PlanQK

Optimization using Quantum Annealers

PlanQK Webinar, 07.05.2020

Christoph Roch
LMU München

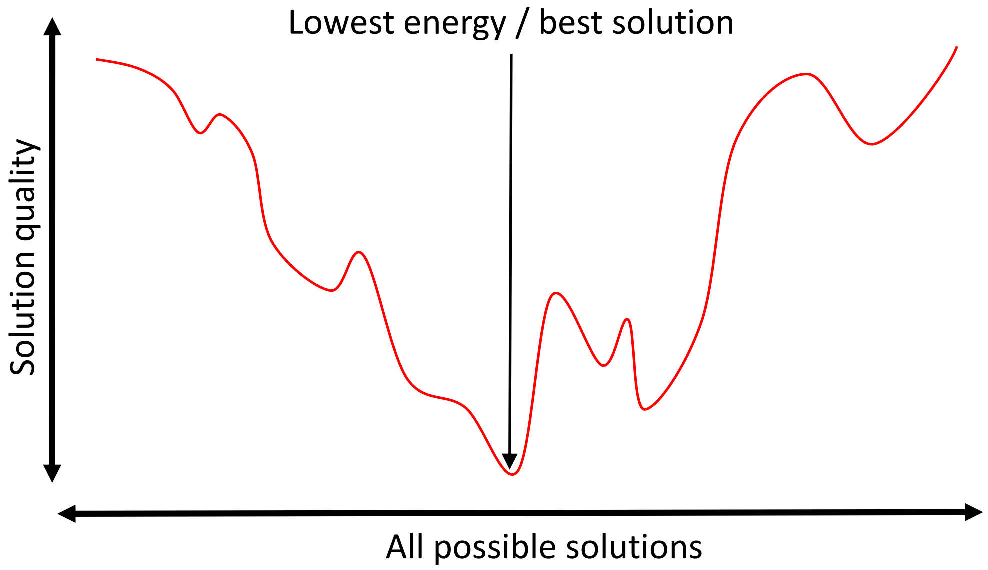
Outline

- Quantum Annealing
- Ising Model and Quadratic Unconstrained Binary Optimization (QUBO)
- QUBO Example: Quadratic Assignment Problem (QAP)
- D-Wave architecture
- Hands-On: Implementing the QAP for the D-Wave Hardware

Later: Fujitsu's Digital Annealer

Quantum Annealing (QA)

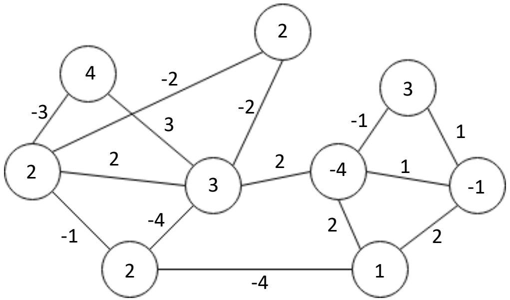
A heuristic approach that solves combinatorial optimization problems



Ising model

Mathematical model for the investigation of phase transitions and other properties of physical systems that evolve over time

$$H(s) = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j$$



Arrangement of n particles on the node of a graph

- Particles can be in the state +1 or -1 (so-called Spins)
- External forces h_i apply to individual particles
- Interaction forces J_{ij} are valid for adjacent particles
- The energy of the spin configuration $s = s_1, \dots, s_n$ is defined through $H(s)$

→ Application of Quantum Annealing to achieve the lowest energy / finding optimal solutions

Quadratic Unconstrained Binary Optimization (QUBO)

Ising Model

$$H(s) = \sum_{i} h_i s_i + \sum_{i<j} J_{ij} s_i s_j$$

QUBO

$$x^t Q x: \;\; x \in \{0, 1\}^n$$

D-Wave`s quantum annealer

D-Wave’s processor chip solves problem instances in form of the Ising model (or QUBO) through the physical implementation of quantum annealing.

