Optimization & Quantum machines

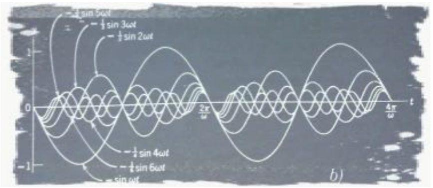
(especially with quantum annealing)

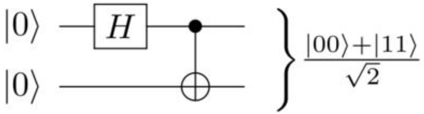
Clermont-Ferrand, May, 2022

Samuel Deleplanque
(prénom.nom@junia.com)

Outline

- |0> |1> Superposition of works
- 1 Quantum Bits
- 2 Quantum Computers
  Focus: D-Wave machines
- 3 The Quantum Annealing process (overview)
- 4 An unconstrained problem: the Max Cut
- 5 The Capacitated Vehicles Routing Problem





Outline

Superposition of works

- Download all the files
- Environment to repeat the experiments.
- D-Wave account
- Online IDE
- OCEAN

To be prepared for the “livecoding”

- **D-Wave subscription**
  - https://cloud.dwavesys.com/leap/signup/
- **Online IDE Info**
  - https://ide.dwavesys.io/workspaces/
- **To Go Further**
  - https://www.youtube.com/watch?v=62gDQ14pjwM

- **Files**
  - This presentation, python codes, papers and other files are available at:
    - https://drive.google.com/drive/folders/1ELoZpo5sHA8TTPF5MQDhyYINq1D5w9rw?usp=sharing
- **shorturl.at/goHIW**

Overview of the Frameworks

- **Cirq**
  - **Google** solution
  - https://cirq.readthedocs.io/en/stable/
- **Qiskit**
  - ***IBM*** *Quantum Experience*
  - https://quantum-computing.ibm.com/
- **Amazon Braket**
  https://aws.amazon.com/fr/braket/

- and few others:
  - Rigetti (Forest/PyQuil),
  - Microsoft (AzureQuantum/LiQui/Q#),
  - From labs: PyTKET, ProjectQ, QuTip etc.)

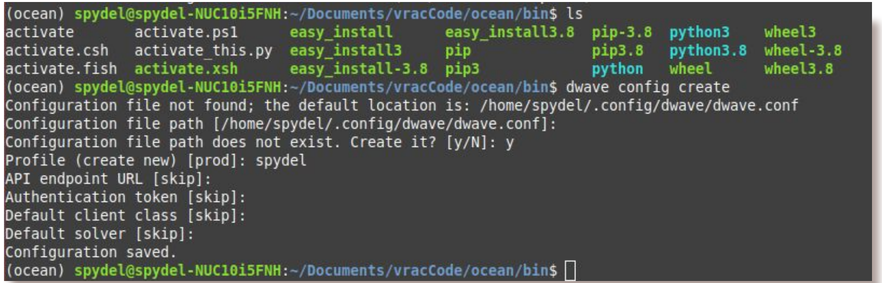
D-Wave
The Quantum Computing Company™

- **D-Wave Leap**
  - **Ocean Library (python)**
  - **Not Universal Quantum Computer**
    - **(no universal set of gates)**
  - **"High Level" programming**
  - https://www.dwavesys.com/take-leap

Some help to install the D-Wave Ocean API on Debian based machine

- Linux commands :
  - sudo apt-get install python<version>
  - sudo pip install virtualenv
  - virtualenv ocean
  - pip install dwave-ocean-sdk
  - git clone https://github.com/dwavesystems/dwave-ocean-sdk.git
  - cd dwave-ocean-sdk ou cd ocean
  - python setup.py install ou ./python ./easy_install install
  - Add to your cod the token available in your dashboard

- More detail: https://docs.ocean.dwavesys.com/en/stable/overview/install.html



```
(ocean) spydel@spydel-NUC10i5FNH:~/Documents/vracCode/ocean/bin$ ls
activate       activate.ps1      easy_install      easy_install3.8  pip-3.8  python3    wheel3
activate.csh   activate_this.py  easy_install3     pip              pip3.8   python3.8  wheel-3.8
activate.fish  activate.xsh      easy_install-3.8  pip3             python   wheel      wheel3.8
(ocean) spydel@spydel-NUC10i5FNH:~/Documents/vracCode/ocean/bin$ dwave config create
Configuration file not found; the default location is: /home/spydel/.config/dwave/dwave.conf
Configuration file path [/home/spydel/.config/dwave/dwave.conf]:
Configuration file path does not exist. Create it? [y/N]: y
Profile (create new) [prod]: spydel
API endpoint URL [skip]:
Authentication token [skip]:
Default client class [skip]:
Default solver [skip]:
Configuration saved.
(ocean) spydel@spydel-NUC10i5FNH:~/Documents/vracCode/ocean/bin$
```

Outline

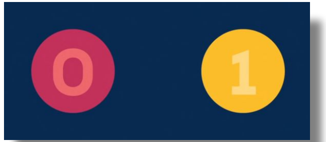
Quantum Bits

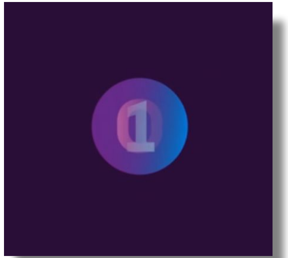
- Superposition
- Measure
- Entanglement
- Decoherence

bits

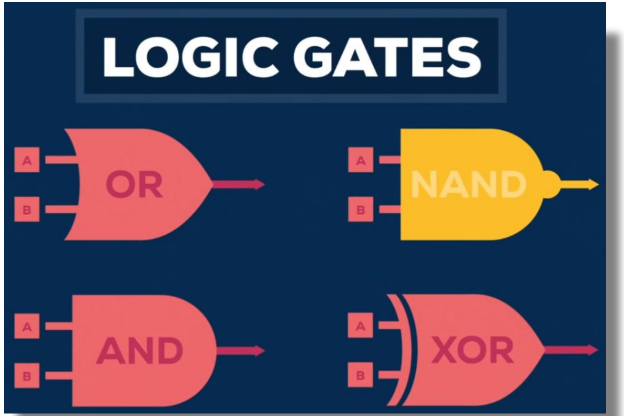
Versus

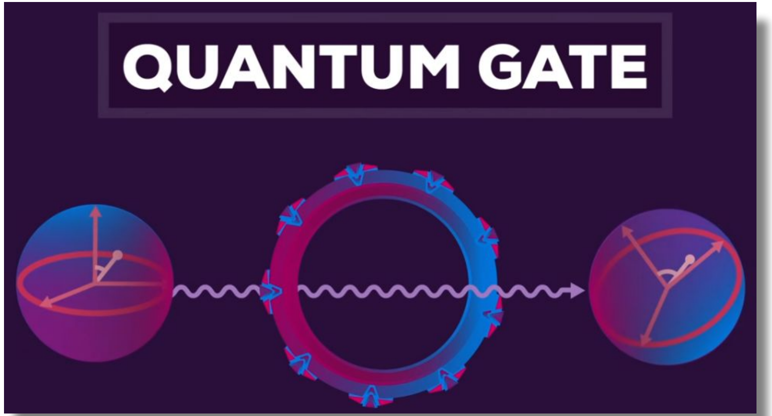
qubits





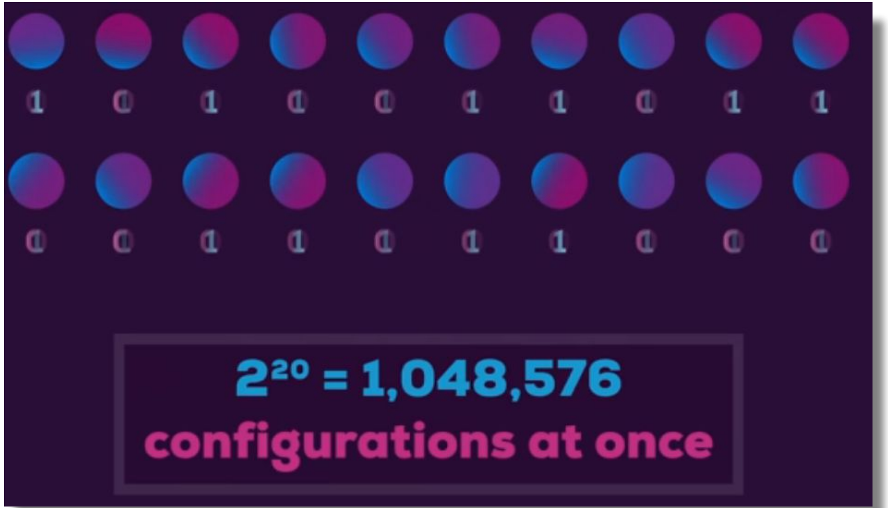
Superposition 0-1
((|0› and |1›)
or (|0› or |1›)





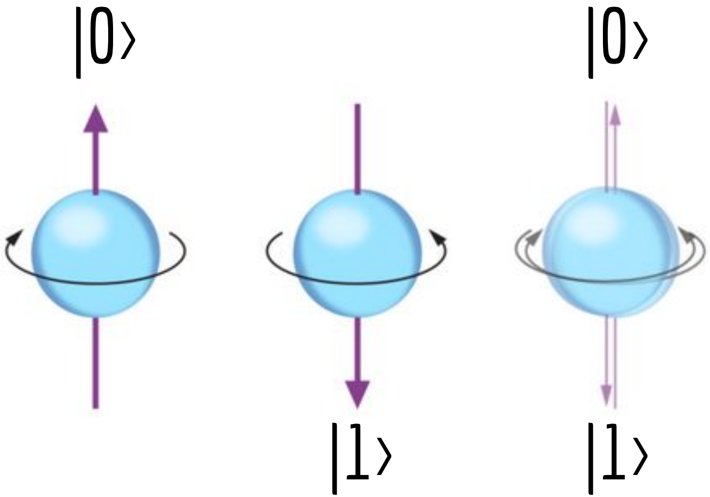
pic: https://www.youtube.com/watch?v=JhHMJCUmq28&ab_channel=Kurzgesagt%E2%80%93InaNutshell

Qubit Superposition



Electron Spin

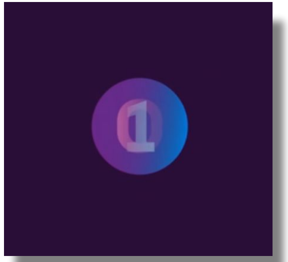
(hardware example of a qubit)



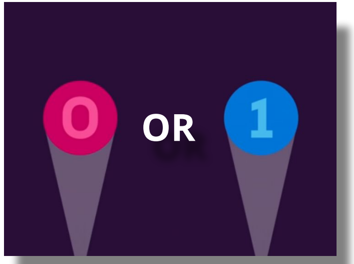
pic: https://www.youtube.com/watch?v=JhHMJCUmq28&ab_channel=Kurzgesagt%E2%80%93InaNutshell
& https://blog.ifs.com/2019/01/a-quantum-leap-in-computing-power/

Measuring a qubit

From the superposition of states (0 **and** 1)
to the classical states (0 **or** 1)



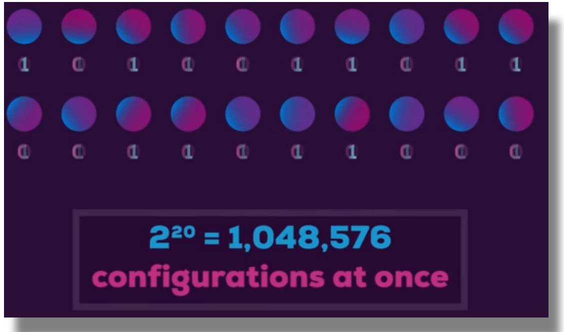
Measure



Superposition 0-1

Measure
(before the end of the coherence time)

Example:



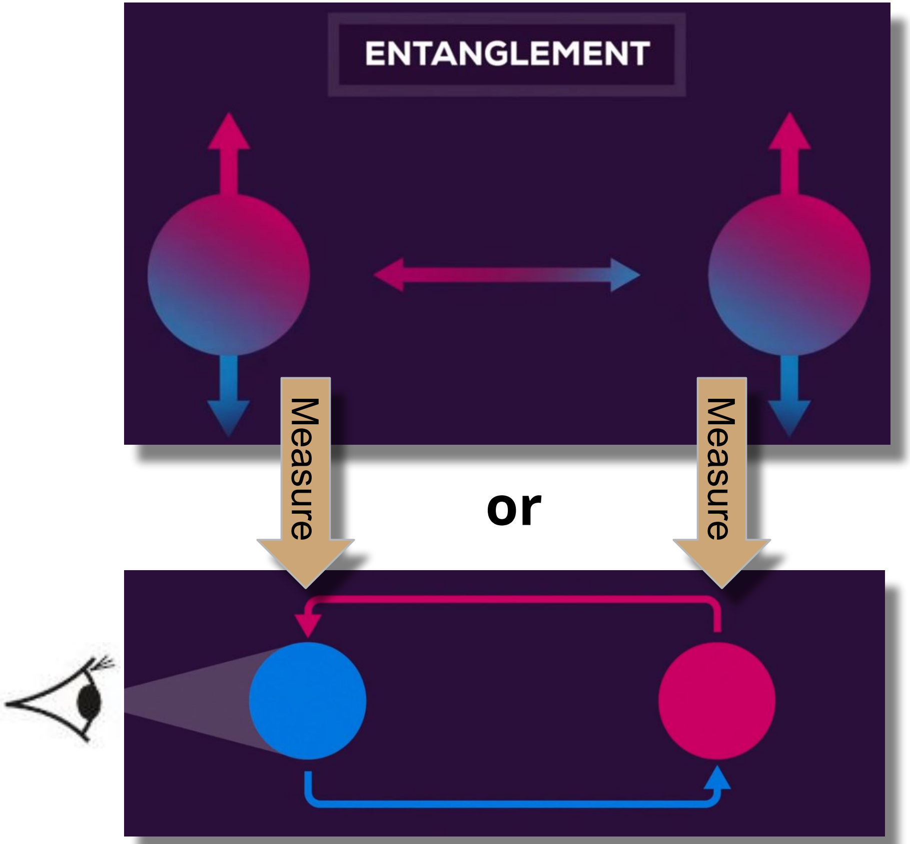
Measure

One solution
(20 qubits became 20 bits
with only one value)

pic: https://www.youtube.com/watch?v=JhHMJCUmq28&ab_channel=Kurzgesagt%E2%80%93InaNutshell

The entanglement

(fr. intrication/enchevêtrement)

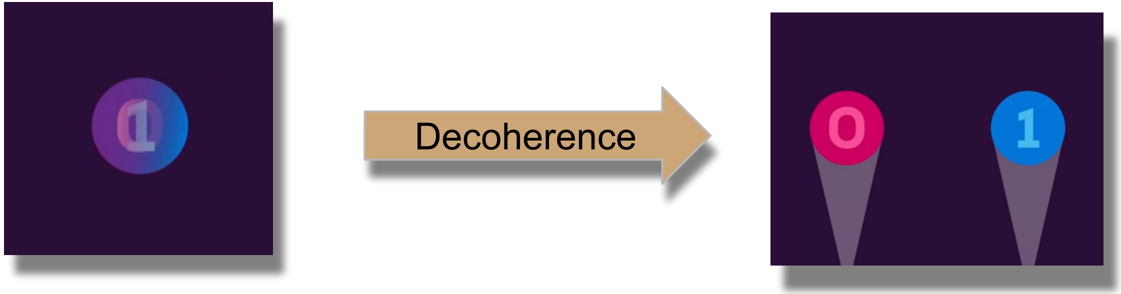


pic: https://www.youtube.com/watch?v=JhHMJCUmq28&ab_channel=Kurzgesagt%E2%80%93InaNutshell

Decoherence

- Appears after a short amount of time because of:
  - radiation, light, sound, heat, magnetic fields, etc.
  - or…the measure!
- The system goes **from a quantum system to a classical one**,
- Limit the time to work on qubits.

Example on a qubit: break the superposition state to ‘1 or ‘0’.

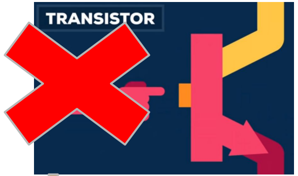


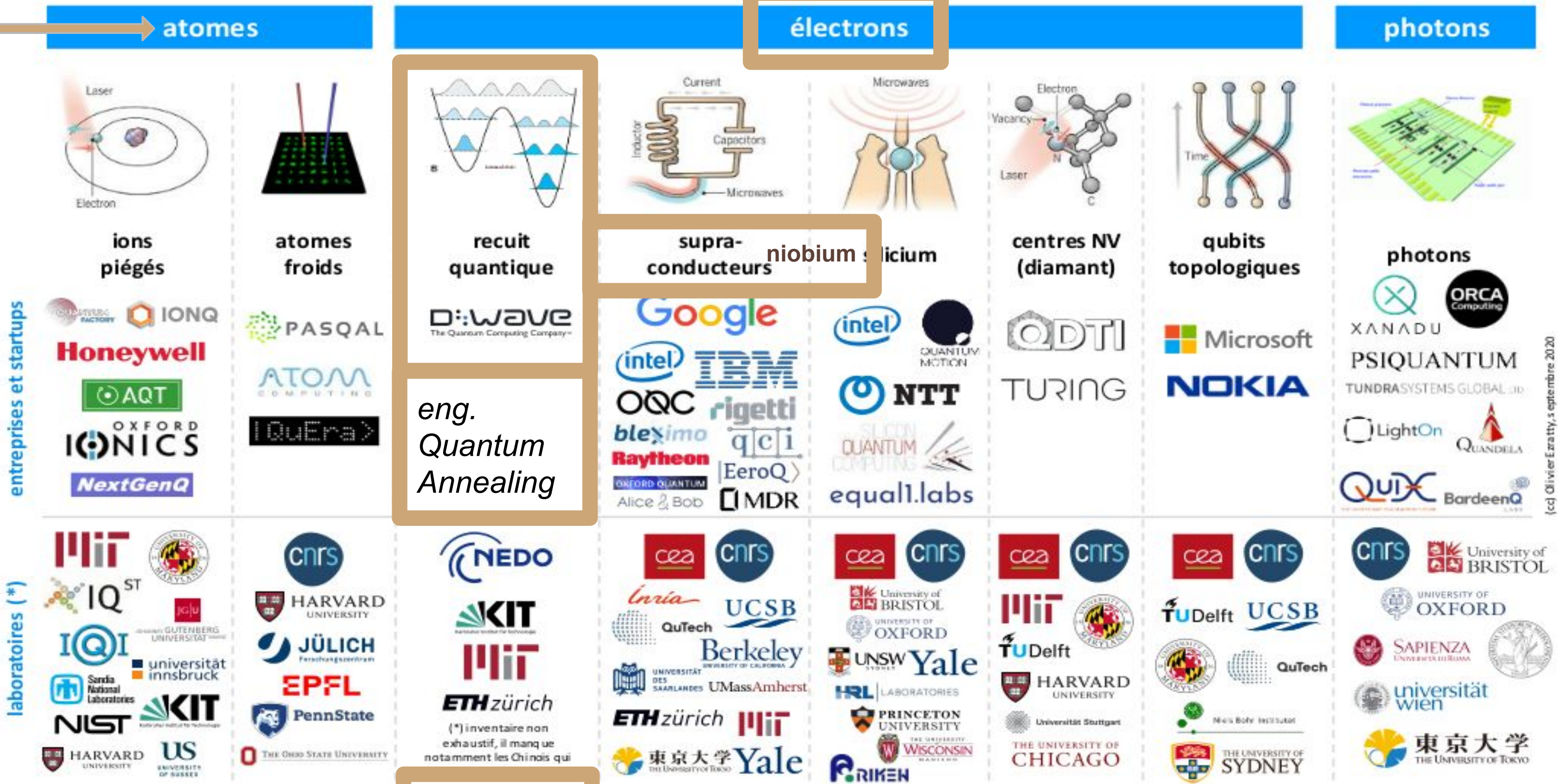
pic: https://www.youtube.com/watch?v=JhHMJCUmq28&ab_channel=Kurzgesagt%E2%80%93InaNutshell

Outline

2 Quantum Computers
Focus: D-Wave machines

D-Wave **qubits** among the quantum computing world





Atos → https://atos.net/wp-content/uploads/2020/07/PR-Atos-opens-up-a-new-path-to-quantum-annealing-simulation.pdf
& https://atos.net/fr/2020/communiques-de-presse_2020_07_07/atos-ouvre-la-voie-a-la-simulation-du-recuit-quantique

original pic: https://www.oezratty.net/wordpress/2020/comprendre-informatique-quantique-edition-2020/

Models of quantum computing

Universal

- **Gate-Based** IBM, Google
- **Adiabatic** ≈ look for the ground state of a **particular** hamiltonian → Quantum Annealing
  - Not implemented yet
- Variational
- One-way
  - For optics

Not for all Hamiltonian (not for all computational problems) => QA is then not universal.

Non-Universal

- **Special-purpose** D-Wave The Quantum Computing Company™
  - **Quantum Annealing**
  - (Optical) coherent Ising Machine
    - (no entanglement :( )
    - Full connectivity
  - Neutral Rydberg atom Simulator
    - can be used to solve the max independent set pbm
    - which corresponds to find the minimal energy configuration of an ensemble of Rydberg atoms.
  - Boson Sampler
- Quantum simulators

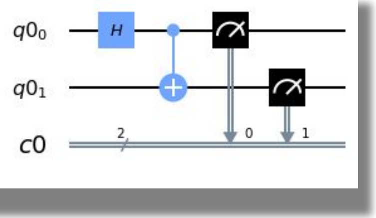
Quantum Computing: 2 main types of machine

Universal Quantum Gates

- Google/IBM/etc. ; <= 127 qubits

Algo:

- Shor (factorize an integer N in time O((log N)^3)))
  - [Shor, 1994]
- Grover (quadratic acceleration to find a value)
  - [Grover, 1996]
- QAOA (Quantum Approximate Optimization Algorithm)
  - [Farhi et al., 2014]



Quantum annealing

- **D-Wave**/Qilimanjaro
- <= 5k qubits.

