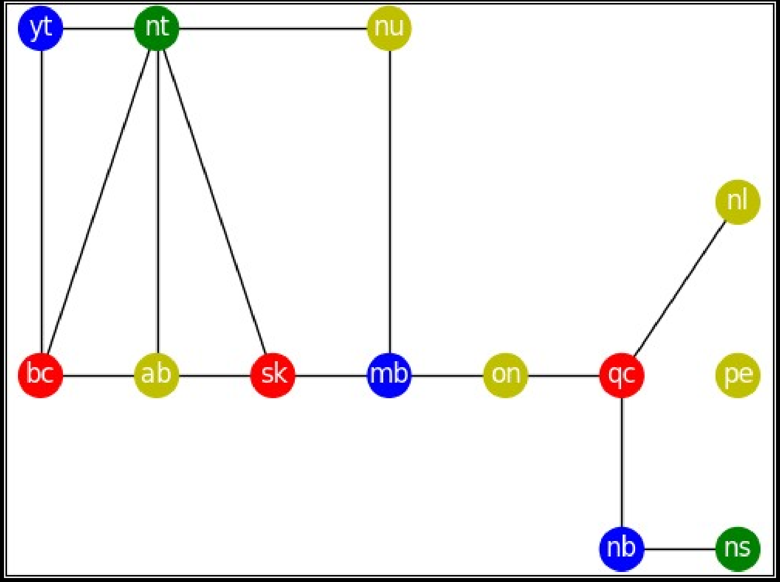
D-Wave Programming

- Ocean example apps

  `https://github.com/orgs/dwave-examples`

- One interesting example: 4-color maps

  `https://github.com/dwave-examples/map-coloring`



D-Wave Cloud Access

- Another free-for-a-minute-in-first-month deal

QAOA?

- Quantum Approximate Optimization Algorithm (QAOA) solves QUBO as a Quantum Circuit
  - Maps QUBO into an Ising
  - Then maps that into a quantum circuit

  https://tensorcircuit.readthedocs.io/en/latest/tutorials/qubo_problem.html

- Other examples using QAOA

  https://tensorcircuit.readthedocs.io/en/latest/tutorials/qaoa.html

QUBOs Without Quantum

- Lots of methods, some often fast & robust
  - Exact solvers using branch-and-bound, etc.
  - Simulated annealing
  - Tabu search
  - Genetic Algorithm
  - From my old Evolutionary Computing course: `https://aggregate.org/EC/DUMEC/`

- MATLAB has support for QUBO

`https://www.mathworks.com/help/matlab/math/quantum-annealing-workflow.html`

Factoring by QUBO

- Not hard to factor a semiprime by QUBO
  - Break multiply down to gate QUBOs
  - Multiply all appropriately-sized values (this will take many extra variables)
  - Result constrained to the semiprime
  - Minimize energy by the QUBO matrix

- I used ChatGPT to make a C++ version!