- Mounted in a protected „dilution refrigerator“ ($< 20mK$)
- Conventional Cloud-Computing-Frontend (Leap 2)
- Hardware that does nothing but:
  - Obtains real values for weights h_i and J_{ij}
  - Finds spin assignment s, which minimizes the objective function:

$$H(s) = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j$$

Spoiler: Fujitsu’s digital annealer also takes QUBO problems as input. Later more…

Quadratic Assignment Problem (QAP)

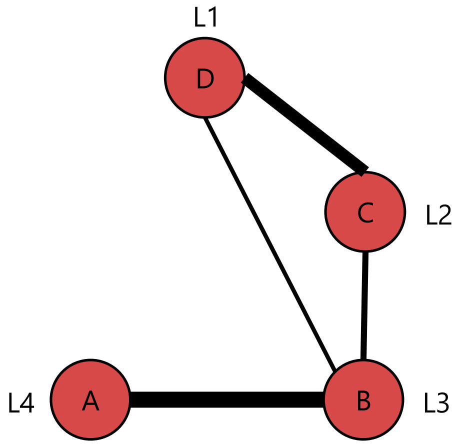
- The goal of the Quadratic Assignment Problem (QAP) is to assign N facilities to N locations so that assignment costs are minimized.
- The assignment costs are defined by a function that multiplies and summarizes the flow/demand (f) and distance (d) for each assignment
- Generalizes also other well known optimization problems like Gate Assignment Problem, Scheduling Problems

Example: Four locations and four facilities

One possible assignment of facilities to locations is:

- Facility D to location 1
- Facility C to location 2
- Facility B to location 3
- Facility A to location 4

The thickness of the line increases with the value of the flow (f).



QUBO: Part 1

$$H = P\sum_{v=1}^{n}\left(1-\sum_{j=1}^{N}x_{v,j}\right)^2 + P\sum_{j=1}^{n}\left(1-\sum_{v=1}^{N}x_{v,j}\right)^2$$

Constraint:
Each facility v is assigned exactly one location

Constraint:
Each location j is assigned exactly one facility

QUBO: Part 2

$$+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\left(f_{i,j} * d_{k,l}\right) x_{i,k} x_{j,l}$$

Minimization/Optimization:
f : flow between facility i und j
d: distance between location k und l

PlanQK

Excursus: Idea of Penalty Value

- Recap: $$P\sum_{v=1}^{n}\left(1-\sum_{j=1}^{n}x_{v,j}\right)^2$$
- Suppose we have n = 4 facilities and we look at facility 1.
  So we set v = 1.
- Let's also assume facility 1 is never assigned to a location. That is, $x_{(1,j)}$ = 0 applies to all j.

$$P\left(1-\sum_{j=1}^{4}x_{1,j}\right)^2 = P\left(1-\left(x_{1,1}+x_{1,2}+x_{1,3}+x_{1,4}\right)\right)^2$$

$$= P\left(1-(0+0+0+0)\right)^2 = P(1)^2 = P$$

Excursus: Idea of Penalty Value

- Recap: $$P\sum_{v=1}^{n}\left(1-\sum_{j=1}^{n}x_{v,j}\right)^2$$

- Let us now assume that facility 1 is assigned to exactly one location, namely location 1, i.e. $x_{(1,1)} = 1$

$$P\left(1-\sum_{j=1}^{4}x_{1,j}\right)^2 = P\left(1-(x_{1,1}+x_{1,2}+x_{1,3}+x_{1,4})\right)^2$$

$$= P\left(1-(1+0+0+0)\right)^2 = P(0)^2 = 0$$

Excursus: Idea of Penalty Value

- Recap: $$P \sum_{v=1}^{n} \left(1 - \sum_{j=1}^{n} x_{v,j}\right)^2$$

- Let's finally assume that facility 1 is assigned to two locations, namely locations 1 and 3.

$$P \left(1 - \sum_{j=1}^{4} x_{1,j}\right)^2 = P\left(1 - \left(x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4}\right)\right)^2$$

$$= P\left(1 - (1 + 0 + 1 + 0)\right)^2 = P(-1)^2 = P$$

How to implement those mathematical constraints and objective functions?

x^*

$x^{t}*$

	A1	A2	A3	A4	B1	B2	B3	B4	C1	C2	C3	C4	D1	D2	D3	D4
A1																
A2																
A3																
A4																
B1																
B2																
B3																
B4																
C1																
C2																
C3																
C4																
D1																
D2																
D3																
D4																

General QUBO formula: $x^t Q x : \; x \in \{0, 1\}^n$

Determing important parameters

- **What is the logical size of the QUBO matrix Q?**
  A spin variable can be considered as an assignment of facility to location (e.g. A1, A2,....) → N^2 Spins → Size of the matrix is N^2 with N equal to the number of facilities/locations

- **How to choose the penalty value P?**
  Consider worst case: Maximum flow (f) between two facilities whose assigned locations have the greatest distance (d) from each other → Penalty value P:

$$P > d_{max} * f_{max}$$

How to get QUBO coefficients?