- Focus on a particular Hamiltonian (ising model):

$$\mathcal{H}_S(s) = -\frac{1}{2}\sum_i \Delta(s)\sigma_i^x + \mathcal{E}(s)\left(-\sum_i h_i\sigma_i^z + \sum_{i<j} J_{ij}\,\sigma_i^z\sigma_j^z\right)$$

(we will come back on this process)

D-Wave over time

D:WAVE The Quantum Computing Company™

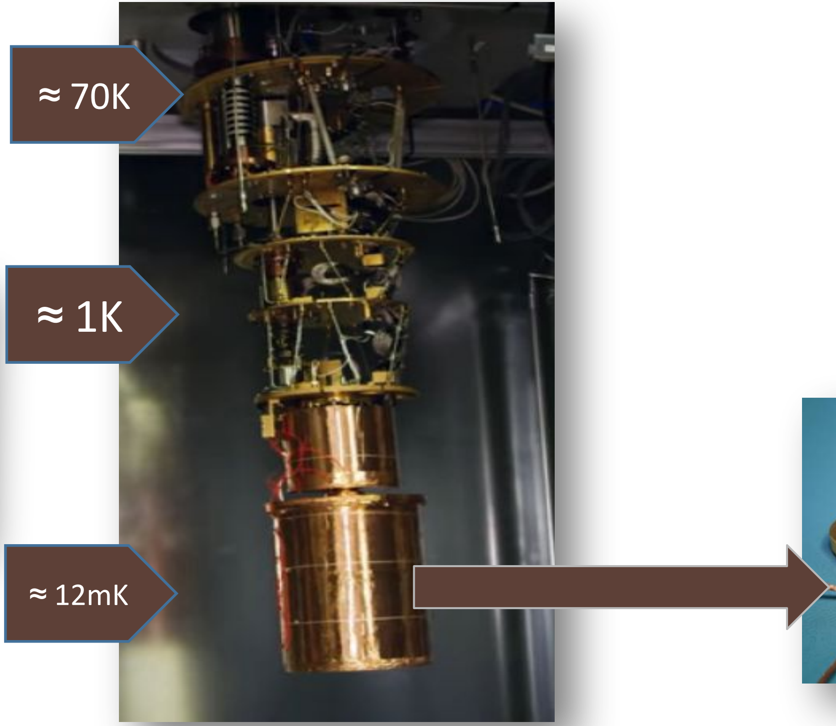
- 1999 > birth
  - 4 founding fathers: **Geordie Rose**, **Haig Farris**, **Bob Wiens** and **Alexandre Zagoski**.
- 2007 > **1st prototype** (**4 qubits**) named "Calypso"
- 2011 > **1st commercialized quantum computer (128 qubits)**
- 2017 > D-Wave 2000Q (cost: $ 15 million)
- end2020> **The 5k qubits Advantage machine is released**

Notable Funders: InQTel (CIA funds), Lockheed Martin, NSA funds (see Research & Technology Penetrating hard targets' *Snowden FIle*) and others.

≤ 2013

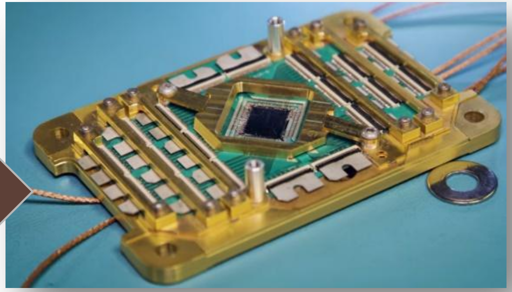
• (S//SI//REL TO USA, FVEY) Conduct basic research in quantum physics and architecture/engineering studies to determine if, and how, a cryptologically useful quantum computer can be built.

Hardware

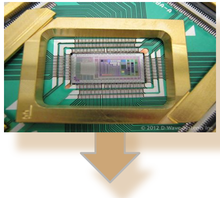


QPU
(Quantum Processing Unit)

Set of D-Wave Nodium superconductors (qubits into the QPU)

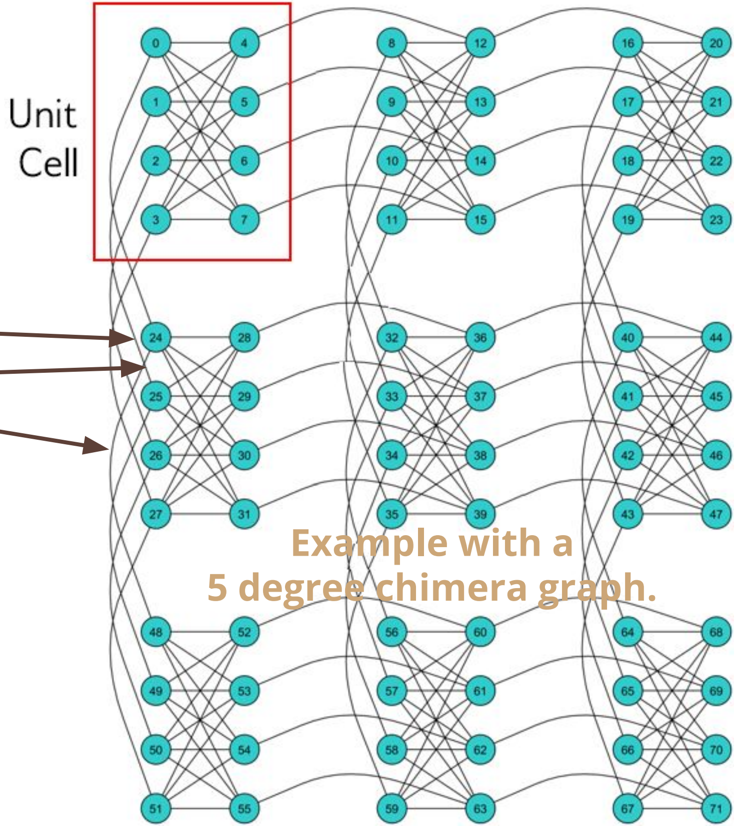


Qubits Graph
(Topology)


QPU

D-Wave's quantum computer hardware can be modeled as a non-oriented graph. This graph has:

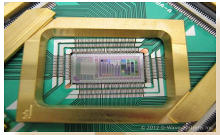
- **qubits** as nodes,
- quantum **couplers** as edges.
- since end of 2020 (*Pegasus* topology), **the degree of the qubits is 15** (6 before).



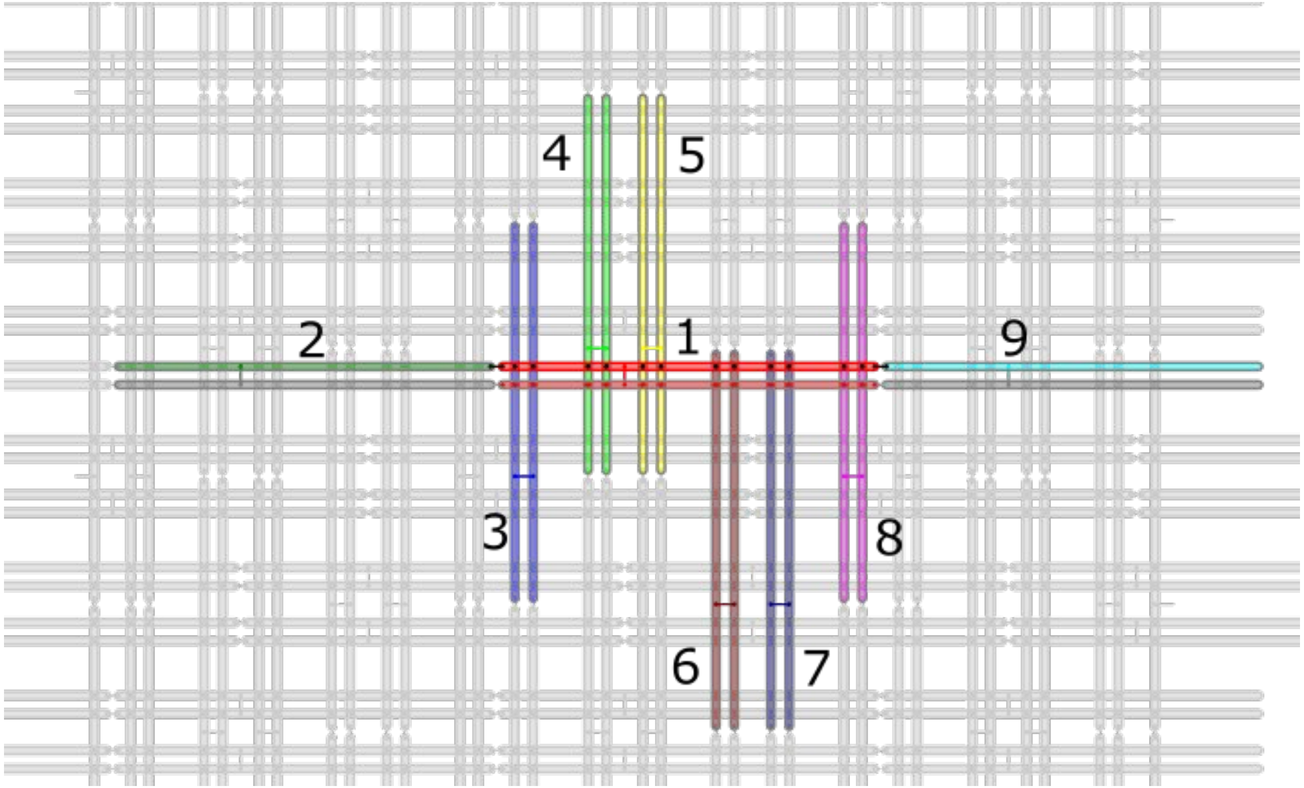
Example with a
5 degree chimera graph.

Qubits Graph

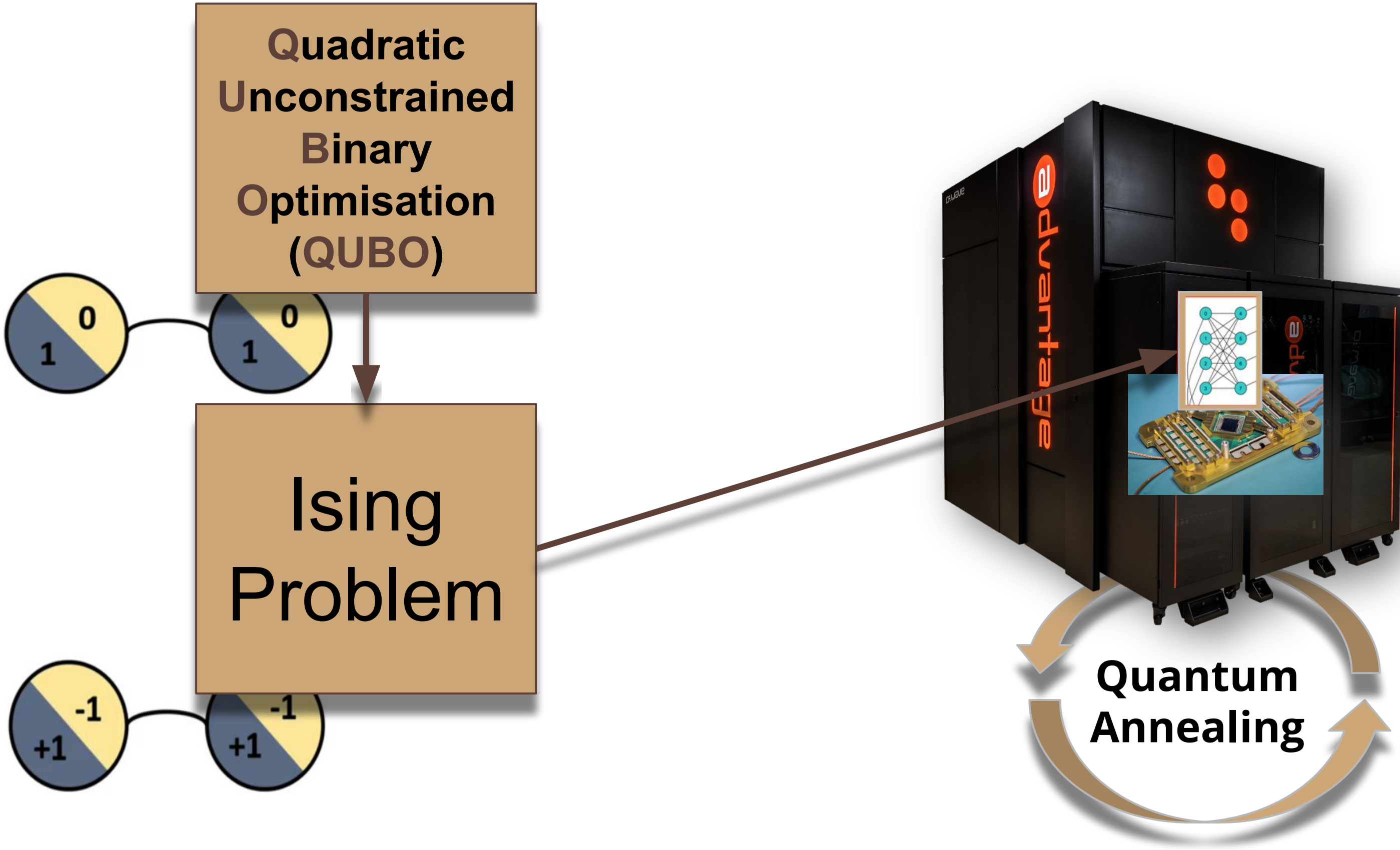
(Pegasus Topology - 15 connectivity)



QPU

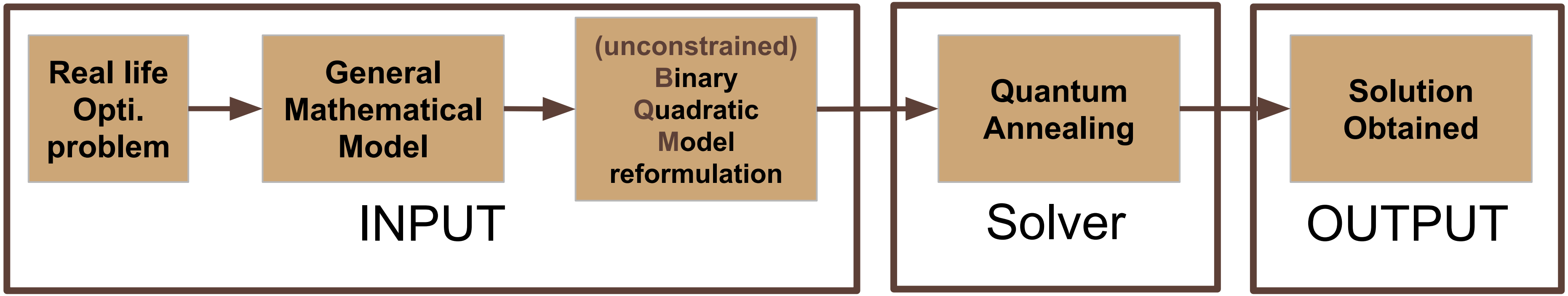


INPUT: Unconstrained Binary Quadratic Programs

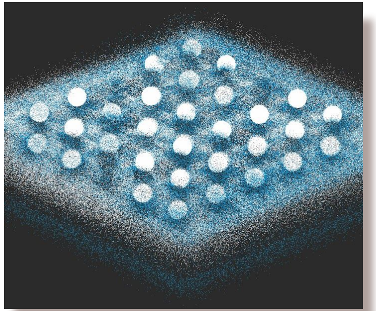


General Process

INPUT › QUANTUM ANNEALING › OUTPUT



Outline

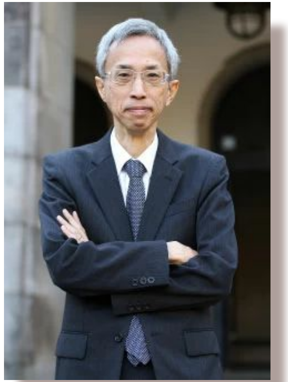


3 The Quantum Annealing process (overview)

"Physics can help solve [Optimization Problems] because we can frame them as energy minimization problems. "

"The D-Wave system can be viewed as a hardware heuristic that minimizes [..] objective functions using a physically realized version of quantum annealing."

D-Wave and Quantum Annealing



Hidetoshi Nishimori

- "Father" of the **Quantum Annealing Metaheuristic**
  - K., Tadashi, and H. Nishimori. "Quantum annealing in the transverse Ising model." *Physical Review E* 58.5 published in **1998**;
  - From an idea of: Apolloni, B., Carvalho, C., & De Falco, D. (1989). Quantum stochastic optimization. *Stochastic Processes and their Applications*, *33*(2), 233-244.
- Based on classical computer Metaheuristic **Simulated Annealing**
  - Kirkpatrick, Scott, C. Daniel Gelatt, and Mario P. Vecchi. "Optimization by simulated annealing." *science* 220.4598 (1983): 671-680;
- Prove that the Quantum Annealing gives better results that the Simulated Annealing.

D-Wave The Quantum Computing Company™

- Starting the work in **1999**, the 1st computer based on quantum annealing released in **2007**.

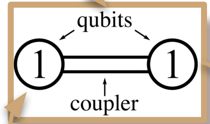
The Hamiltonian during the quantum annealing....

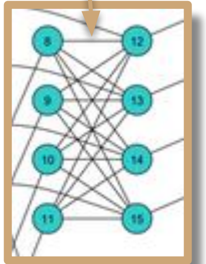
- The **Hamiltonian ($\mathcal{H}$)** includes 2 types of interaction:
  - h_i : the "programmable" external magnetic field called a **biais**,
  - J_{ij} : the interaction between each neighboring qubits ***i*** and ***j*** **(coupling)**.

weight

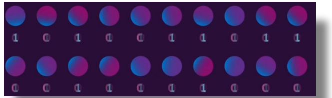
$$\mathcal{H}_S(s) = -\frac{1}{2}\sum_i \Delta(s)\sigma_i^x + \mathcal{E}(s)\left(-\sum_i h_i\sigma_i^z + \sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z\right)$$

weight





Initial Hamiltonian



Final (ising) **Hamiltonian**

$$\begin{cases} \max x_0 - \sum_{i=1}^{n} x_i - \sum_{(i,j)\in E} x_i x_j, \\ x \text{ binary.} \end{cases}$$

pics: https://www.youtube.com/watch?v=tnikftltqE0&t=125s

Slow Final Hamiltonian Integration

(based on the Adiabatic Theorem)

$$\mathcal{H}_S(s) = -\frac{1}{2}\sum_i \Delta(s)\sigma_i^x + \mathcal{E}(s)\left(-\sum_i h_i\sigma_i^z + \sum_{i<j} J_{ij}\,\sigma_i^z\sigma_j^z\right)$$

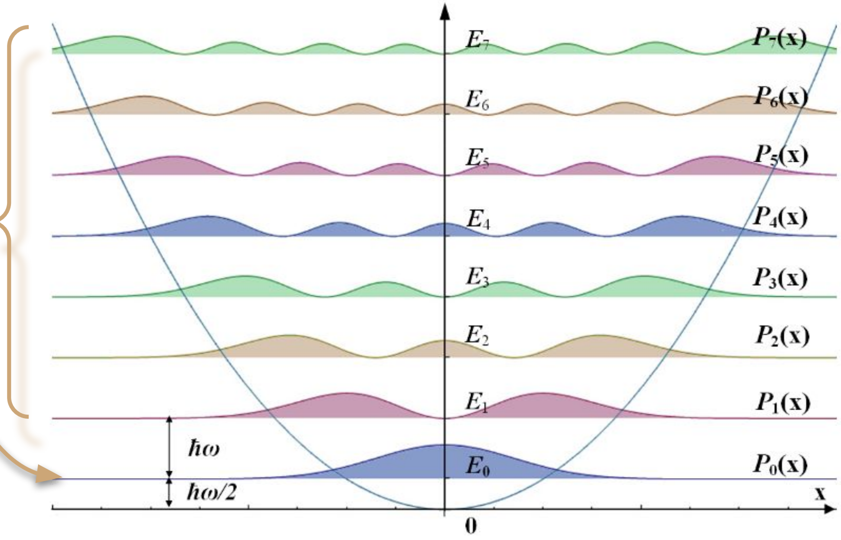
$$H = (1-s)H_0 + sH_1$$

Principle: we need to integrate slowly the final Hamiltonian (i.e., our model) step by step avoiding the electrons (qubits) to “jump” in an…

…excited state

… and staying in the **ground state**

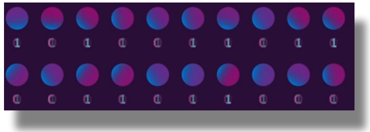
. In some way: we map slowly the qubits and couplers from our model to obtain the optimal solution.



Quantum annealing

First Step

The qubits start in a superposition state (**zero and one**).



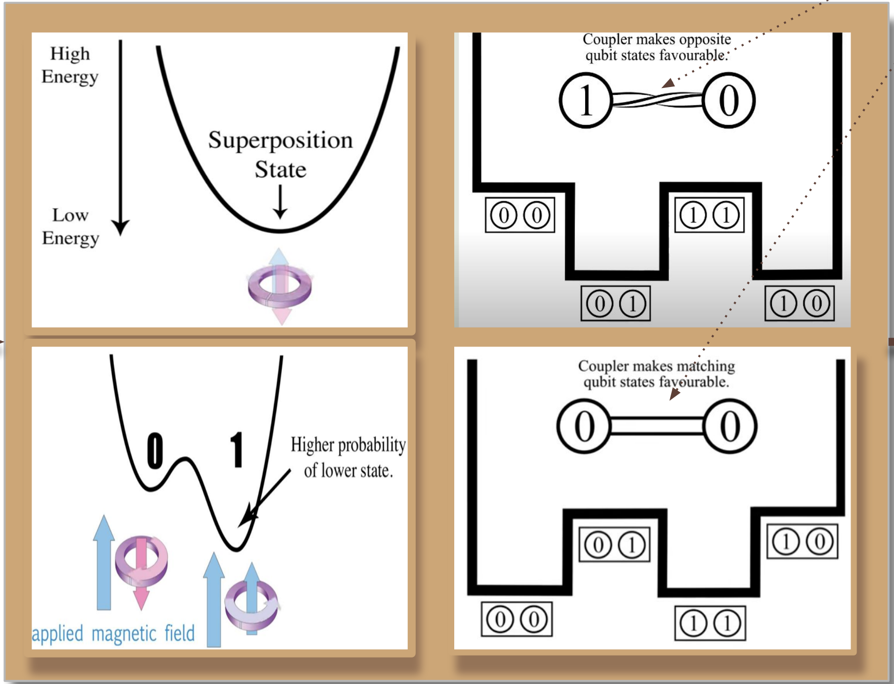
Work on the 0 and 1 states probabilities (amplitudes).