  `https://aggregate.org/QC/qubo_factor.cpp`

Factoring by QUBO

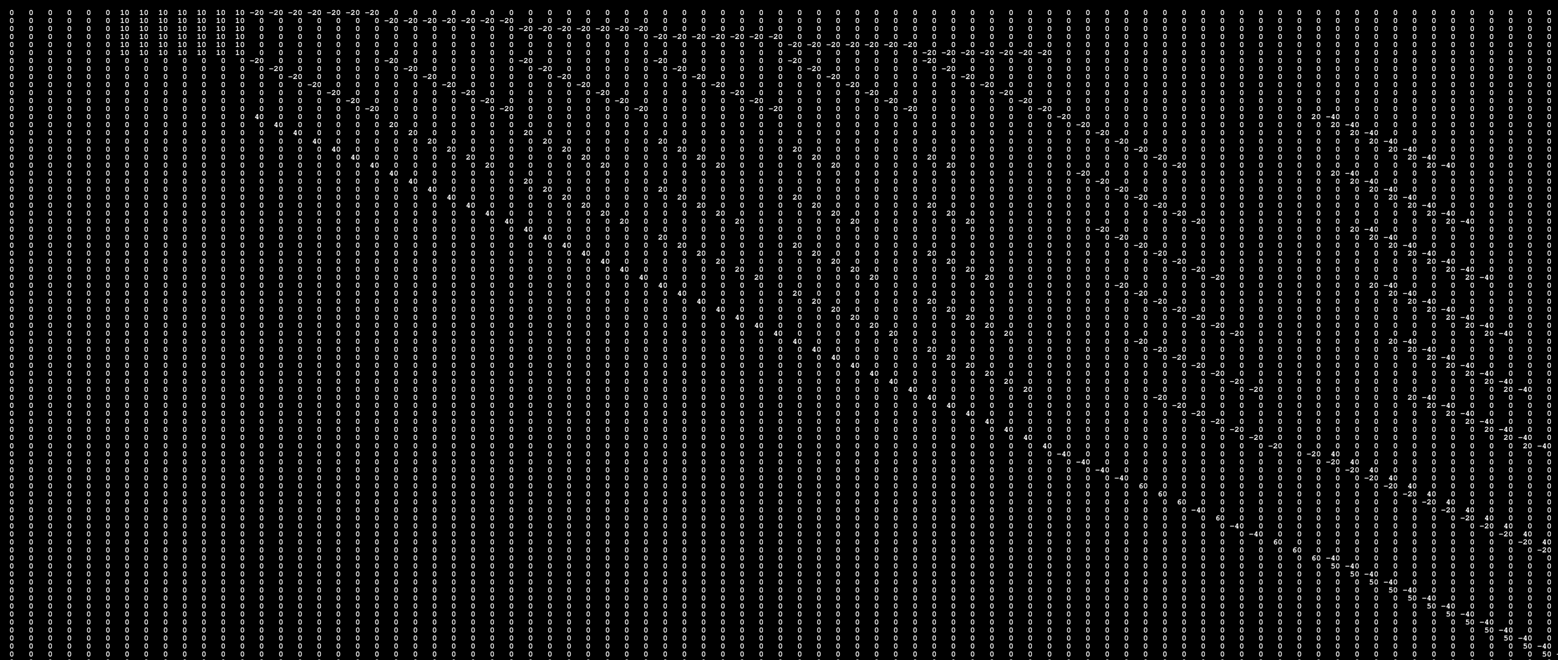
- Here's the QUBO matrix for factoring 15:

0	0	0	10	10	10	−20	−20	−20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	0	0	0	−20	−20	−20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	0	0	0	0	0	0	−20	−20	−20	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	−20	0	0	−20	0	0	−20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	−20	0	0	−20	0	0	−20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	−20	0	0	−20	0	0	−20	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	−20	0	0	0	0	0	20	−40	0	0	0	0	0
0	0	0	0	0	0	0	40	0	20	0	0	0	0	0	0	−20	0	0	0	0	0	20	−40	0	0	0	0
0	0	0	0	0	0	0	0	40	0	20	0	20	0	0	0	0	−20	0	0	0	0	0	20	−40	0	0	0
0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	−20	0	0	0	0	0	20	−40	0	0	0	0
0	0	0	0	0	0	0	0	0	0	40	0	20	0	0	0	0	−20	0	0	0	0	0	20	−40	0	0	0
0	0	0	0	0	0	0	0	0	0	0	40	0	20	0	0	0	0	−20	0	0	0	0	0	20	−40	0	0
0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	−20	0	0	0	0	0	20	−40	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	−20	0	0	0	0	0	20	−40	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	−20	0	0	0	0	0	20	−40	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−40	0	0	0	0	0	−20	40	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−40	0	0	0	0	0	−20	40	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−40	0	0	0	0	0	−20	40	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−40	0	0	0	0	0	−20	40	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	−20	40	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	−20	40
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	−40	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	−40	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	−40	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	−40	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	−40	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	−40
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40

- Set A=3; B=3; N=15; and potential factors are 3 bits each, but this has 28 qubits, not just 6

Factoring by QUBO

- Here's the QUBO matrix for factoring 1679:



- Potential factors are 6 and 7 bits, but this has 86 qubits, not just 13… and is usually wrong