- Recap: $$P\sum_{v=1}^{n}\left(1-\sum_{j=1}^{n}x_{v,j}\right)^2 \rightarrow P\sum_{v=1}^{4}\left(1-\sum_{j=1}^{4}x_{v,j}\right)^2$$

$$\boxed{P\,((-x(1,1)-x(1,2)-x(1,3)-x(1,4)+1)^2}$$
$$+(-x(2,1)-x(2,2)-x(2,3)-x(2,4)+1)^2$$
$$+(-x(3,1)-x(3,2)-x(3,3)-x(3,4)+1)^2$$
$$+(-x(4,1)-x(4,2)-x(4,3)-x(4,4)+1)^2)$$

$$P(-2\,x(1,1)+x(1,1)^2-2\,x(1,2)+2\,x(1,1)x(1,2)+x(1,2)^2-2\,x(1,3)+2\,x(1,1)x(1,3)$$
$$+2\,x(1,2)x(1,3)+x(1,3)^2-2\,x(1,4)+2\,x(1,1)\,x(1,4)+2\,x(1,2)\,x(1,4)$$
$$+2\,x(1,3)\,x(1,4)+x(1,4)^2+1)$$

	1,1	1,2	1,3	1,4
1,1	- 1*P	2*P	2*P	2*P
1,2		- 1*P	2*P	2*P
1,3			-1*P	2*P
1,4				- 1*P

x*

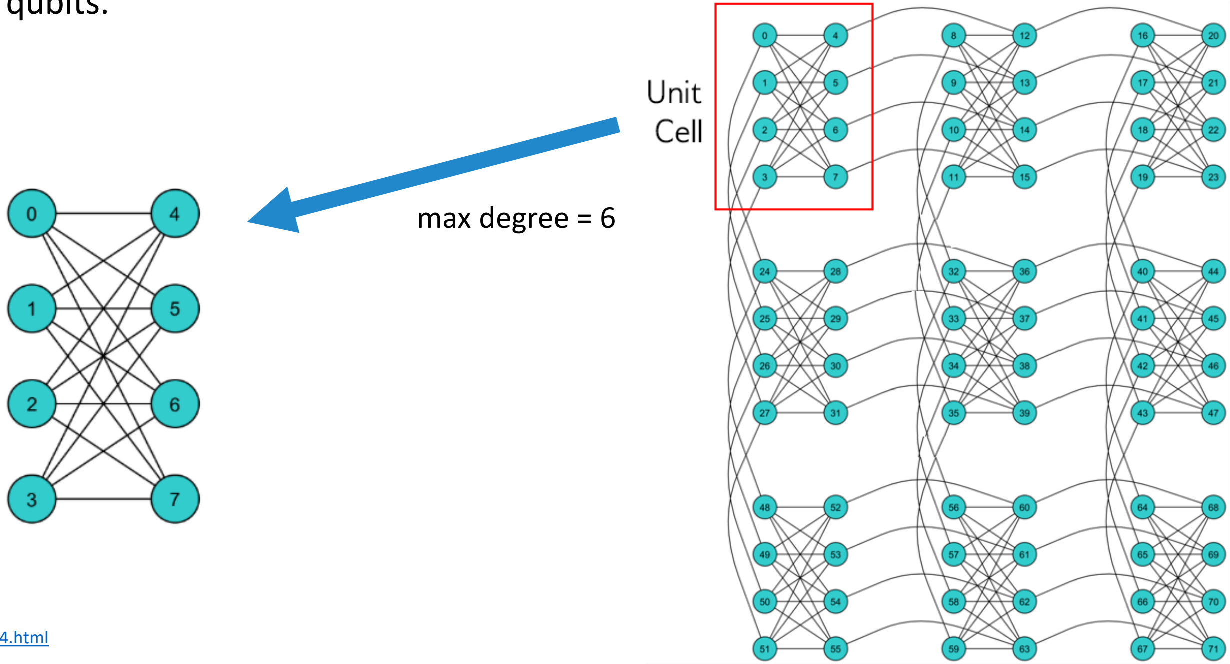
x^t *

	A1	A2	A3	A4	B1	B2	B3	B4	C1	C2	C3	C4	D1	D2	D3	D4
A1	- 2P	2*P	2*P	2*P	2*P	d*f	d*f	d*f	2*P	d*f	d*f	d*f	2*P	d*f	d*f	d*f
A2		- 2P	2*P	2*P	d*f	2*P	d*f	d*f	d*f	2*P	d*f	d*f	d*f	2*P	d*f	d*f