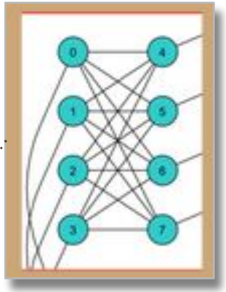
Last step

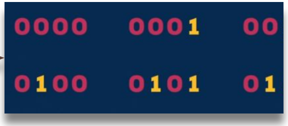
The measure of each qubit make them going to either the **zero state or** the **one state** (according to their probability).

Pics: D-Wave

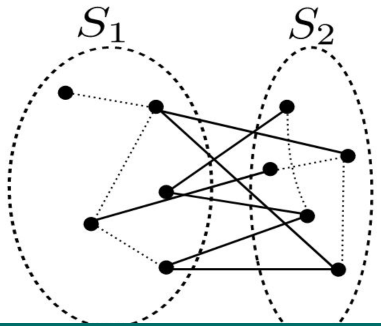
Quantum annealing





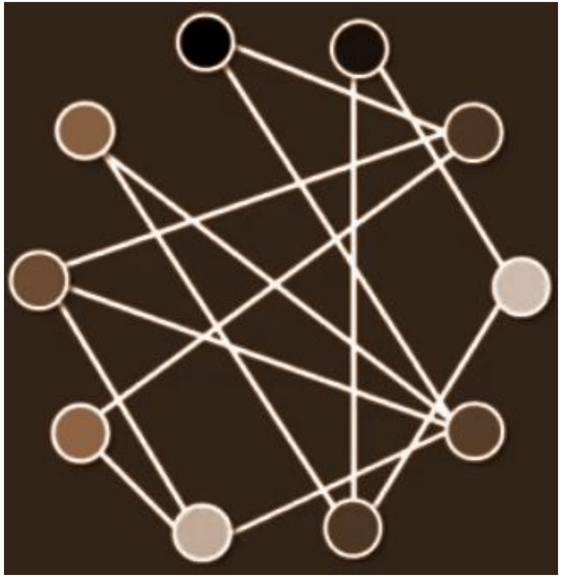


Outline



4 **An unconstrained problem: the Max Cut**

- Process
- Quadratic model
- Let's code it!

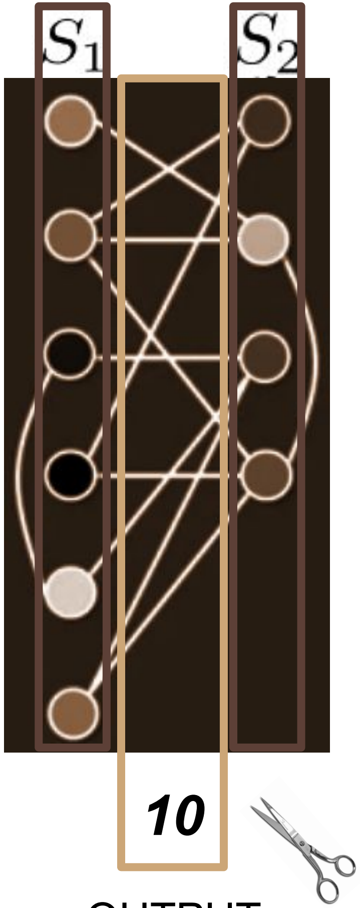


INPUT

The Max-Cut Problem

Finding 2 sets of nodes S1 and S2, such that the **total cost** of the edges between the set S1 and the set S2 is **as large as possible**.

(here: no weighted)



OUTPUT

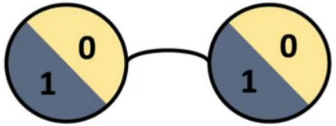
Problem formulation: 2 mathematically **equivalent** models

Choice n°1

Choice n°2

Binary Quadratic Model

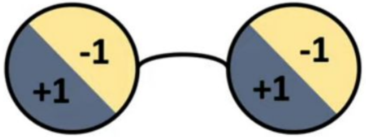
Quadratic Unconstrained Binary Optimisation (QUBO)

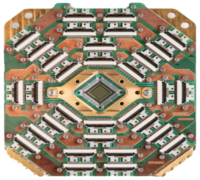


Mathematically **Equivalent models**

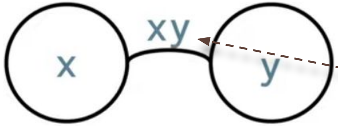
“Fit in” D-Wave QPU

Ising Problem



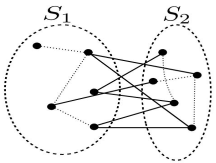


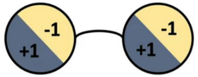
1 coupler/var prod.



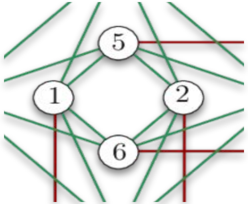
2 qubits/variables

(Hardware) the **couplers** (making pairs of qubits) corresponds to (software) **quadratic expressions**.





Max-cut Ising Model Formulation

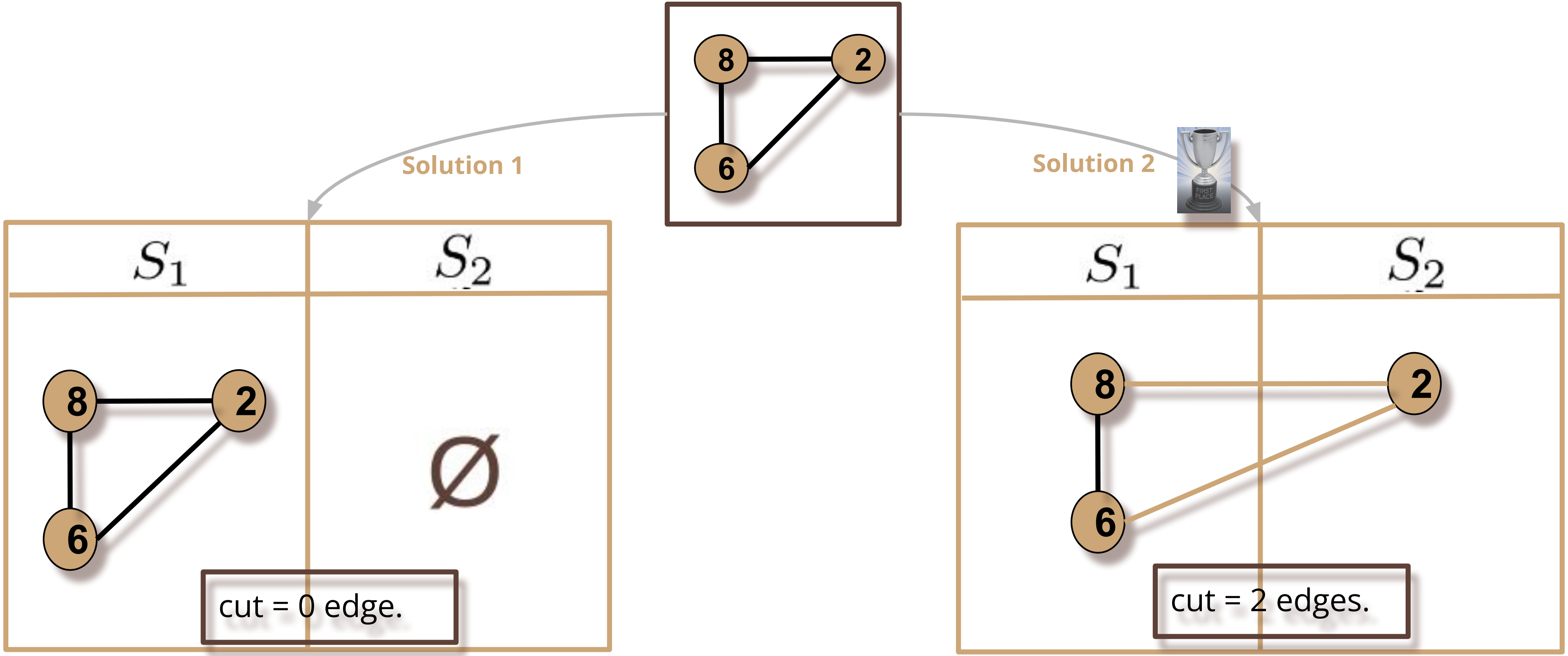


- Let's take a Graph **G=(X,E)** where:
  - **X** is the set of nodes and **E** the set of edges.
- We can identify each node **i** of the graph with a **binary variable {-1;1}** Z_i such that:
  - Z_i = 1 if **i** is in one group *S*,
  - Z_i = -1 if **i** is in the other group.

(w(i,j) weights)

- The Max-cut problem consists in minimizing: $\sum_{(i,j)\in E} Z_i Z_j$

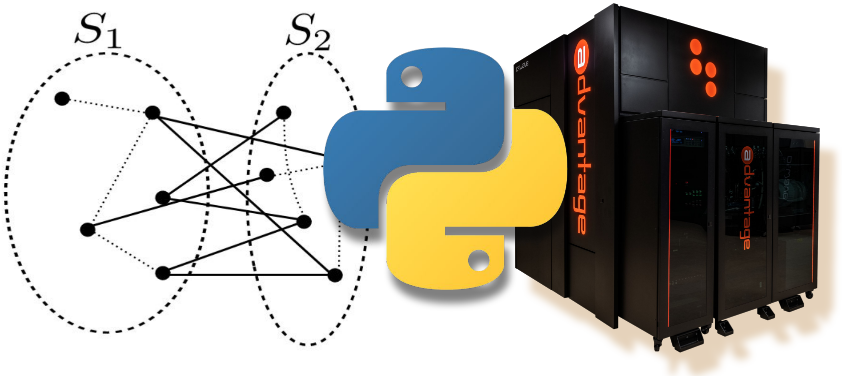
Examples



f(Solution 1) = 3

f(Solution i) = $Z_2 Z_6 + Z_6 Z_8 + Z_8 Z_2$

f(Solution 2) = -1



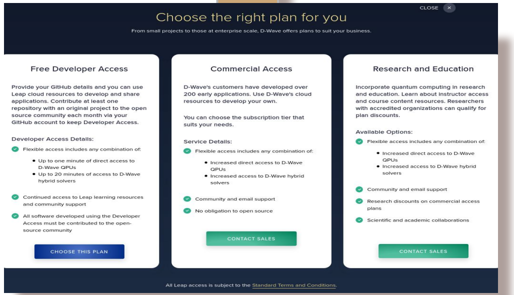
To solve the MaxCut

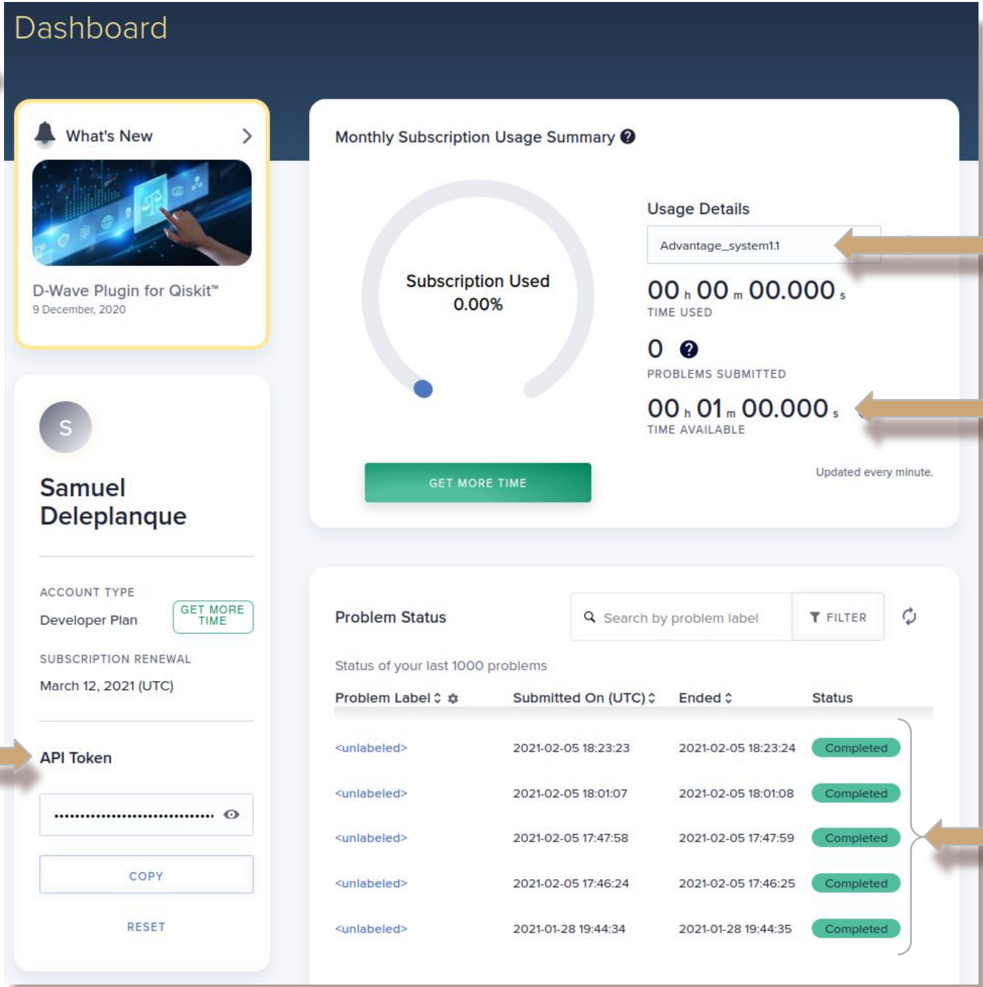
-

A quadratic formulation for a D-Wave machine through a Python Code.

How can I solve a problem on a D-Wave machine? https://cloud.dwavesys.com/leap/

Subscription





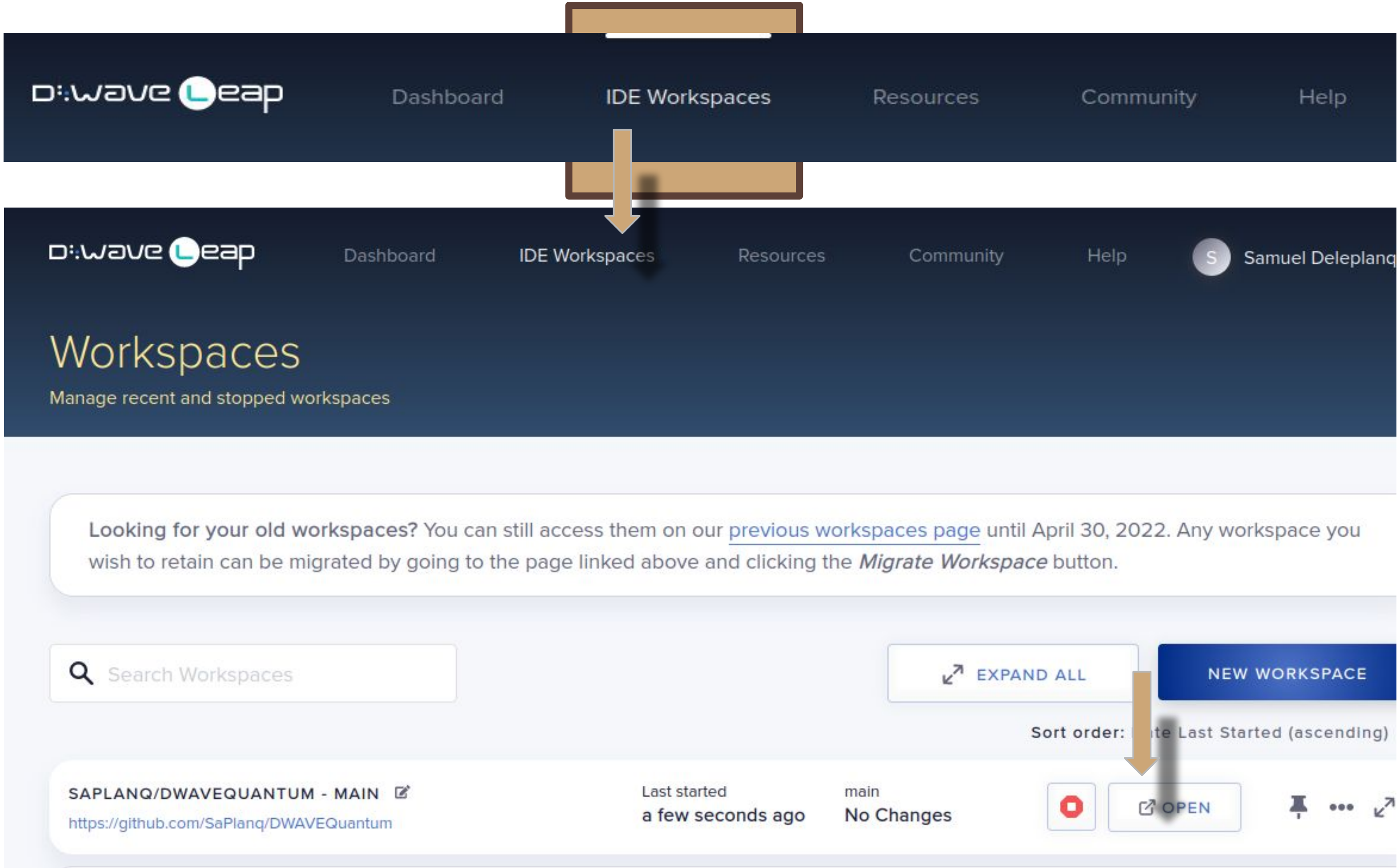
D-Wave Machine

Remaining time

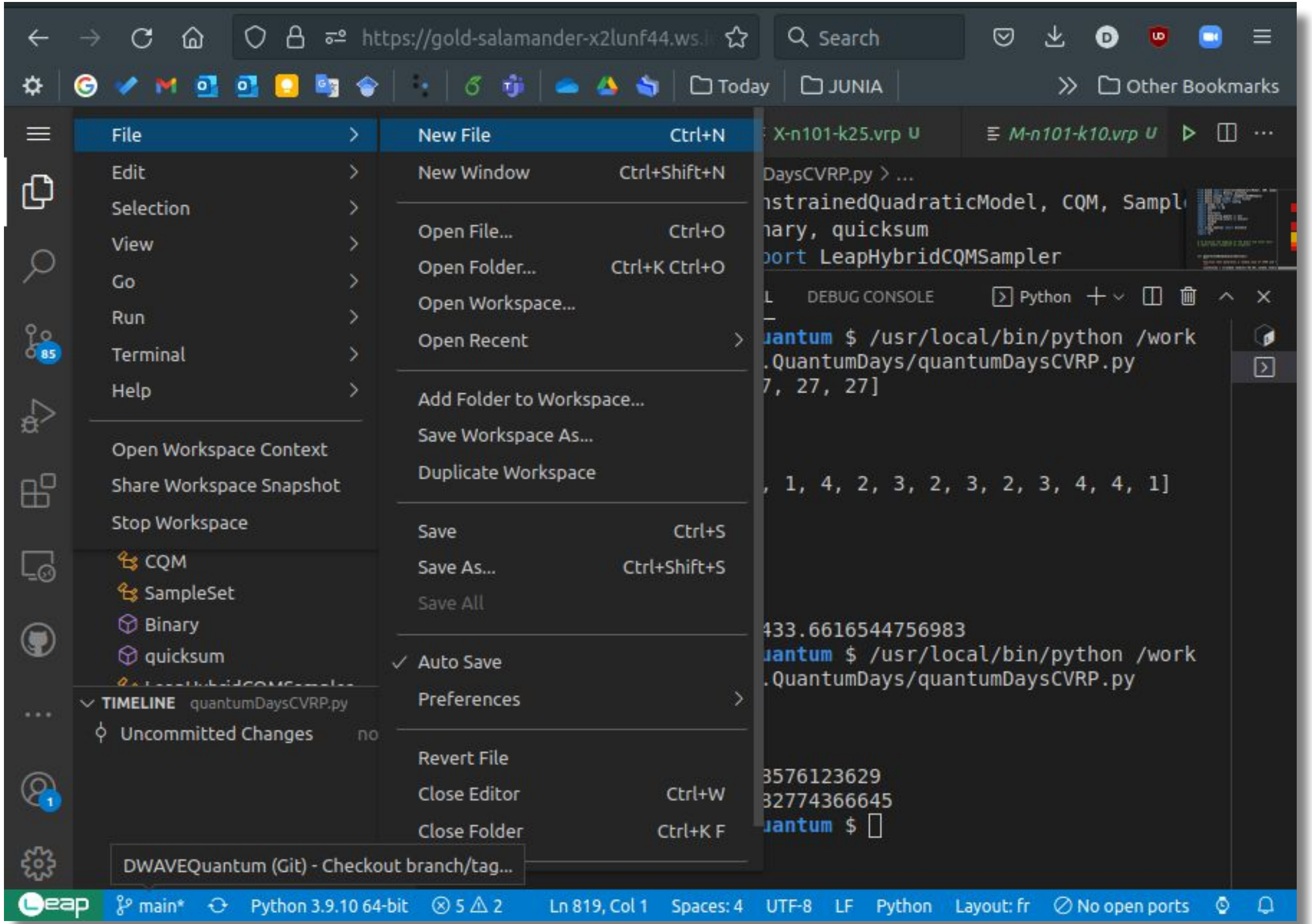
Previous Problems solved

Tokens to use into you local OCEAN code (useless for online IDE)

How can I solve a problem on a D-Wave machine?



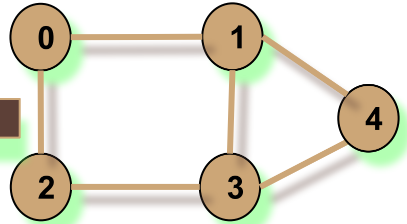
How can I solve a problem on a D-Wave machine?



Implementation of the hamiltonian

$$\mathcal{H}_S(s) = -\frac{1}{2}\sum_i \Delta(s)\sigma_i^x + \mathcal{E}(s)\left(-\sum_i h_i\sigma_i^z + \sum_{i<j} J_{ij}\,\sigma_i^z\sigma_j^z\right)$$

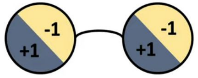
```
# We enter the 6 edges of our graph with a weight equals to 1.
J = {(0,1):1,(0,2):1,(1,3):1,(1,4):1,(2,3):1,(3,4):1}
h = {} # We do not have external magnetic field in this case
```



No external field (biais)

Couplers / edges no weight (':1')

```
model = dimod.BinaryQuadraticModel(h, J, 0.0, dimod.SPIN)
```



quantumDays.MaxCut.0.py

Implementation of the hamiltonian

```
sampler = EmbeddingComposite(DWaveSampler())
```

```
##    Quantum Annealing of D-WAVE Advantage Machine          ##
sampler = EmbeddingComposite(DWaveSampler(solver='Advantage_system1.1'))
sampler_name = sampler.properties['child_properties']['chip_id']
response = sampler.sample(model, num_reads=5000)
print("The solution obtained by D-Wave's quantum annealer",sampler_name,"is")
print(response)
```

Number of anneals

```
## Send problem to Sampler
sampleset = sampler.sample_qubo(Q)
```

Example for a QUBO.

quantumDays.MaxCut.0.py

Problem Inspector

```
import dwave.inspector as inspector
from dwave.system import DWaveSampler, EmbeddingComposite
```

 # blablacode

```
inspector.show(response)
```

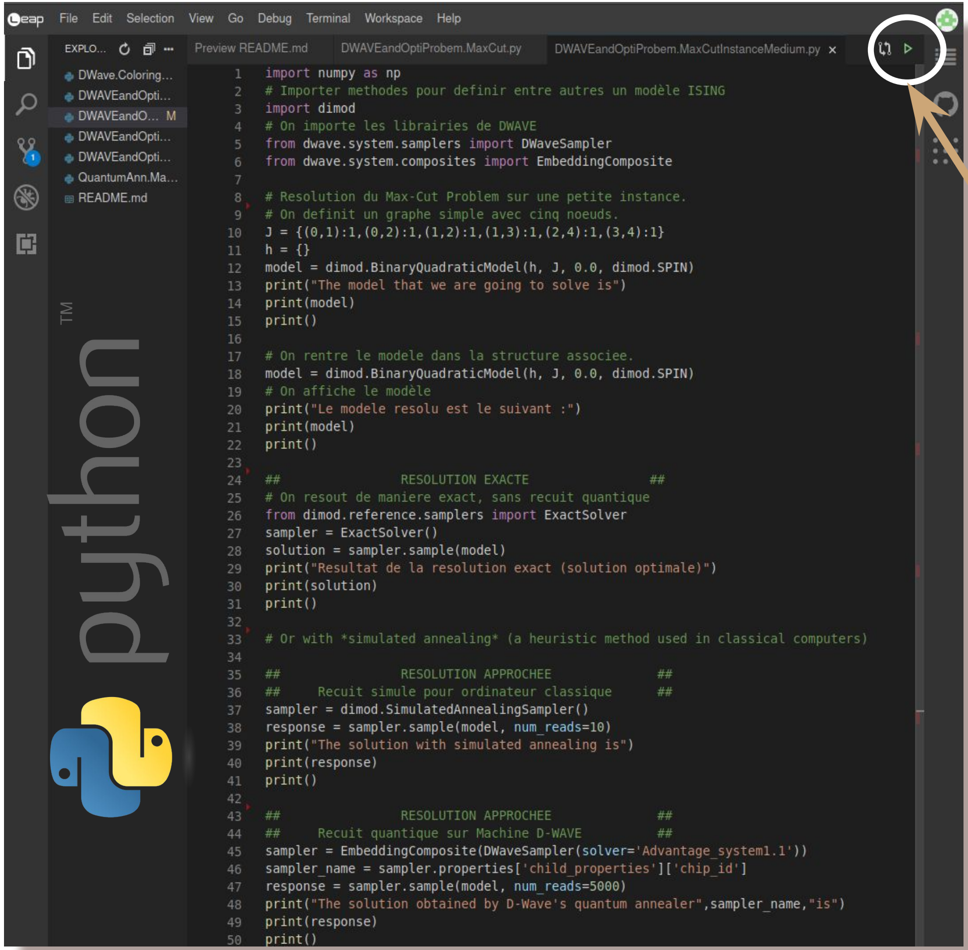
quantumDays.MaxCut.0.py

https://ide.dwavesys.io/#https://github.com/SaPlanq/DWAVEQuantum

Your github address

How can I solve a problem on a D-Wave machine?

1

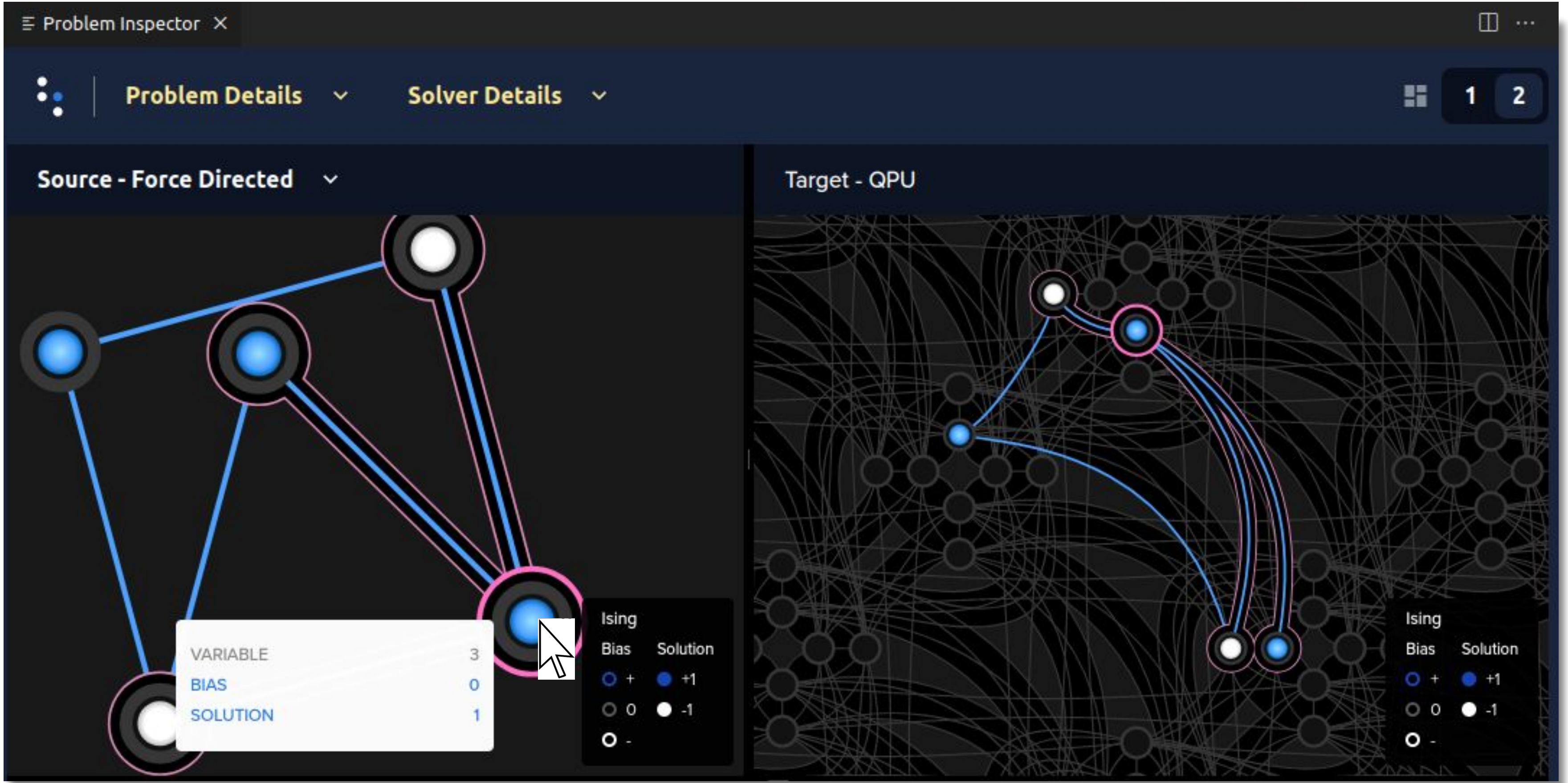


You launch the resolution directly here, and you will have an online terminal showing the result.

2

```
The solution obtained by D-Wave's quantum annealer Advantage_system1.1 is
   0  1  2  3  4 energy num_oc. chain_.
0 -1 +1 +1 -1 +1   -4.0    1398     0.0
1 +1 -1 -1 +1 -1   -4.0    1029     0.0
2 -1 +1 +1 -1 -1   -4.0    1487     0.0
3 +1 -1 -1 +1 +1   -4.0    1084     0.0
4 -1 +1 +1 +1 -1   -2.0       1     0.0
5 +1 -1 -1 -1 +1   -2.0       1     0.0
['SPIN', 6 rows, 5000 samples, 5 variables]
```

D-Wave Quantum Annealing results



D-Wave Quantum Annealing **results**

For sure we did not use all the qubits...

```
The solution obtained by D-Wave's quantum annealer Advantage_system1.1 is
   0  1  2  3  4 energy num oc. chain .
0 -1 +1 +1 -1 +1   -4.0    1398    0.0
1 +1 -1 -1 +1 -1   -4.0    1029    0.0
2 -1 +1 +1 -1 -1   -4.0    1487    0.0
3 +1 -1 -1 +1 +1   -4.0    1084    0.0
4 -1 +1 +1 +1 -1   -2.0       1    0.0
5 +1 -1 -1 -1 +1   -2.0       1    0.0
['SPIN', 6 rows, 5000 samples, 5 variables]
```

lower cost

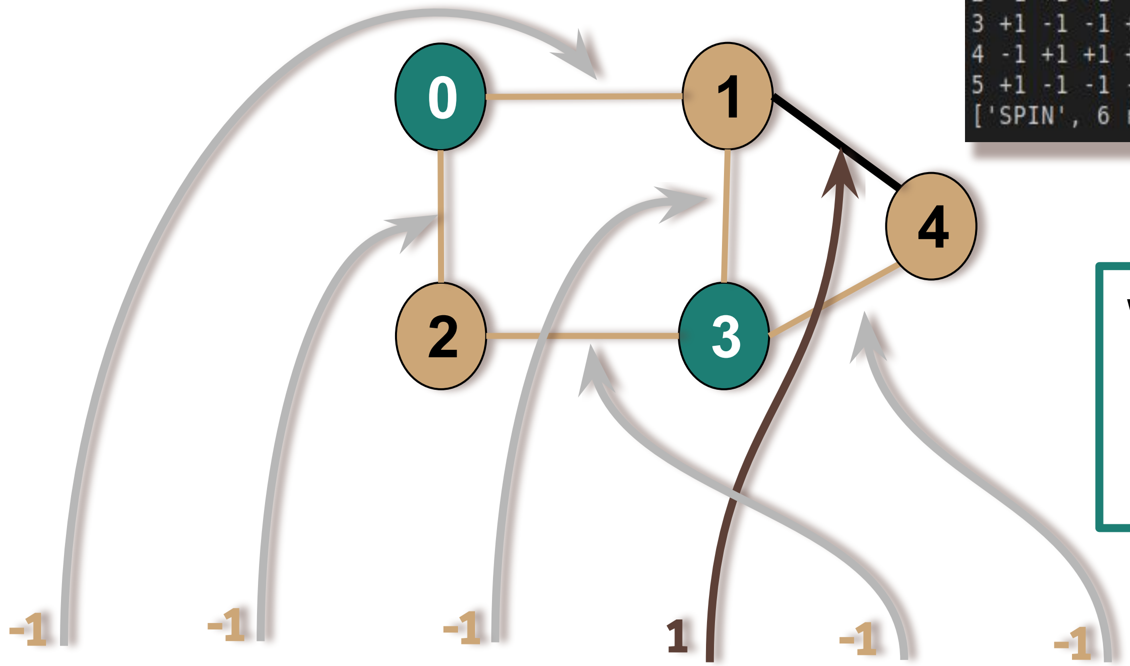
Number of occurrences during the 5000 anneals

$$H = Z_0Z_1 + Z_0Z_2 + Z_1Z_3 + Z_1Z_4 + Z_2Z_3 + Z_3Z_4 = -5 + 1 = -4$$

$$Z_0 = \mathbf{-1} \quad Z_1 = \mathbf{1} \quad Z_2 = \mathbf{1} \quad Z_3 = \mathbf{-1} \quad Z_4 = \mathbf{1}$$

Max-cut-Problem: From the solution to the graph.

```
   0  1  2  3  4 energy num_oc. chain_.
0 -1 +1 +1 -1 +1   -4.0    1398    0.0
1 +1 -1 -1 +1 -1   -4.0    1029    0.0
2 -1 +1 +1 -1 -1   -4.0    1487    0.0
3 +1 -1 -1 +1 +1   -4.0    1084    0.0
4 -1 +1 +1 +1 -1   -2.0       1    0.0
5 +1 -1 -1 -1 +1   -2.0       1    0.0
['SPIN', 6 rows, 5000 samples, 5 variables]
```



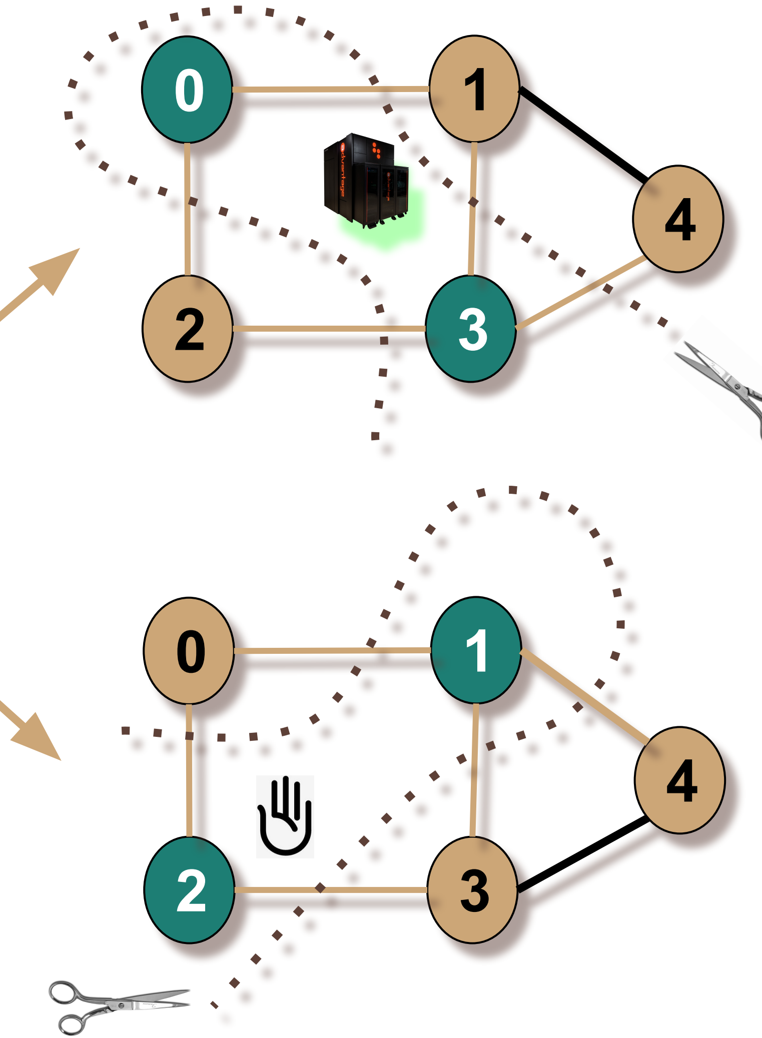
We deduce a **5 edges** cut.

$$H = Z_0Z_1 + Z_0Z_2 + Z_1Z_3 + Z_1Z_4 + Z_2Z_3 + Z_3Z_4 = -5+1 = -4$$

$$Z_0 = \mathbf{-1} \quad Z_1 = \mathbf{1} \quad Z_2 = \mathbf{1} \quad Z_3 = \mathbf{-1} \quad Z_4 = \mathbf{1}$$

Max-cut-Problem: several optimal solutions

```
The solution obtained by D-Wave's quantum a
   0  1  2  3  4 energy num oc. chain .
0 -1 +1 +1 -1 +1   -4.0    1398     0.0
1 +1 -1 -1 +1 -1   -4.0    1029     0.0
2 -1 +1 +1 -1 -1   -4.0    1487     0.0
3 +1 -1 -1 +1 +1   -4.0    1084     0.0
4 -1 +1 +1 +1 -1   -2.0       1     0.0
5 +1 -1 -1 -1 +1   -2.0       1     0.0
['SPIN', 6 rows, 5000 samples, 5 variables]
```

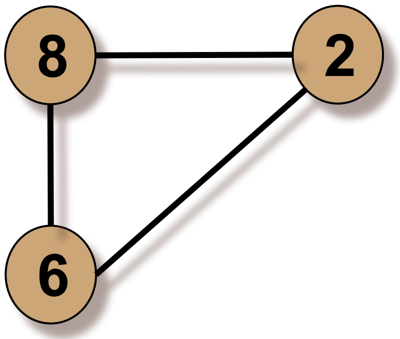


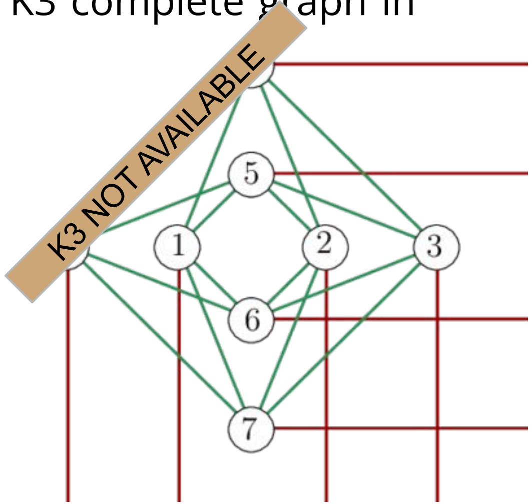
We obtained ALL the optimal solutions.

Observation: Problem of low connectivity

$$H = Z_2Z_6 + Z_6Z_8 + Z_8Z_2 + \text{...blablavar...}$$

That requires the following K3 complete graph in the hardware:



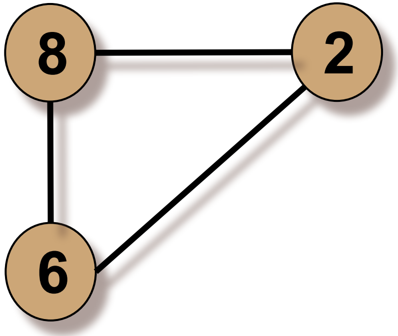


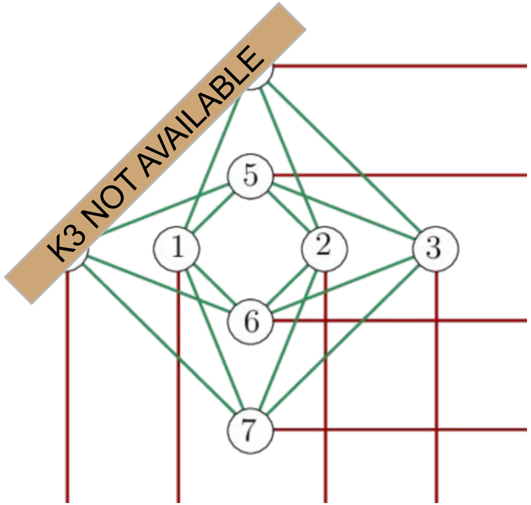
- Daniel Vert. Étude des performances des machines à recuit quantique pour la résolution de problèmes combinatoires. Thèse.
- Vert, D., Sirdey, R., & Louise, S. (2021). Benchmarking Quantum Annealing Against “Hard” Instances of the Bipartite Matching Problem. *SN Computer Science*, *2*(2), 1-12.

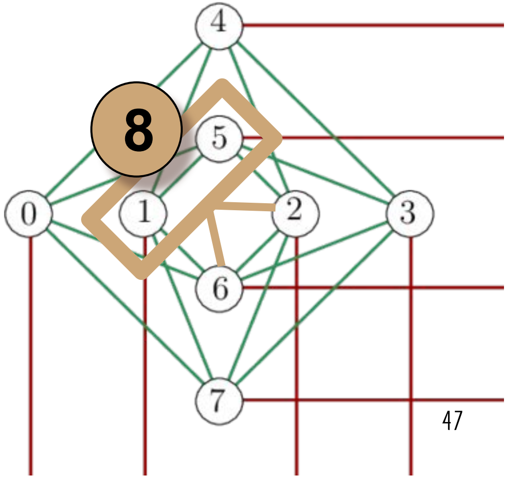
Problem of low connectivity

$H = Z_2Z_6 + Z_6Z_8 + Z_8Z_2 +$ ...blablavar...