A3			-2P	2*P	d*f	d*f	2*P	d*f	d*f	d*f	2*P	d*f	d*f	d*f	2*P	d*f
A4				- 2P	d*f	d*f	d*f	2*P	d*f	d*f	d*f	2*P	d*f	d*f	d*f	2*P
B1					- 2P	2*P	2*P	2*P	2*P	d*f	d*f	d*f	2*P	d*f	d*f	d*f
B2						- 2P	2*P	2*P	d*f	2*P	d*f	d*f	d*f	2*P	d*f	d*f
B3							- 2P	2*P	d*f	d*f	2*P	d*f	d*f	d*f	2*P	d*f
B4								- 2P	d*f	d*f	d*f	2*P	d*f	d*f	d*f	2*P
C1									- 2P	2*P	2*P	2*P	2*P	d*f	d*f	d*f
C2										-2 P	2*P	2*P	d*f	2*P	d*f	d*f
C3											- 2P	2*P	d*f	d*f	2*P	d*f
C4												- 2P	d*f	d*f	d*f	2*P
D1													- 2P	2*P	2*P	2*P
D2														- 2P	2*P	2*P
D3															- 2P	2*P
D4																- 2P

General QUBO formula: $x^t Qx: \; x \in \{0, 1\}^n$

D-Wave architecture: Chimera Graph

The D-Wave 2000Q QPU supports a C16 Chimera Graph: its 2048 qubits are logically mapped into a 16x16 matrix of unit cells of 8 qubits.