That requires the following K3 complete graph in the hardware:

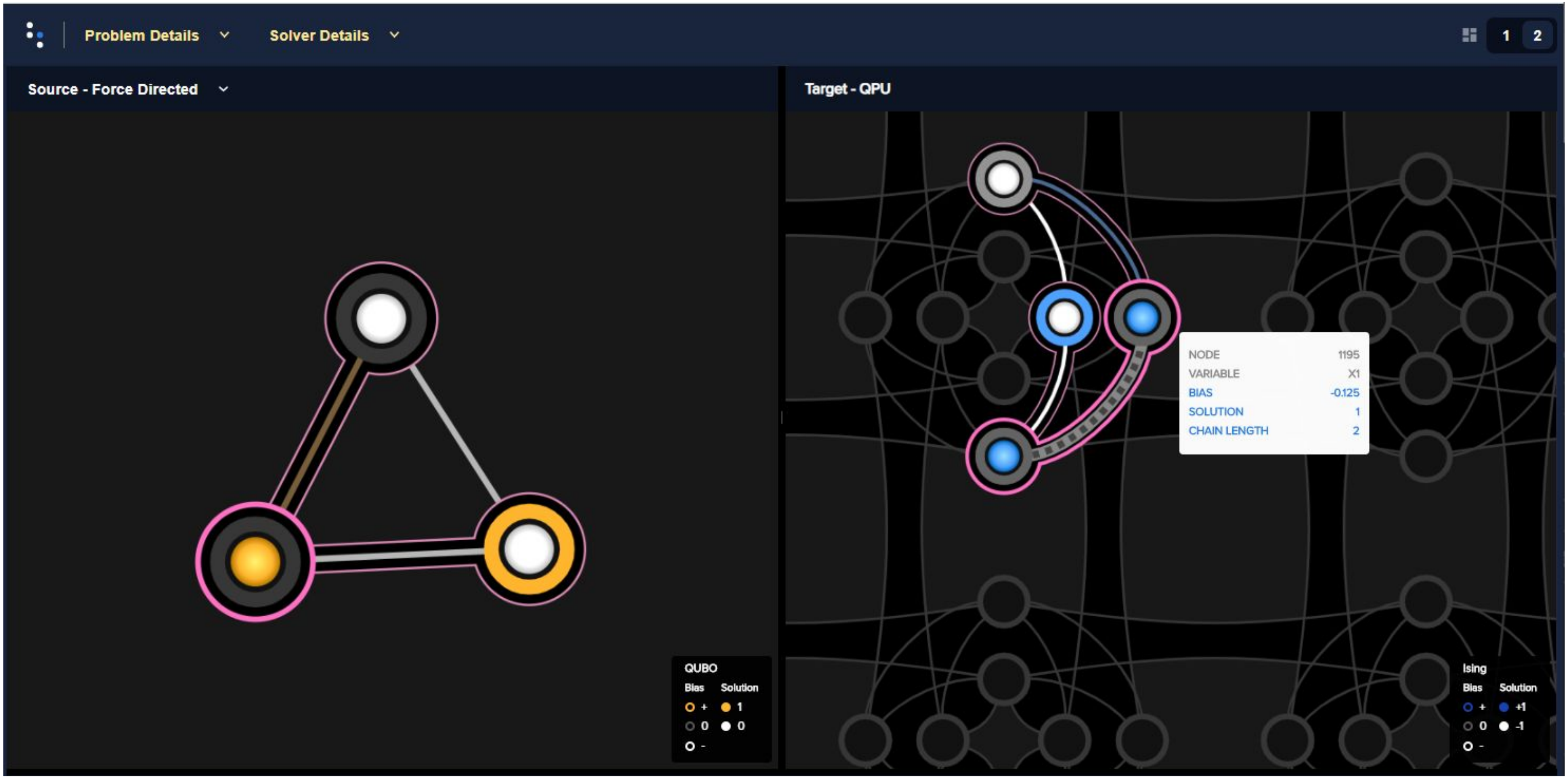




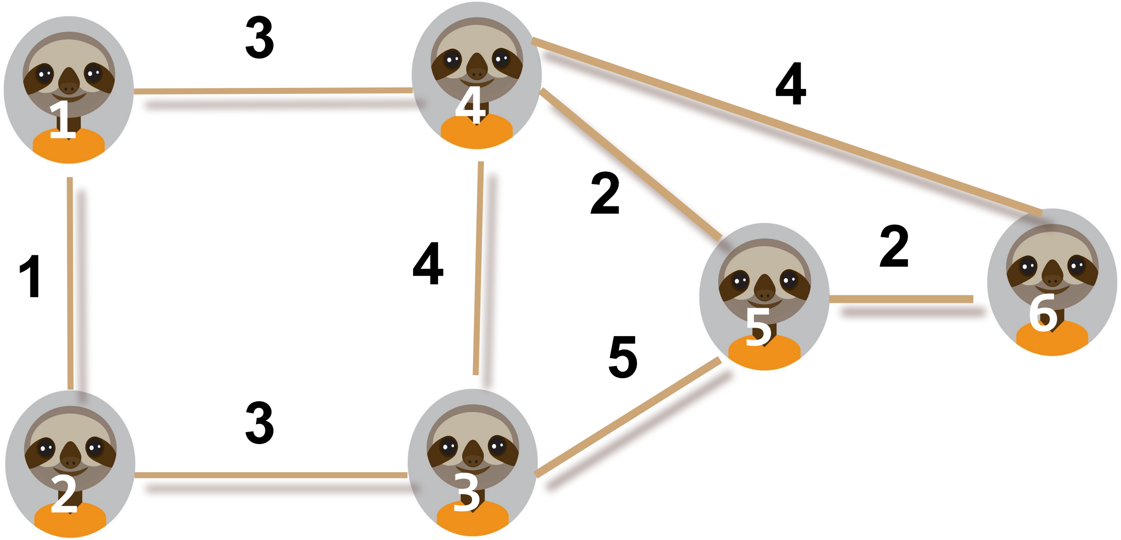


47

- Daniel Vert. Étude des performances des machines à recuit quantique pour la résolution de problèmes combinatoires. Thèse.
- Vert, D., Sirdey, R., & Louise, S. (2021). Benchmarking Quantum Annealing Against “Hard” Instances of the Bipartite Matching Problem. *SN Computer Science*, *2*(2), 1-12.

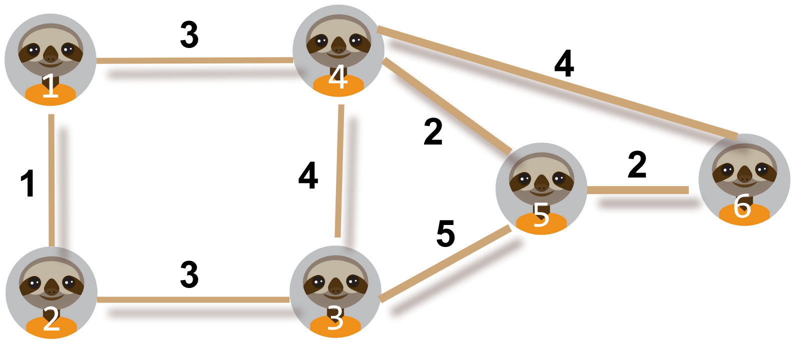

Problem Details
Solver Details
1
2
Source - Force Directed
Target - QPU
NODE 1195
VARIABLE X1
BIAS -0.125
SOLUTION 1
CHAIN LENGTH 2
QUBO
Bias
Solution
+
1
0
0
-
Ising
Bias
Solution
+
+1
0
-1
-

Lets solve a weighted case



A weighted case

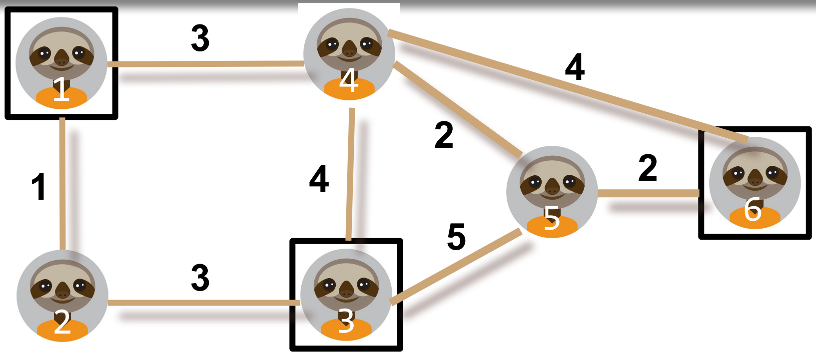
```
J = {(1,2):1, (1,4):3, (2,3):3, (4,3):4, (4,6):4, (4,5):2, (3,5):5, (5,6):2}
```



quantumDays.MaxCut.1.py

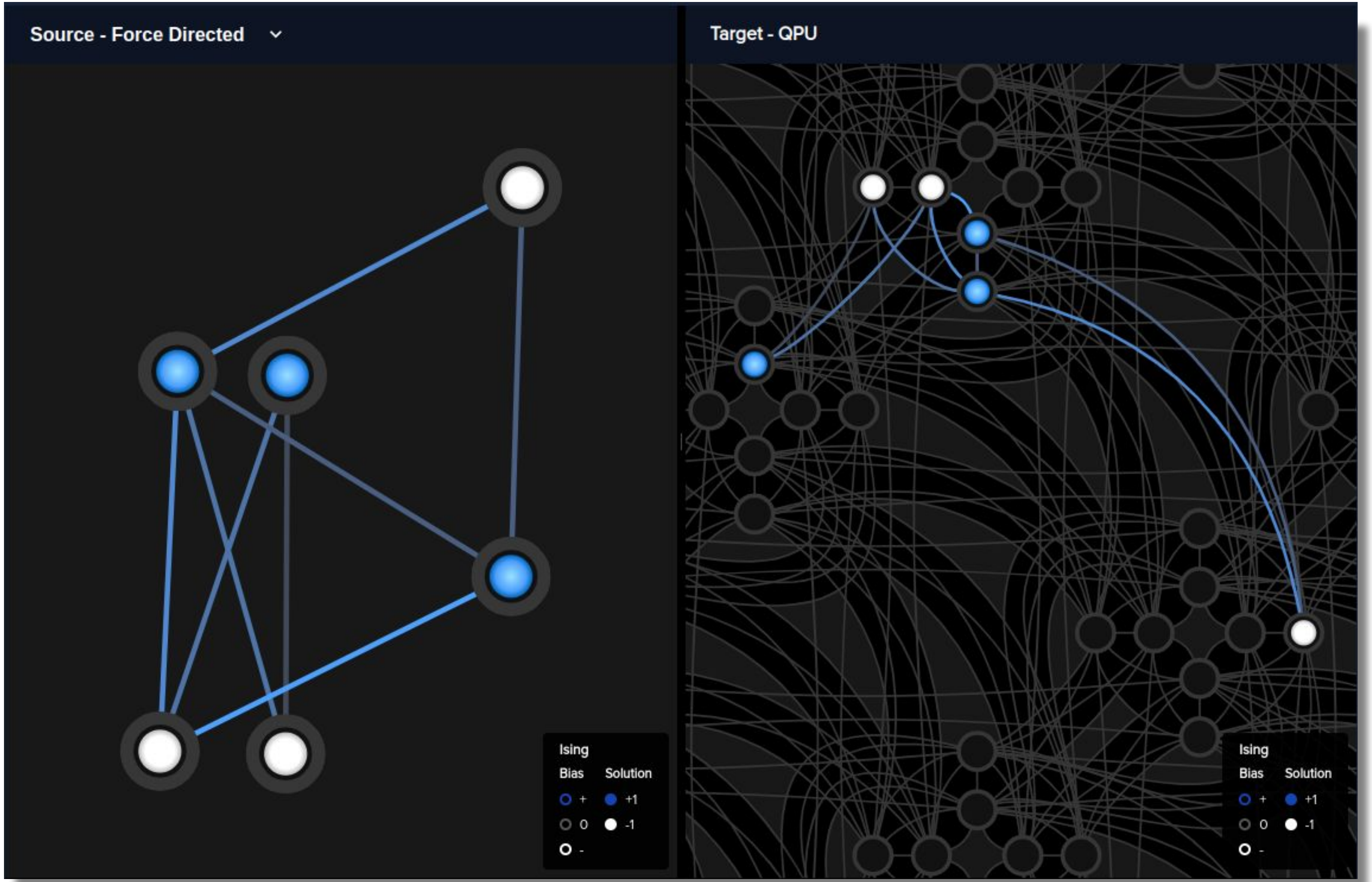
A weighted case

```
Leap IDE /workspace/DWAVEQuantum $ /usr/local/bin/python /workspace/DWAVEQuantum/2022.05.QuantumD
ys/quantumDays.MaxCut.1.py
Le modele resolu est le suivant :
BinaryQuadraticModel({1: 0.0, 2: 0.0, 4: 0.0, 3: 0.0, 6: 0.0, 5: 0.0}, {(2, 1): 1.0, (4, 1): 3.0,
(3, 2): 3.0, (3, 4): 4.0, (6, 4): 4.0, (5, 4): 2.0, (5, 3): 5.0, (5, 6): 2.0}, 0.0, 'SPIN')
The solution obtained by D-Wave's quantum annealer Advantage_system4.1 is
   1  2  3  4  5  6 energy num_oc. chain_.
0 -1 +1 -1 +1 +1 -1  -20.0    2991     0.0
1 +1 -1 +1 -1 -1 +1  -20.0    2004     0.0
2 -1 -1 +1 -1 -1 +1  -12.0       1     0.0
3 -1 -1 -1 +1 +1 -1  -12.0       2     0.0
4 +1 -1 +1 -1 +1 +1  -10.0       1     0.0
5 +1 -1 +1 -1 -1 -1   -8.0       1     0.0
['SPIN', 6 rows, 5000 samples, 6 variables]
Leap IDE /workspace/DWAVEQuantum $
```



quantumDays.MaxCut.1.py

A weighted case



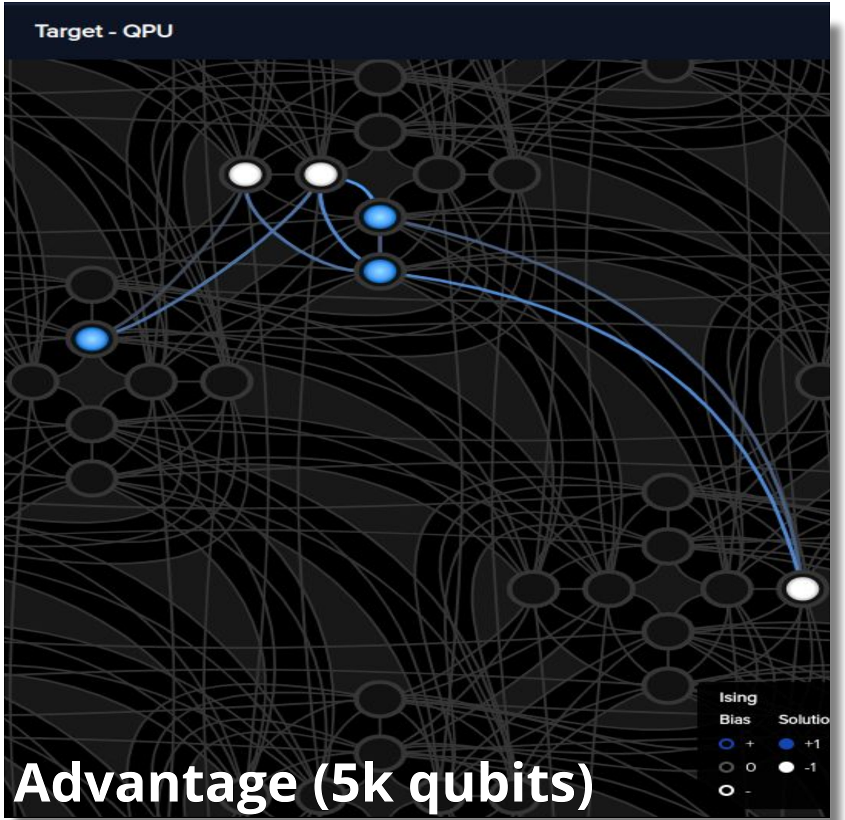
To solve with an older machine (2k qubits)

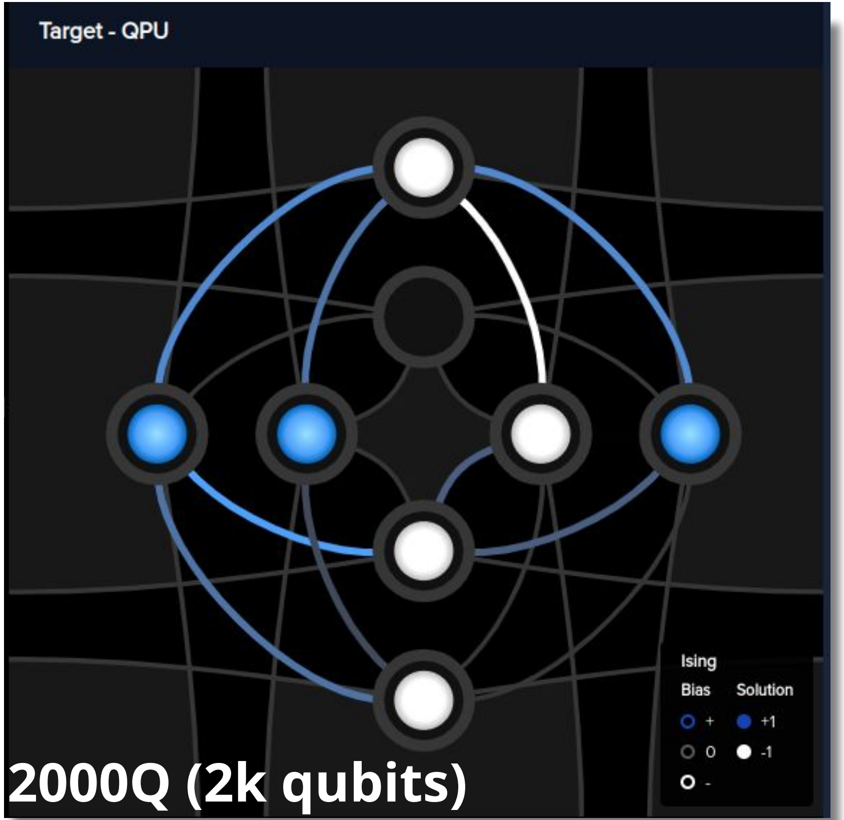
```
sampler =
EmbeddingComposite(DWaveSampler())
```

⬇

```
sampler =
EmbeddingComposite(DWaveSampler(solver='DW_2000Q_6'))
```

To solve with an older machine (2k qubits)





```
Leap IDE /workspace/DWAVEQuantum $ /usr/local/bin/python /workspace/DWAVEQuantum/2022.05.QuantumD
ys/quantumDays.MaxCut.1.py
Le modele resolu est le suivant :
BinaryQuadraticModel({1: 0.0, 2: 0.0, 4: 0.0, 3: 0.0, 6: 0.0, 5: 0.0}, {(2, 1): 1.0, (4, 1): 3.0,
(3, 2): 3.0, (3, 4): 4.0, (6, 4): 4.0, (5, 4): 2.0, (5, 3): 5.0, (5, 6): 2.0}, 0.0, 'SPIN')
The solution obtained by D-Wave's quantum annealer Advantage_system4.1 is
   1  2  3  4  5  6 energy num_oc. chain_.
0 -1 +1 -1 +1 +1 -1  -20.0    2991     0.0
1 +1 -1 +1 -1 -1 +1  -20.0    2004     0.0
2 -1 -1 +1 -1 -1 +1  -12.0       1     0.0
3 -1 -1 -1 +1 +1 -1  -12.0       2     0.0
4 +1 -1 +1 -1 +1 +1  -10.0       1     0.0
5 +1 -1 +1 -1 -1 -1   -8.0       1     0.0
['SPIN', 6 rows, 5000 samples, 6 variables]
Leap IDE /workspace/DWAVEQuantum $
```

Advantage (5k qubits)

```
Leap IDE /workspace/DWAVEQuantum $ /usr/local/bin/python /workspace/DWAVEQuantum/QuantumDays/quantumDays.MaxCut.1.py
Le modele resolu est le suivant :
BinaryQuadraticModel({1: 0.0, 2: 0.0, 4: 0.0, 3: 0.0, 6: 0.0, 5: 0.0}, {(2, 1): 1.0, (4, 1): 3.0, (3, 2): 3.0, (3, 4)
: 4.0, (6, 4): 4.0, (5, 4): 2.0, (5, 3): 5.0, (5, 6): 2.0}, 0.0, 'SPIN')
The solution obtained by D-Wave's quantum annealer DW_2000Q_6 is
   1  2  3  4  5  6 energy num_oc. chain_.
0 -1 +1 -1 +1 +1 -1  -20.0    3421     0.0
1 +1 -1 +1 -1 -1 +1  -20.0    1564     0.0
2 +1 +1 +1 -1 -1 +1  -12.0       1     0.0
3 +1 +1 -1 +1 +1 -1  -12.0       5     0.0
4 -1 -1 +1 -1 -1 +1  -12.0       5     0.0
5 -1 +1 -1 +1 -1 -1  -10.0       2     0.0
6 -1 +1 -1 +1 +1 +1   -8.0       1     0.0
7 +1 -1 +1 -1 -1 -1   -8.0       1     0.0
['SPIN', 8 rows, 5000 samples, 6 variables]
Leap IDE /workspace/DWAVEQuantum $
```

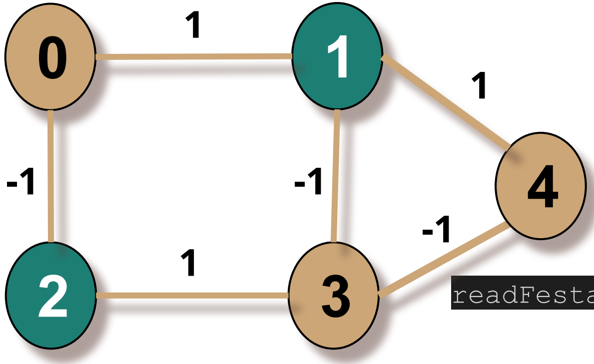
2000Q (2k qubits)

…and on literature instances?

quantumDays.MaxCut.LitInst.2.py

Few sets of instances here:

- https://grafo.etsii.urjc.es/optsicom/maxcut/#best-known-values
- Set 2 from “ad hoc selection” on the density (according to the qubits degree),
    - Test on sg3dl051000.mc (**vertices: 125; edges: 375; weights in {-1;1}; max cut = 110**).



Let’s try to obtain ‘110’ with the right number of anneals.

```
readFesta2002Instances("sg3dl051000.mc")
```

Ref:

- Festa, P., P. M. Pardalos, M. G. C. Resende, C. C. Ribeiro. 2002. Randomized heuristics for the max-cut problem. *Optim. Methods Software* 7 1033–1058 (where the instances are coming from)
- De Santis, M., Festa, P., Liuzzi, G., Lucidi, S., & Rinaldi, F. (2016). A nonmonotone grasp. *Mathematical Programming Computation*, *8*(3), 271-309 (where the optimal is given)

Result on the sg3dl051000.mc instance

- Number of Anneals:
  - `numberOfAnneals = 1`
  - The only cut obtain : **102**

```
The solution obtained by D-Wave's quantum annealer Advantage_system4.1 is
   1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 ... 125 energy num_oc. ...
0 -1 +1 -1 +1 +1 +1 +1 +1 +1 +1 -1 -1 +1 -1 -1 -1 +1 ...  -1 -203.0       1 ...
['SPIN', 1 rows, 1 samples, 125 variables]
{1: -1, 2: 1, 3: -1, 4: 1, 5: 1, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: -1, 12: -1, 13: 1,
4: -1, 15: -1, 16: -1, 17: 1, 18: -1, 19: 1, 20: -1, 21: 1, 22: 1, 23: 1, 24: 1, 25: 1,
26: 1, 27: -1, 28: 1, 29: -1, 30: 1, 31: 1, 32: -1, 33: 1, 34: -1, 35: 1, 36: 1, 37: 1,
38: -1, 39: -1, 40: 1, 41: -1, 42: 1, 43: -1, 44: 1, 45: 1, 46: 1, 47: -1, 48: -1, 49:
, 50: 1, 51: 1, 52: -1, 53: -1, 54: -1, 55: -1, 56: 1, 57: -1, 58: 1, 59: -1, 60: 1, 61
 -1, 62: 1, 63: -1, 64: -1, 65: 1, 66: -1, 67: -1, 68: -1, 69: 1, 70: 1, 71: 1, 72: 1,
3: 1, 74: 1, 75: 1, 76: 1, 77: 1, 78: -1, 79: 1, 80: -1, 81: 1, 82: 1, 83: 1, 84: -1, 8
: 1, 86: 1, 87: 1, 88: 1, 89: -1, 90: -1, 91: 1, 92: -1, 93: -1, 94: -1, 95: 1, 96: -1,
97: 1, 98: 1, 99: -1, 100: 1, 101: 1, 102: 1, 103: -1, 104: 1, 105: -1, 106: -1, 107: 1
 108: 1, 109: 1, 110: 1, 111: -1, 112: 1, 113: -1, 114: 1, 115: -1, 116: 1, 117: -1, 11
: -1, 119: 1, 120: -1, 121: -1, 122: -1, 123: -1, 124: -1, 125: -1}
102
Leap IDE /workspace/DWAVEQuantum $
```

  - `numberOfAnneals = 1000`
  - The only cut obtain : **108**

quantumDays.MaxCut.LitInst.2.py

Result on the sg3dl051000.mc instance

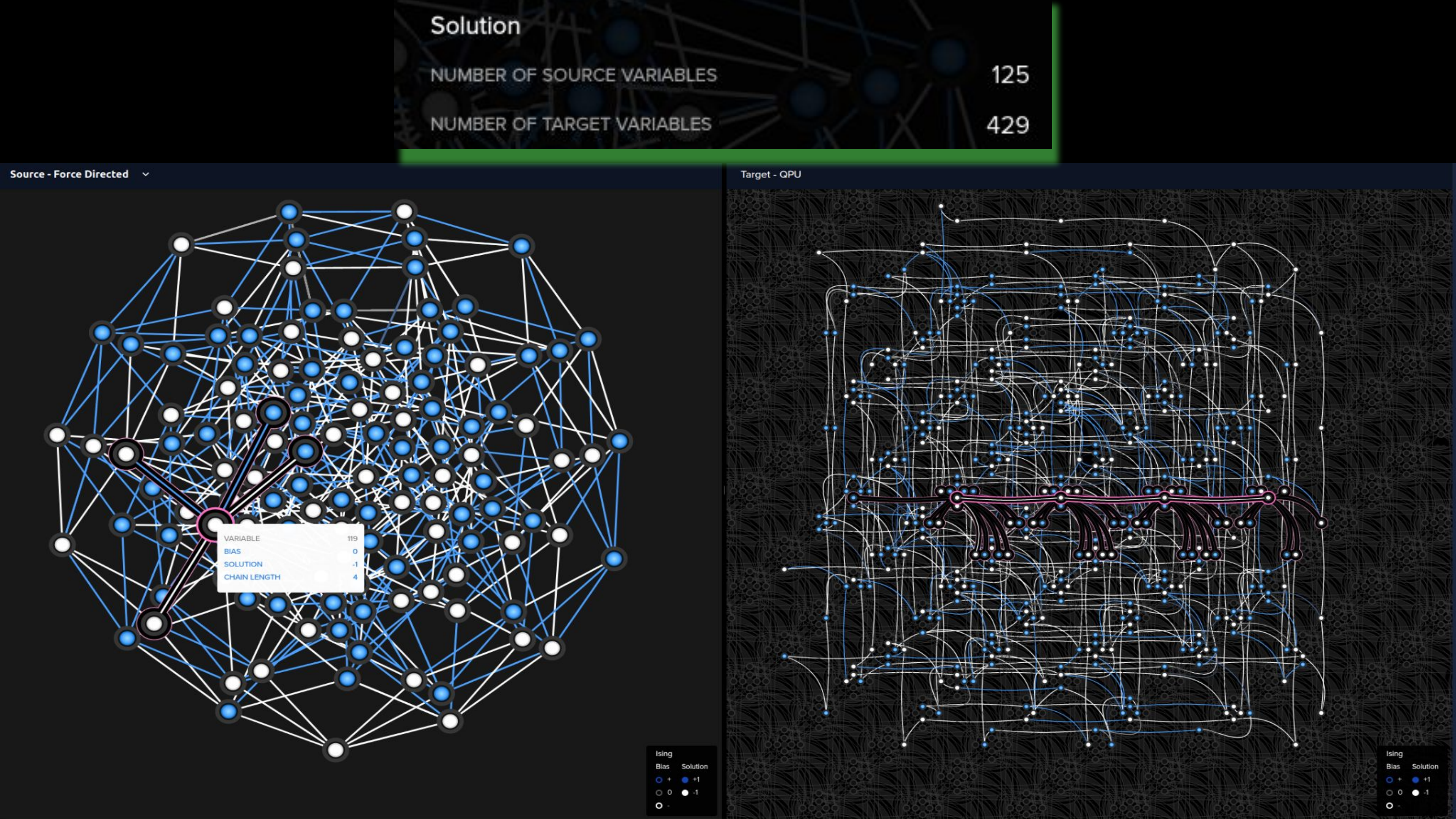
- Number of Anneals:
  - `numberOfAnneals = 2500`
  - The only cut obtained : **110 (optimal)**
  - Energy (min sum ZiZj) : -219

Timing	
QPU ACCESS TIME (µs)	468321.16
QPU PROGRAMMING TIME (µs)	15071.16
QPU SAMPLING TIME (µs)	453250
TOTAL POST PROCESSING TIME (µs)	7686
POST PROCESSING OVERHEAD TIME (µs)	835

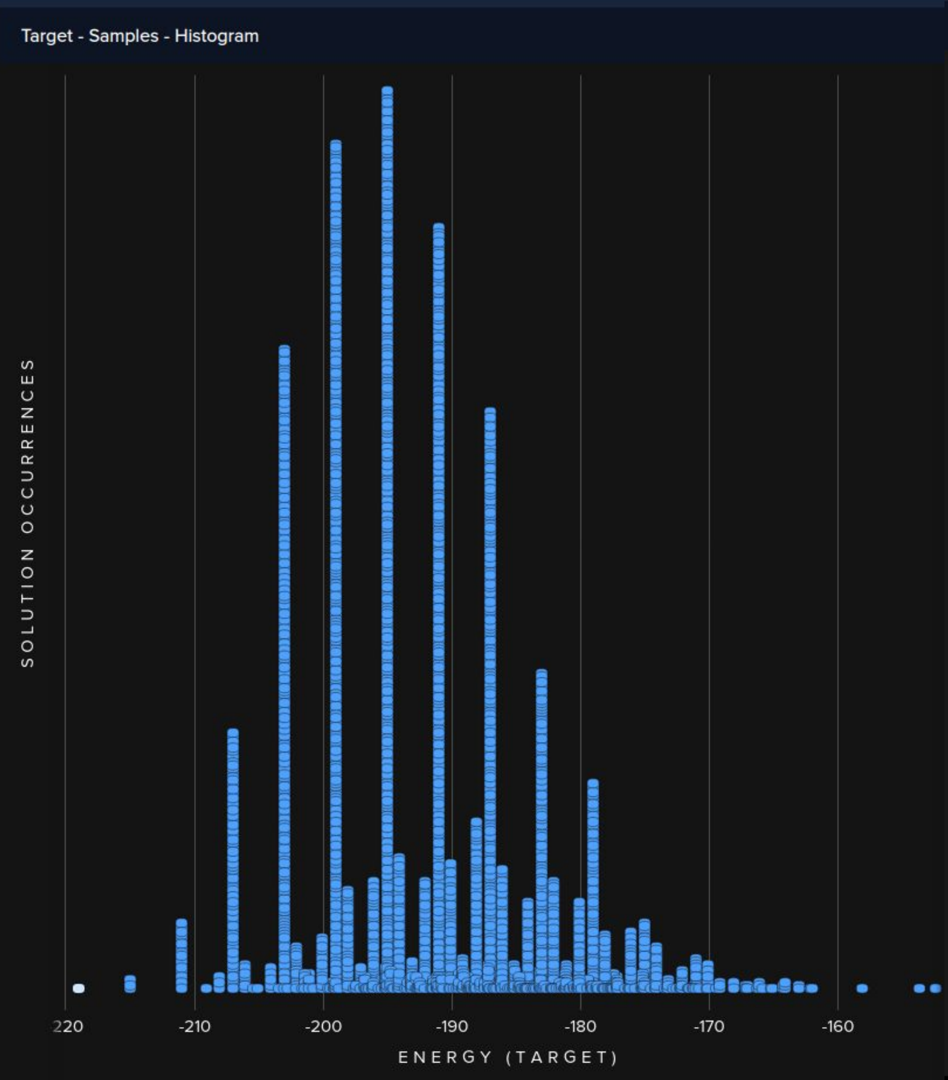
```
2497 -1 +1 -1 -1 +1 +1 +1 -1 -1 +1 -1 -1 -1 +1 -1 -1 ...  +1 -163.0       1 ...
['SPIN', 2500 rows, 2500 samples, 125 variables]
{1: -1, 2: -1, 3: 1, 4: -1, 5: 1, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1, 11: -1, 12: -1, 13: 1, 14: -1, 15: -1, 16: -1
, 17: 1, 18: -1, 19: -1, 20: -1, 21: 1, 22: 1, 23: 1, 24: -1, 25: -1, 26: 1, 27: 1, 28: -1, 29: 1, 30: -1, 31:
 1, 32: 1, 33: 1, 34: -1, 35: 1, 36: 1, 37: 1, 38: -1, 39: -1, 40: 1, 41: -1, 42: 1, 43: -1, 44: 1, 45: 1, 46:
 1, 47: 1, 48: -1, 49: 1, 50: 1, 51: 1, 52: 1, 53: -1, 54: -1, 55: -1, 56: 1, 57: -1, 58: 1, 59: -1, 60: 1, 61
: -1, 62: 1, 63: -1, 64: -1, 65: 1, 66: -1, 67: -1, 68: -1, 69: 1, 70: 1, 71: 1, 72: -1, 73: 1, 74: 1, 75: 1,
76: 1, 77: 1, 78: -1, 79: 1, 80: -1, 81: 1, 82: 1, 83: 1, 84: -1, 85: 1, 86: 1, 87: 1, 88: 1, 89: -1, 90: -1,
91: -1, 92: -1, 93: -1, 94: -1, 95: 1, 96: 1, 97: -1, 98: 1, 99: -1, 100: 1, 101: 1, 102: 1, 103: -1, 104: 1,
105: -1, 106: -1, 107: 1, 108: 1, 109: 1, 110: 1, 111: 1, 112: 1, 113: -1, 114: 1, 115: -1, 116: -1, 117: -1,
118: -1, 119: -1, 120: 1, 121: -1, 122: -1, 123: -1, 124: -1, 125: 1}
110
```

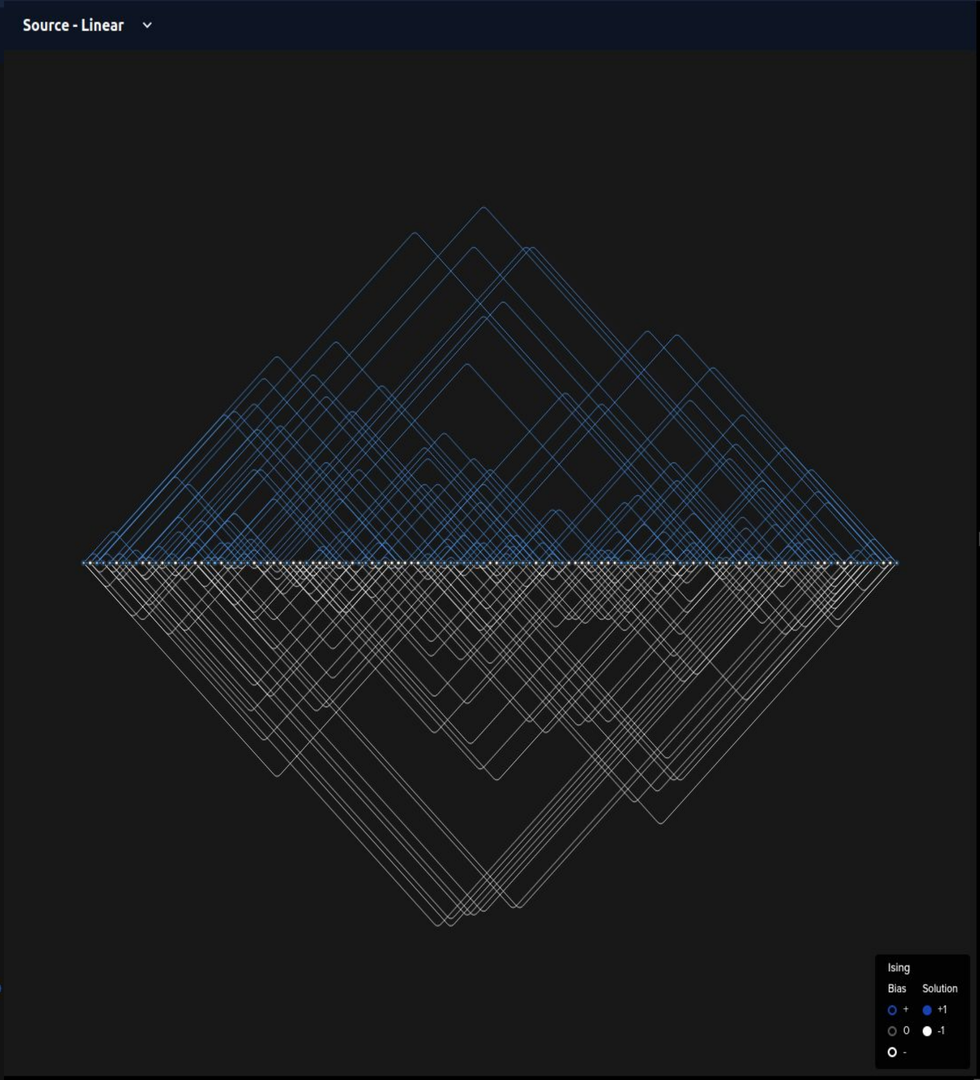
maxCutResult.txt

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	...	125	energy	num_oc.	...
0	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
1	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	-1	-1	-1	+1	+1	-1	...	+1	-219.0	1	...
2	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
3	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
4	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
5	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	-1	...	+1	-219.0	1	...
6	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
7	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	-1	...	+1	-219.0	1	...
8	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
9	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
10	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
11	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
12	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
13	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	+1	+1	+1	-1	...	+1	-219.0	1	...
14	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
15	-1	-1	+1	-1	+1	+1	+1	+1	+1	+1	-1	-1	+1	-1	-1	-1	...	+1	-219.0	1	...
16	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
17	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
18	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	-1	...	+1	-219.0	1	...
19	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	-1	...	+1	-219.0	1	...
20	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
21	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
22	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	-1	-219.0	1	...
23	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	-1	-1	-1	+1	+1	-1	...	+1	-219.0	1	...
101	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
106	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
360	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-219.0	1	...
24	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-215.0	1	...
25	+1	+1	-1	+1	-1	-1	-1	-1	-1	-1	+1	+1	-1	+1	+1	+1	...	+1	-215.0	1	...



Solution
NUMBER OF SOURCE VARIABLES 125
NUMBER OF TARGET VARIABLES 429
Source - Force Directed
Target - QPU
VARIABLE 119
BIAS 0
SOLUTION -1
CHAIN LENGTH 4
Ising
Bias
Solution
+
+1
0
-1
-
Ising
Bias
Solution
+
+1
0
-1
-


Target - Samples - Histogram
SOLUTION OCCURRENCES
220
-210
-200
-190
-180
-170
-160
ENERGY (TARGET)

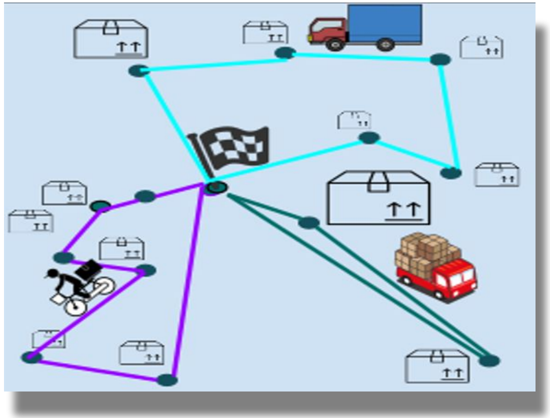

Source - Linear
Ising
Bias
Solution
+
+1
0
-1
-

Future work on the max-cut problem

- Experiment other formulations
    - Nguyen, V.H., Minoux, M. Linear size MIP formulation of Max-Cut: new properties, links with cycle inequalities and computational results. *Optim Lett* **15,** 1041–1060 (2021). https://doi.org/10.1007/s11590-020-01667-z

- BUT… Still bounded by either the density of the graph (connectivity of the qubits -15-) or the number of the vertices (≈5000 qubits)

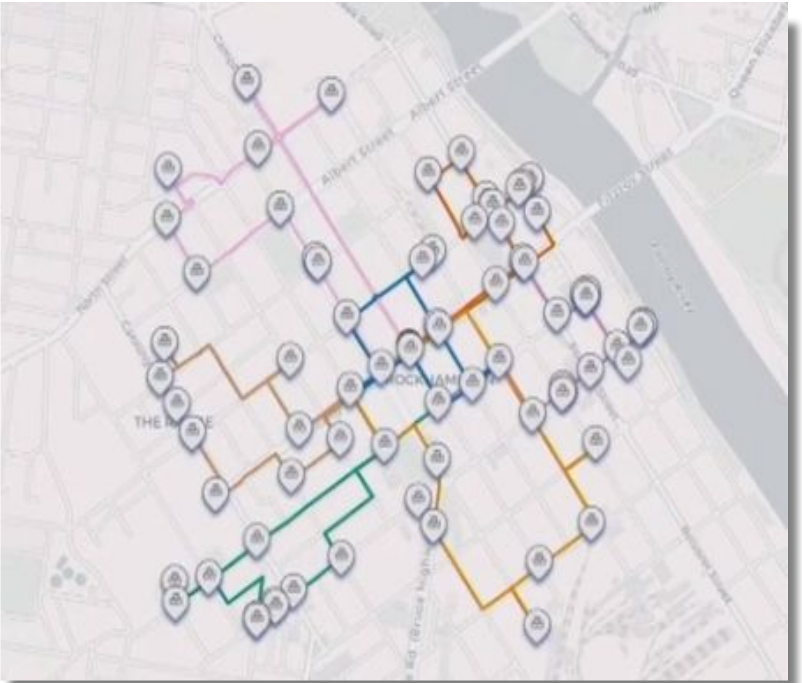
Outline



5 The Capacitated Vehicles Routing Problem

NEC experiment on the CVRP

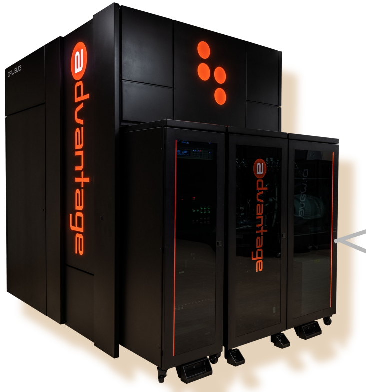
(from David Garvin, Nec Australia at Qubits conf. 2021-10)



Constraints: from one depot node, supply of the other nodes with 1 of the **K vehicles** with a capacity,
Objective: Min total distance.