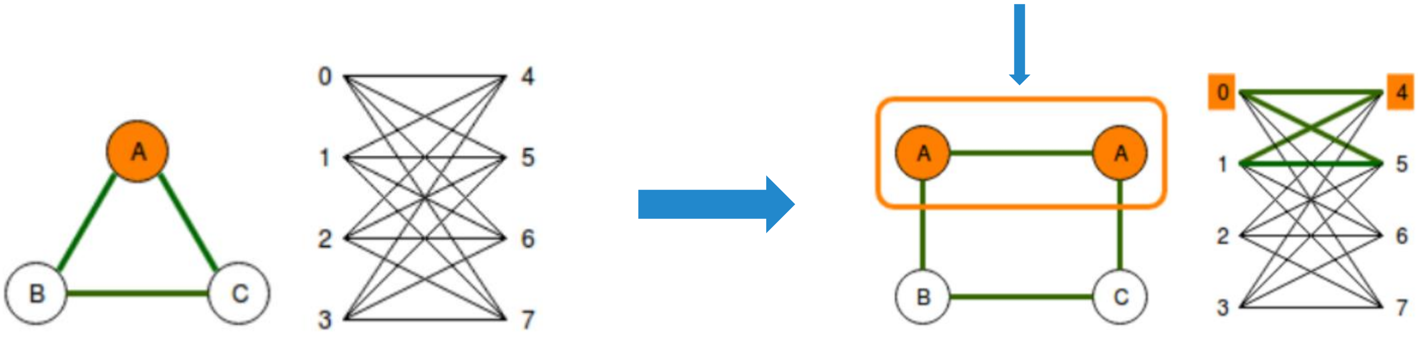


https://docs.dwavesys.com/docs/latest/c_gs_4.html

Chains and Minor Embedding

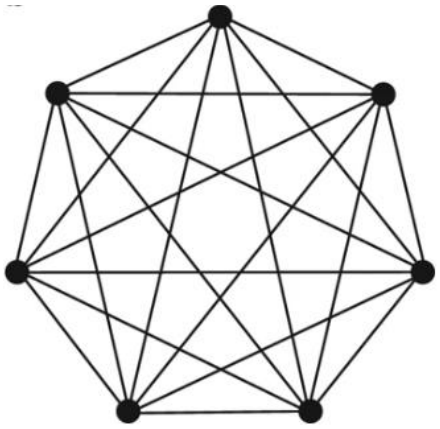
The nodes and edges on the problem graph that represents an objective function translate to the qubits and couplers in Chimera. Each logical qubit, in the graph of the objective function, may be represented by one or more physical qubits. The process of mapping the logical qubits to physical qubits is known as **minor embedding**.

Chain strength is needed (see later at Hands on)

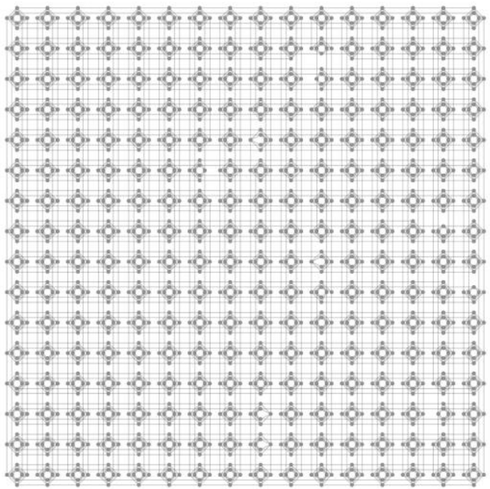


Chains and Minor Embedding

D-Wave's actual 2000Q Quantum Annealer can embbed a 64 node fully connected graph directly on the QPU. D-Wave's *minorminer* library (https://github.com/dwavesystems/minorminer)



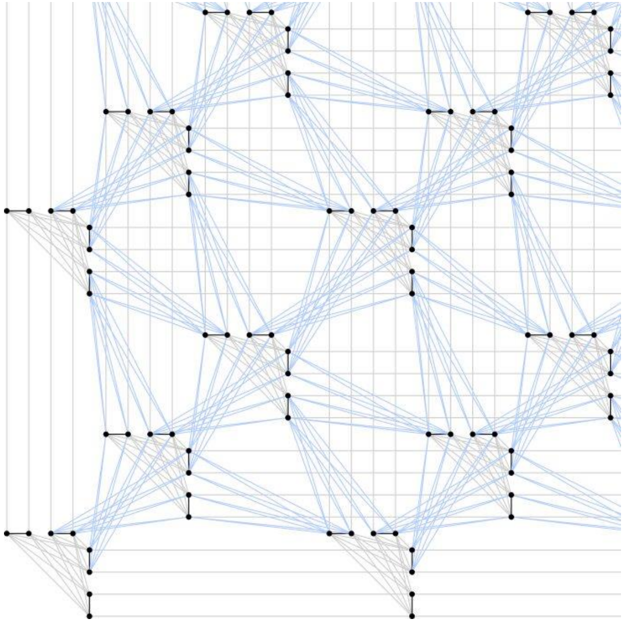
Fully connected Problem Graph with 64 nodes



D-Wave 2000Q with 2048 Qubits

D-Wave‘s Next Generation - Pegasus

- D-Wave‘s Pegasus architecture has 5640 Qubits with a max connectivity degree of 15
- Complete graph of 180 nodes embeddable



https://arxiv.org/abs/1901.07636

Ocean SDK and QBSolv

Ocean SDK is a suite of tools D-Wave Systems provides on the D-Wave GitHub repository for solving hard problems with quantum computers

https://github.com/dwavesystems/dwave-ocean-sdk

QBSolv is a decomposing solver, that finds a minimum value of a quadratic unconstrained binary optimization (QUBO) problem

https://github.com/dwavesystems/qbsolv

Questions?

Now Hands-On:

https://github.com/ChrisRoch/QAP-QUBO