NEC experiment

- Resolution Scheme:
  - 1. **Clustering**
    - quadratic formulation
    - quantum computer
  - 2. **Routing** (for each vehicle)
    - basic heuristic solving the TSP
    - classical computer



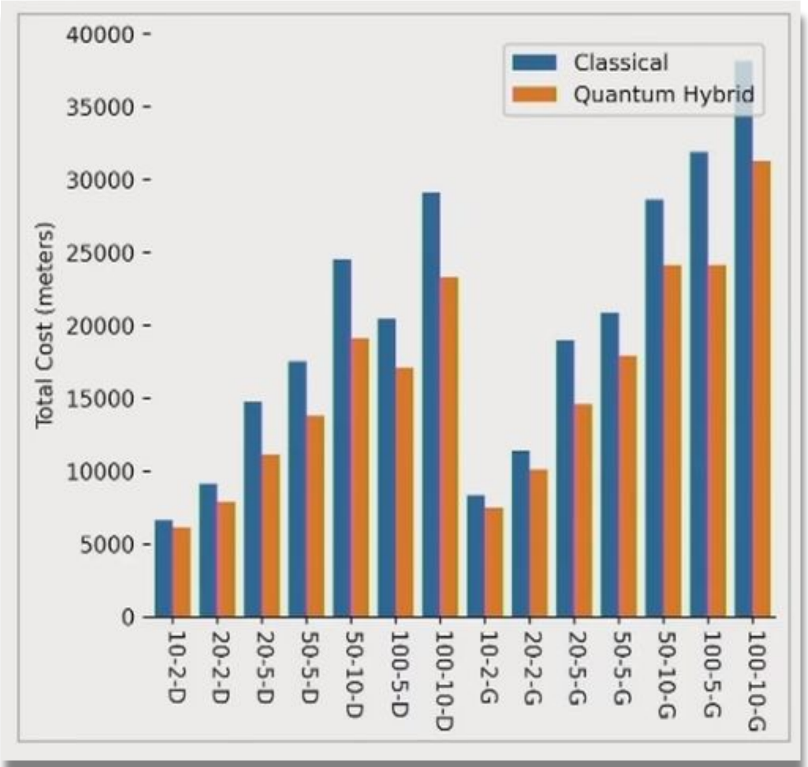
Our XP

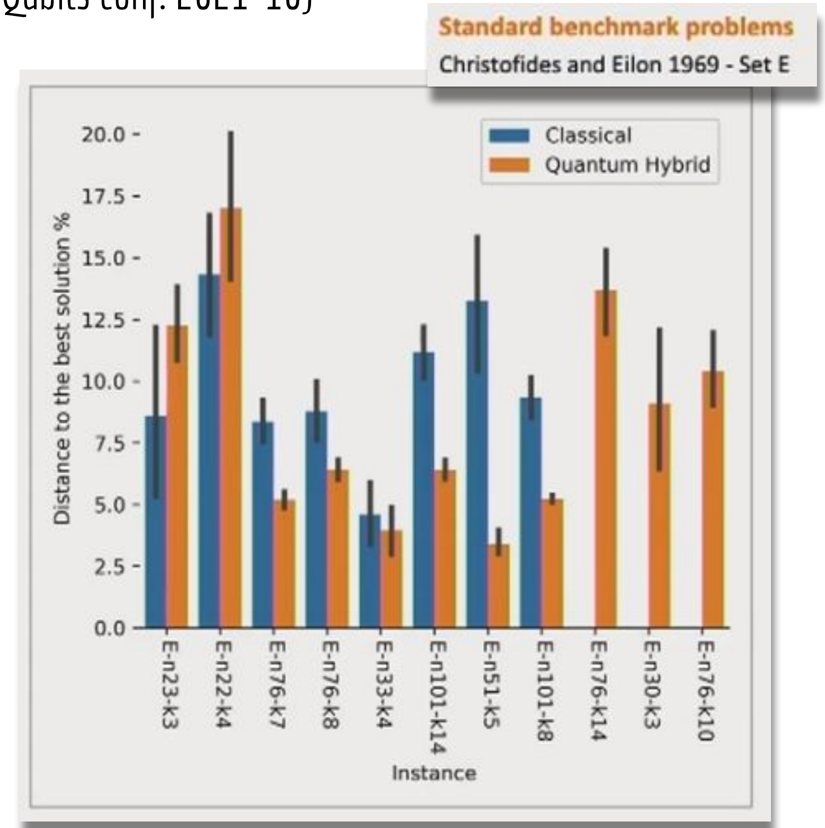
- Resolution Scheme:
  - 1. **Clustering**
    - quadratic formulation
    - quantum computer
  - 2. **Routing** (for each vehicle)
    - quadratic formulation
    - quantum computer

(A work with 4 Master Students (JUNIA):
Xavier Bertauld, Florent Descamps,
Téo Gras and Nicolas Tinel.)

NEC experiment: CVRP

(David Garvin, Nec Australia at Qubits conf. 2021-10)





Approximate the CVRP

a Master/Slaves scheme

We denote by:

- K the set of vehicles and k a vehicle index such that $k = 1..|K|$,
- N the set of cities and i and j two city indexes such that $i, j = 1..|N|$ while p denotes the city position in the tour of the related vehicle.
- u_i the size of the package to deliver in city i,
- C_k the capacity of the vehicle k,
- c_{ij} the cost for going from city i to city j.

The binary variables can be defined as follows:

- Variable x_{ik} assumes value 1 if the vehicle k visits city i and 0 otherwise.
- Variable r_{ip} is equal to 1 if the city i is the p^{th} visited city and 0 otherwise.

Clustering

$$Min \sum_{k} \sum_{i} \sum_{j<i} c_{ij} x_{ik} x_{jk}$$

$$\sum_{i} u_i x_{ik} \leq C_k, \quad \forall k$$

$$\sum_{k} x_{ik} = 1, \quad \forall i$$

|K| TSPs

TSP formulation

$$Min \sum_{p}^{n-1} \sum_{i} \sum_{j \neq i} c_{ij} r_{ip} r_{jp+1}$$

n is the number of cities of the related cluster.

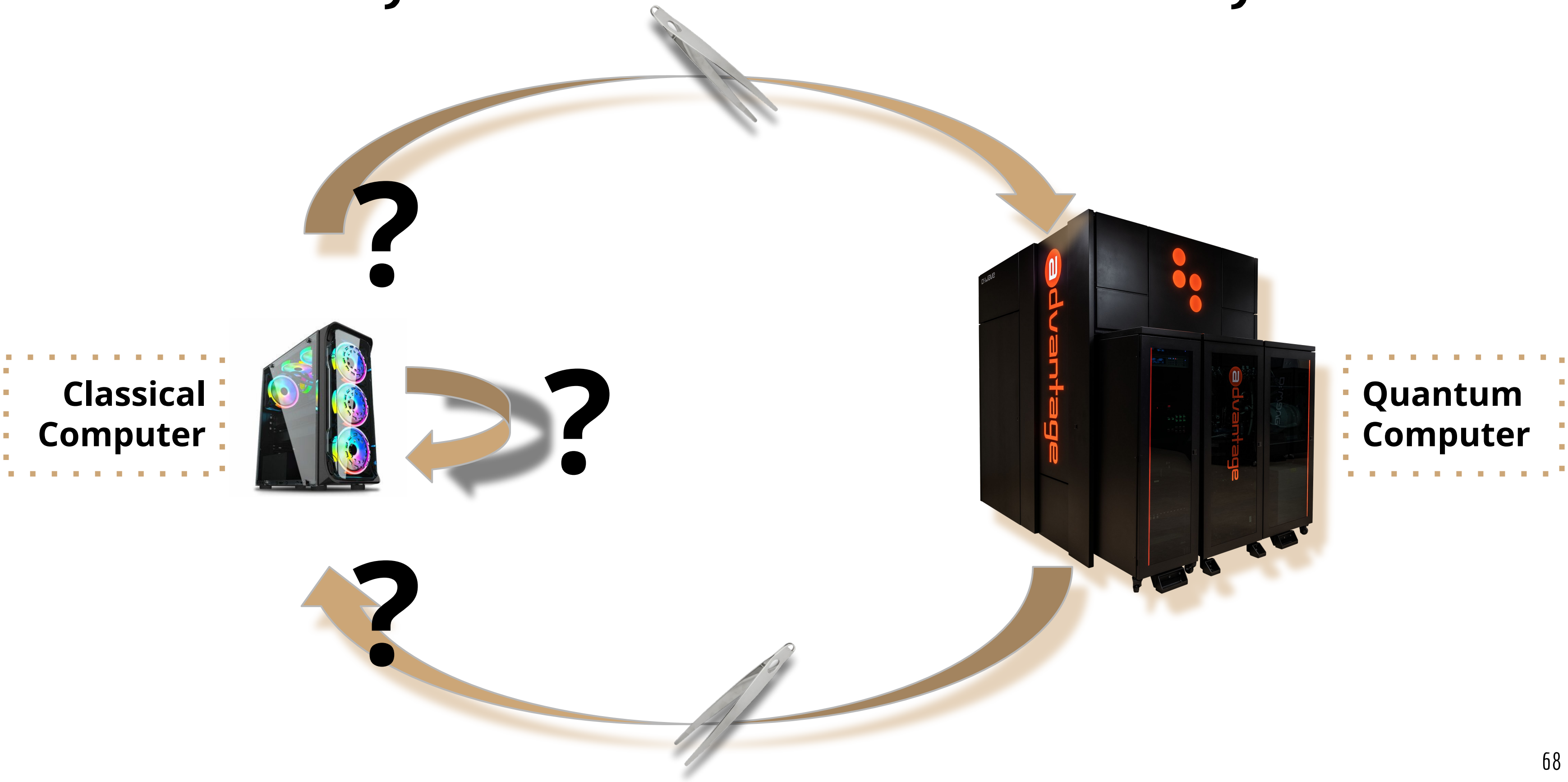
$$\sum_{p} r_{ip} = 1, \quad \forall i$$

$$\sum_{i} r_{ip} = 1, \quad \forall p$$

CVRP: let’s code it!

(for D-Wave **hybrid** machines)

The D-Wave Hybrid machines and their mysteries



Librairies

```
from dimod import ConstrainedQuadraticModel, CQM, SampleSet
from dimod import Binary, quicksum
from dwave.system import LeapHybridCQMSampler
from dwave.cloud.client import Client
from dwave.cloud import config
import numpy as np
import pandas as pd
import ast
import itertools
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import random
import math
from scipy.spatial import distance
import time
import re
```

quantumDaysCVRP.py

Cluster

```
# Define our model
  cqm=ConstrainedQuadraticModel()
```

`cqm` pour constrained quadratic model

Variable x_{ik} assumes value 1 if the vehicle k visits city i and 0 otherwise.

```
  # Preparation of our matrix that
  # will got our solution
  x = {
   (i, d): Binary('x{}_{}'.format(i, d))
  for i in range(nbOfPointToCluster)
  for d in range(nbOfCluster)}
```

quantumDaysCVRP.py

CLuster

```
# ----------------------------------------------
#                Objective function:
# ----------------------------------------------
objective = quicksum(matrixOfCost[i][j] * x[(i,d)] * x[(j,d)]
    for i in range(nbOfPointToCluster)
    for j in range(i+1, nbOfPointToCluster)
    for d in range(nbOfCluster) )

cqm.set_objective(objective)
```

Objective function

$$Min \sum_{k} \sum_{i} \sum_{j<i} c_{ij} x_{ik} x_{jk}$$

cqm for constrained quadratic model

quantumDaysCVRP.py

CLuster

```
    # --------------------------------
    #    subject to the constraints:
    # --------------------------------
    #We want the depot in every cluster
    for d in range(nbOfCluster):
        cqm.add_constraint(x[(0,d)] == 1)
```

`Constraints`

`add_constraint` for adding the depot node in each cluster.

quantumDaysCVRP.py

CLuster

```
# The sum of the capacity require by the point
# should not exceed the total capacity of the cluster
for d in range(nbOfCluster):
    cqm.add_constraint(quicksum(vectorOfVolume[i] * x[(i,d)]
    for i in range(nbOfPointToCluster)) <= vectorOfCapacity[d])


#Every point should be in 1 and only 1 cluster except the depot
for i in range(1,nbOfPointToCluster):
    cqm.add_constraint(quicksum(x[(i,d)]
    for d in range(nbOfCluster)) == 1)
```

Constraints

$$\sum_i u_i x_{ik} \leq C_k, \quad \forall k$$

$$\sum_k x_{ik} = 1, \quad \forall i$$

quantumDaysCVRP.py

CLuster

Resolution & data analysis

```
#We get our solution
cqm_sampler=LeapHybridCQMSampler()
sampleset=cqm_sampler.sample_cqm(cqm)
```

```
#We transform it in a panda dataframe
dataFrame = sampleset.to_pandas_dataframe(sample_column=True)
dataFrame = dataFrame[['sample','energy','is_feasible']]
dataFrame = dataFrame.sort_values(by = 'energy')
dataFrame.to_csv("clustering.csv")
```

quantumDaysCVRP.py

Some XP on "home" instance

Break on the clusters

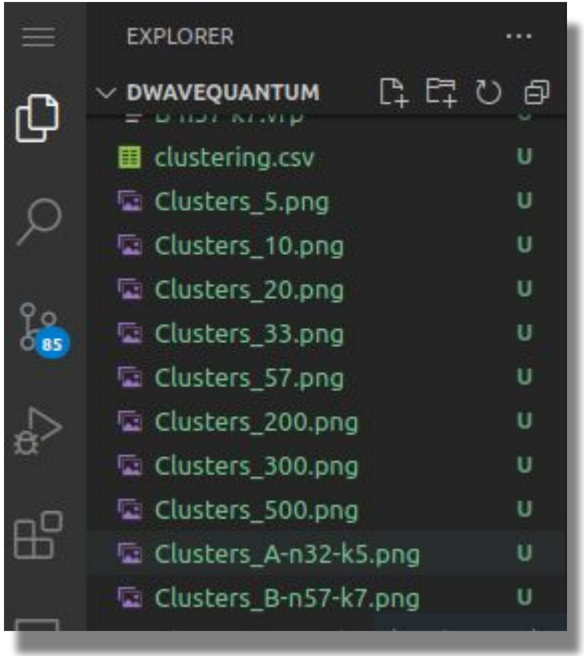
```
selfgeneration(numberOfVehicules, numberOfCity, capaConsumptionMin, capaConsumptionMax):
[...]
```

Instance Generation

- (X,Y) clients randomly generated in a [50 ; 50] square
- `capacityOfCarInt = math.ceil( (capaConsumptionMax * numberOfCities) / numberOfCars)`

Some XP on a "home" instance

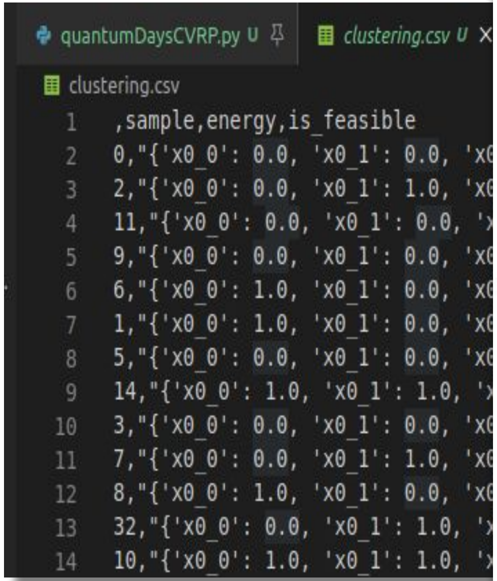
Break on the clusters

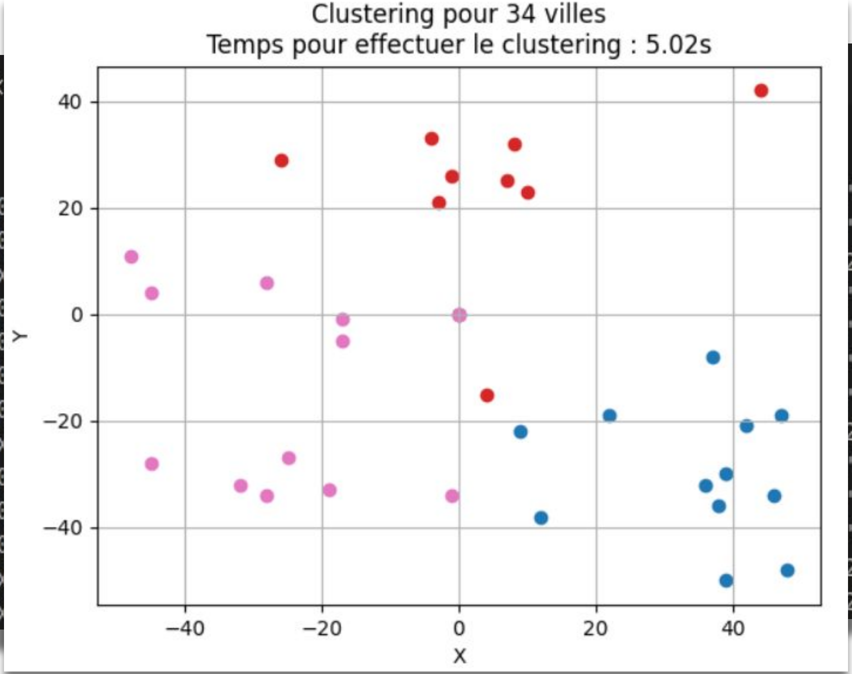


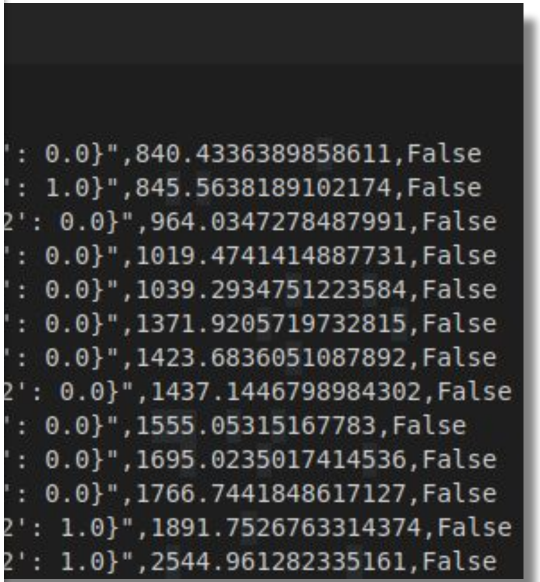
```
numberOfCars          = 3
numberOfCities        = 33
capaConsumptionMin    = 1
capaConsumptionMax    = 4
```

Some XP on a "home" instance

Break on the clusters







TSP

```
cqm=ConstrainedQuadraticModel()
```

Variable r_{ip} is equal to 1 if the city i is the p^{th} visited city and 0 otherwise.

```
#Preparation of our variables
x = {
(c, p): Binary('x{}_{}'.format(c, p))
for c in range(nbOfPoint)
for p in range(nbOfPoint+1)}
#+1 cause depository take the first and last position
```

Variables

(x in the code, r in the model)

quantumDaysCVRP.py

Objective Function

$$Min \sum_{p}^{n-1} \sum_{i} \sum_{j \neq i} c_{ij} r_{ip} r_{jp+1}$$

```
#Objective function
objective = quicksum(matrixOfCost[c1][c2] * x[(c1,p)] * x[(c2,p+ 1)]
        for c1 in range(nbOfPoint)
        for c2 in range(nbOfPoint)
        for p in range(nbOfPoint) )
    #No need to put -1 because we got 1 extra position compare to the number of city
cqm.set_objective(objective)
```

quantumDaysCVRP.py

```
#Every position needs to get only 1 city
for p in range(nbOfPoint):
    cqm.add_constraint(quicksum(x[(c,p)]
    for c in range(nbOfPoint)) == 1)
```

```
#Every city needs to have only 1 position
for c in range(1,nbOfPoint):
    cqm.add_constraint(quicksum(x[(c,p)]
    for p in range(nbOfPoint)) == 1)
```

Constraints

(x in the code, r in the model)

$$\sum_p r_{ip} = 1, \quad \forall i$$

$$\sum_i r_{ip} = 1, \quad \forall p$$

quantumDaysCVRP.py

Resolution & data analysis

```
#Get the solution
cqm_sampler=LeapHybridCQMSampler()
sampleset=cqm_sampler.sample_cqm(cqm)
```

```
#Transform the solution in a panda dataframe
dataFrame = sampleset.to_pandas_dataframe(sample_column=True)
dataFrame = dataFrame[['sample','energy','is_feasible']]
dataFrame = dataFrame.sort_values(by = 'energy')
#Save in a .csv
dataFrame.to_csv(fileName)
```

quantumDaysCVRP.py

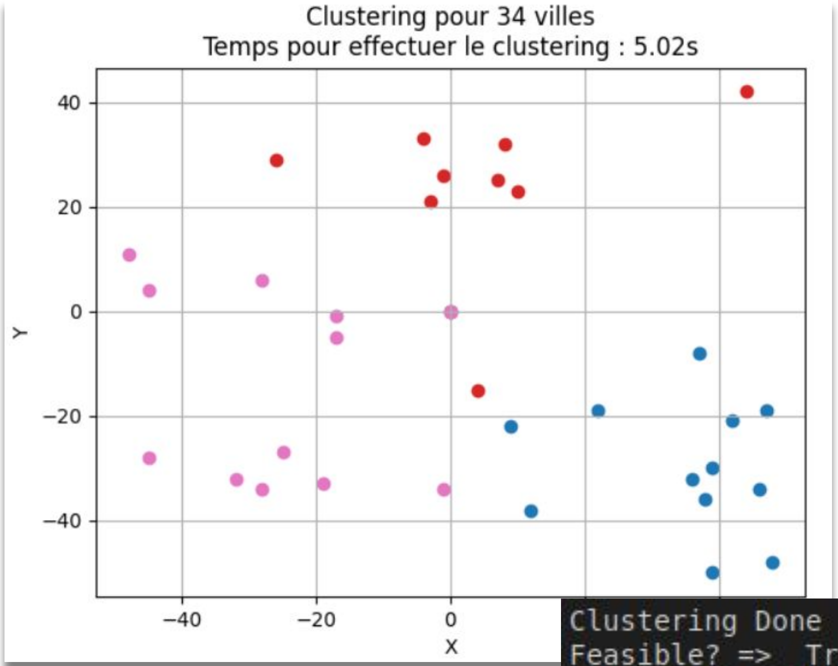
for each TSP

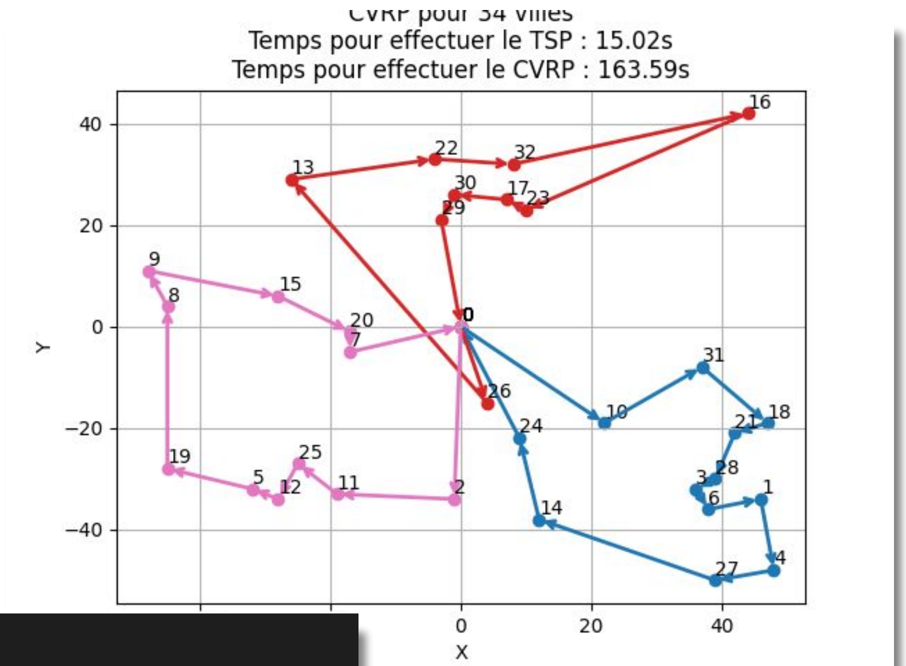
```
A TSP for each cluster
```

```
    for i in range (len(listClusters)):
        TSPTimer +=
TSP(len(listClusters[i]),clusteurCostMatrix[i],
                                    str(i)+".csv")
```

```
numberOfCars          = 3
numberOfCities        = 33
capaConsumptionMin    = 1
capaConsumptionMax    = 4
```

Some XP on “home” instances

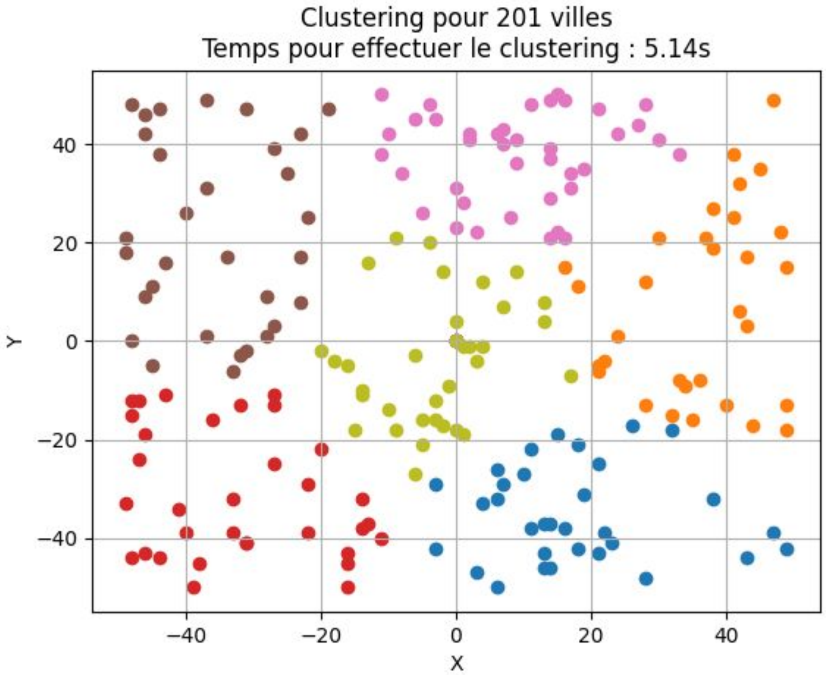


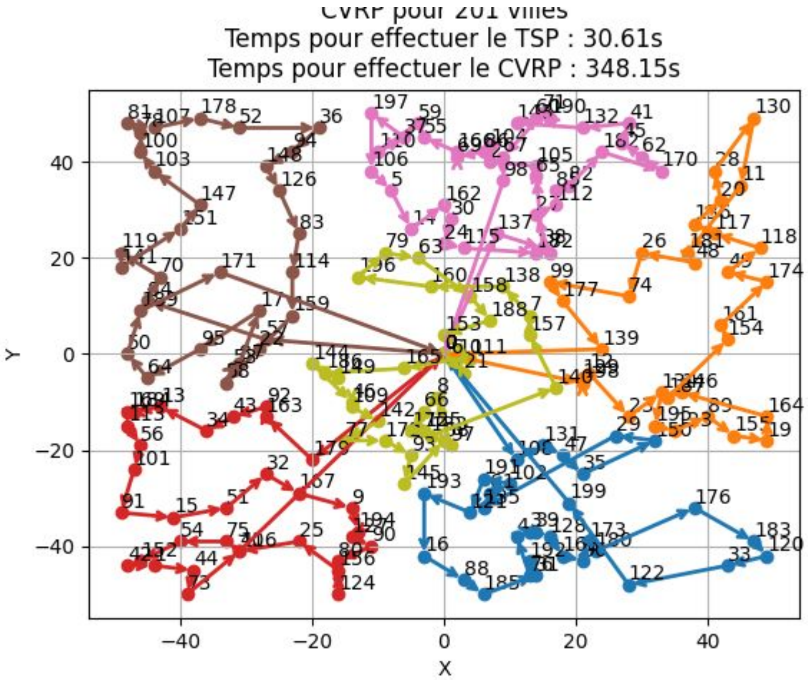



```
Clustering Done
Feasible? =>  True
D-Wave Hybrid Resolution: 585.6540996201489
```

```
numberOfCars          = 6
numberOfCities        = 200
capaConsumptionMin    = 1
capaConsumptionMax    = 4
```

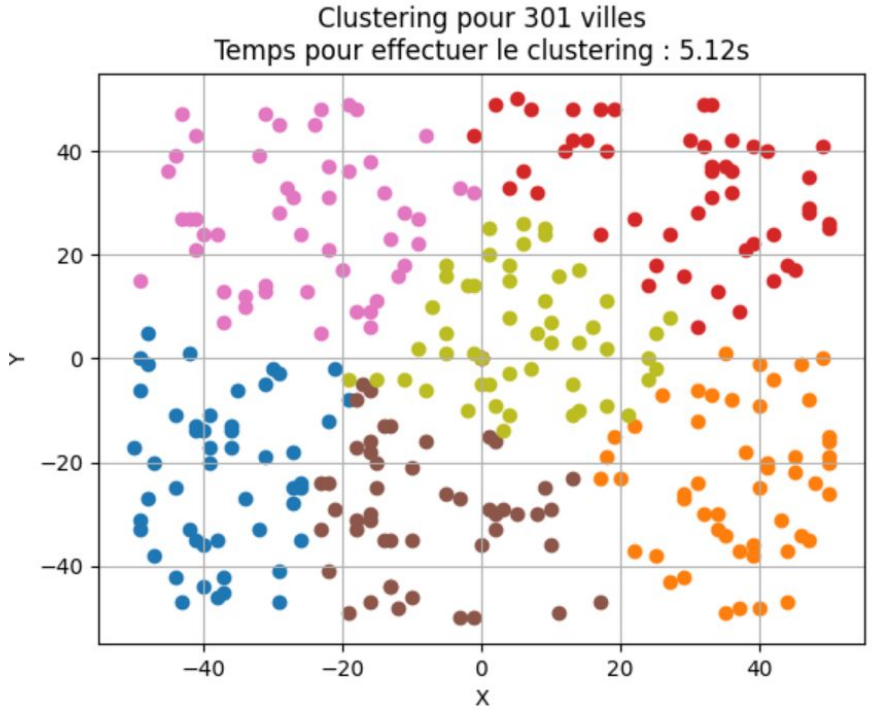
Some XP on "home" instances

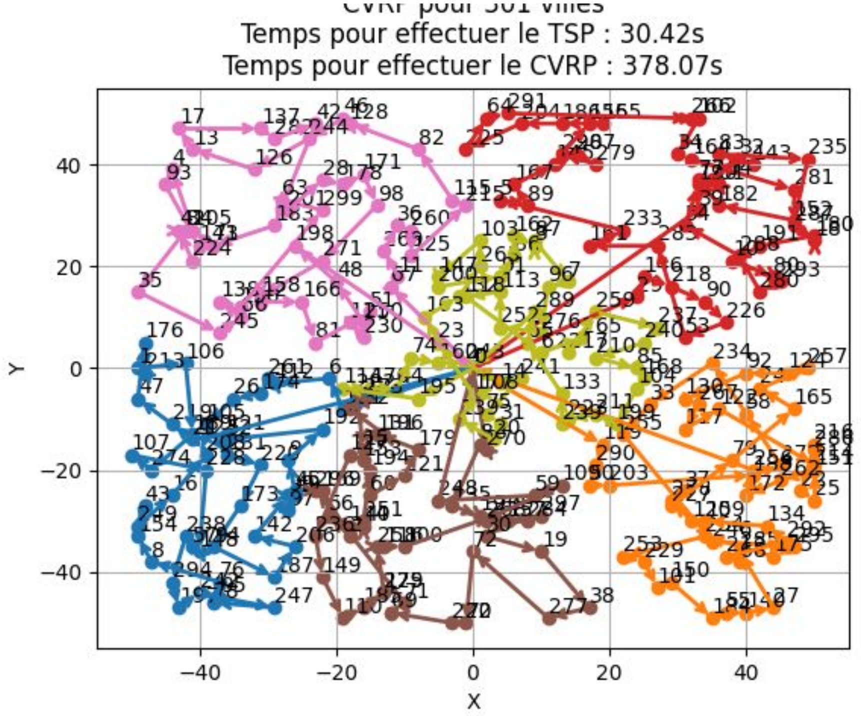




```
numberOfCars          = 6
numberOfCities        = 300
capaConsumptionMin    = 1
capaConsumptionMax    = 4
```

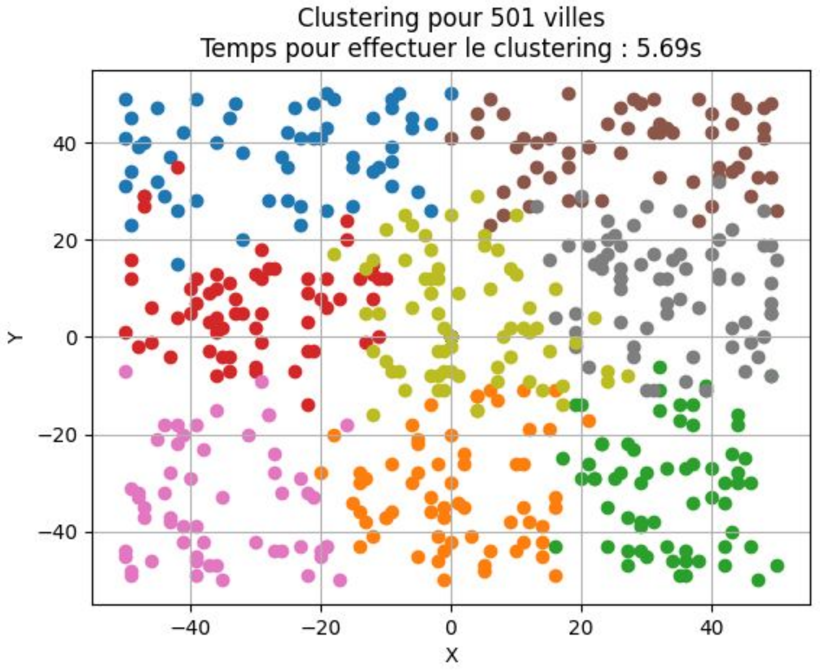
Some XP on "home" instances

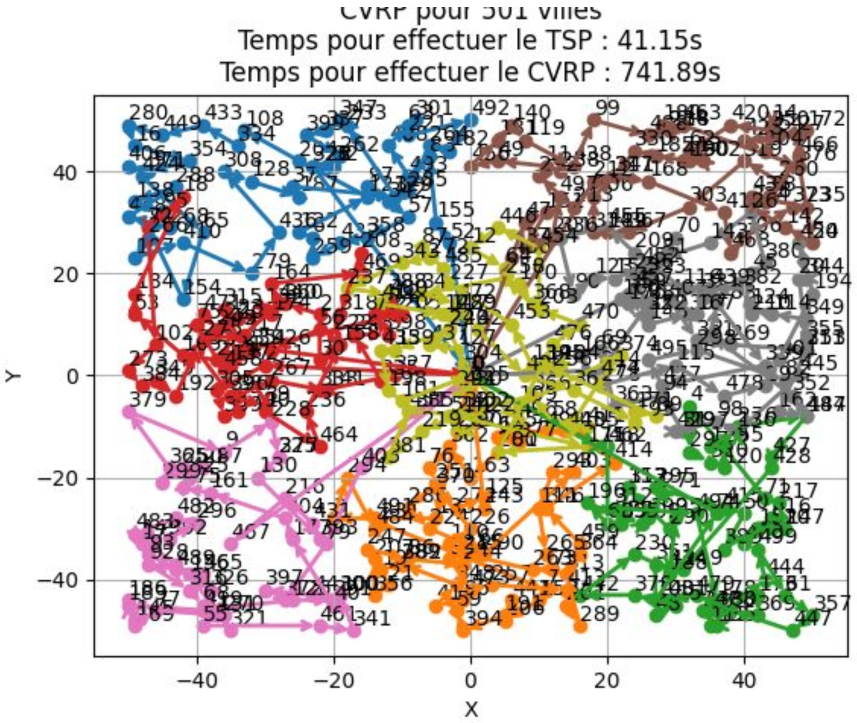




```
numberOfCars          = 8
numberOfCities        = 500
capaConsumptionMin    = 1
capaConsumptionMax    = 4
```

Some XP on “home” instances





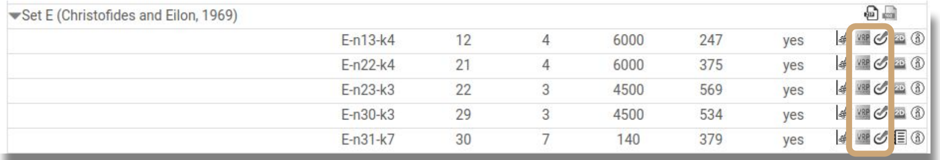
Hybrid machine limits

```
numberOfCars          = 10
numberOfCities        = 800
```

```
Traceback (most recent call last):
  File "/workspace/DWAVEQuantum/2022.05.QuantumDays/quantumDaysCVRP.py", line 768, in <module>
    selfgeneration(numberOfCars, numberOfCities, capaConsumptionMin, capaConsumptionMax)
  File "/workspace/DWAVEQuantum/2022.05.QuantumDays/quantumDaysCVRP.py", line 630, in selfgeneration
    ClusterTimer = Classification(numberOfCity,len(capacityOfCar),c2,capacityOfCar,volume)
  File "/workspace/DWAVEQuantum/2022.05.QuantumDays/quantumDaysCVRP.py", line 150, in Classification
    sampleset=cqm_sampler.sample_cqm(cqm)
  File "/usr/local/lib/python3.9/site-packages/dwave/system/samplers/leap_hybrid_sampler.py", line 731,
in sample_cqm
    raise ValueError(
ValueError: constrained quadratic model must have 5000 or fewer variables; given model has 8000. Contac
 D-Wave at sales@dwavesys.com if your application requires scale or performance that exceeds the curren
ly advertised capabilities of this hybrid solver.
Leap IDE /workspace/DWAVEQuantum $
```

Solving some literature instances

- You can download the sets of instances here:
  - http://vrp.galgos.inf.puc-rio.br/index.php/en/


Set E (Christofides and Eilon, 1969)

E-n13-k4	12	4	6000	247	yes
E-n22-k4	21	4	6000	375	yes
E-n23-k3	22	3	4500	569	yes
E-n30-k3	29	3	4500	534	yes
E-n31-k7	30	7	140	379	yes

- You can check the results from the literature here:
  - http://vrp.galgos.inf.puc-rio.br/index.php/en/plotted-instances?data=E-n30-k3

into the code...

- Comment the end of your code...

```
"""
#                                       self generation
numberOfCars        = 3
numberOfCities      = 20
capaConsumptionMin  = 1
capaConsumptionMax  = 4
selfgeneration(numberOfCars, numberOfCities, capaConsumptionMin, capaConsumptionMax)
"""
```

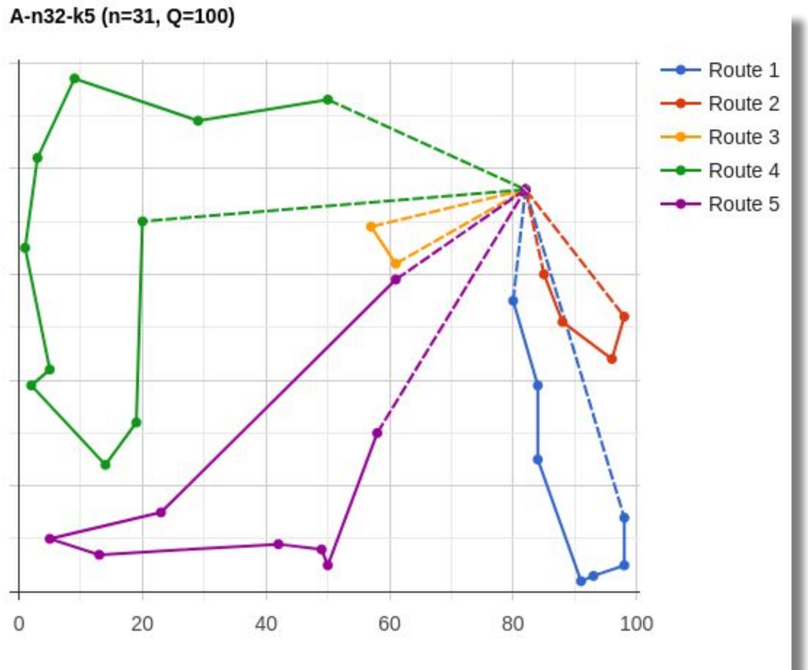
- Write/Uncomment one of the following instructions:

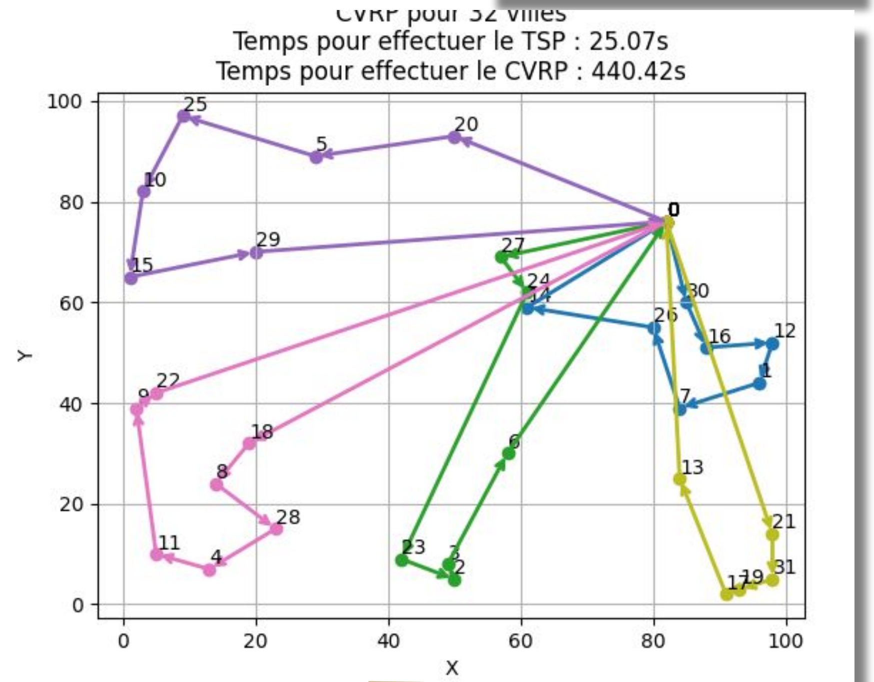
```
# Set A (Augerat, 1995)
literatureGeneration("A-n32-k5")
```

```
# Set E (Christofides and Eilon, 1969)
literatureGeneration("E-n30-k3")
```

Set A (Augerat, 1995)

A-n32-k5.sol
A-n32-k5.vrp

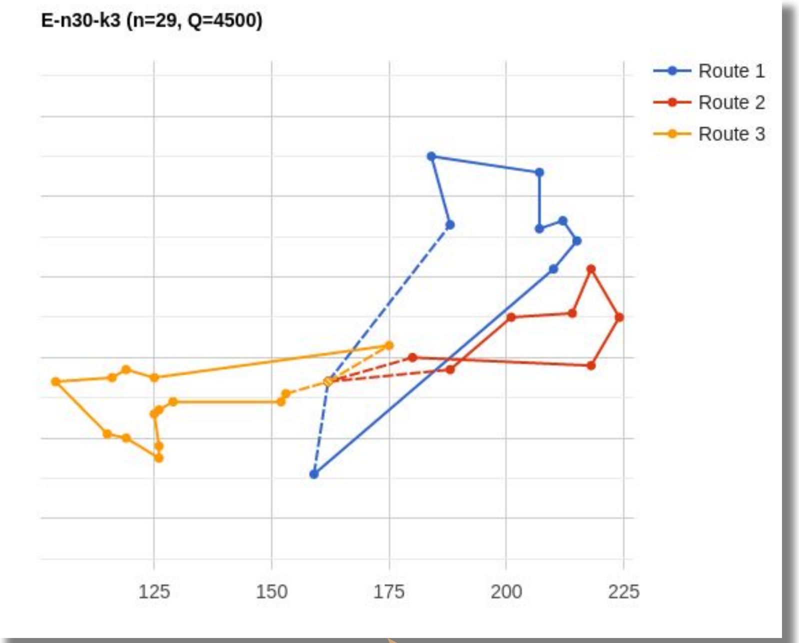


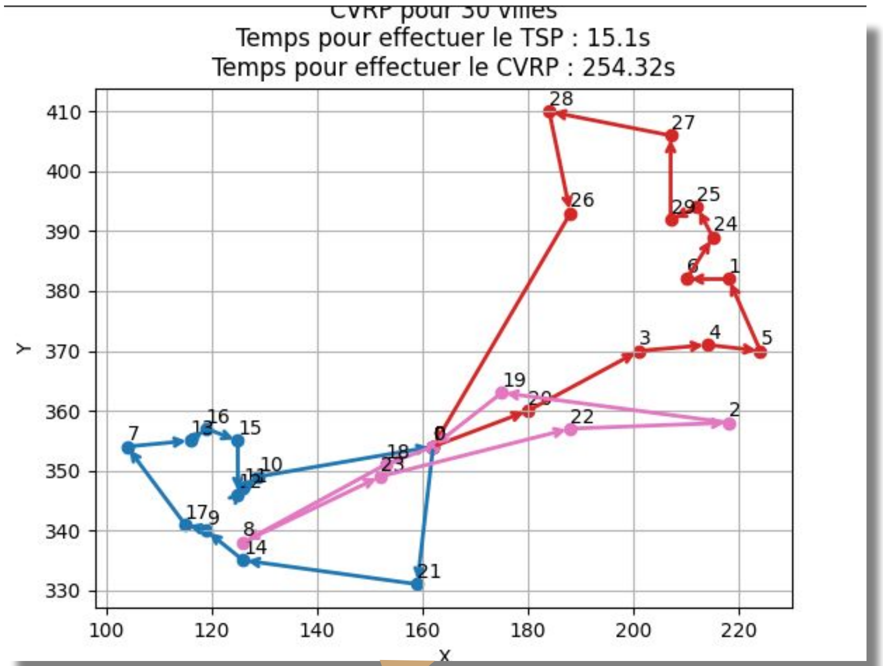



```
Clustering Done
Quantum Resolution: 886.098576123629
Optimal Resolution: 787.8082774366645
Leap IDE /workspace/DWAVEQuantum $
```

Set E (Christofides and Eilon, 1969)

E-n30-k3.sol
E-n30-k3.vrp

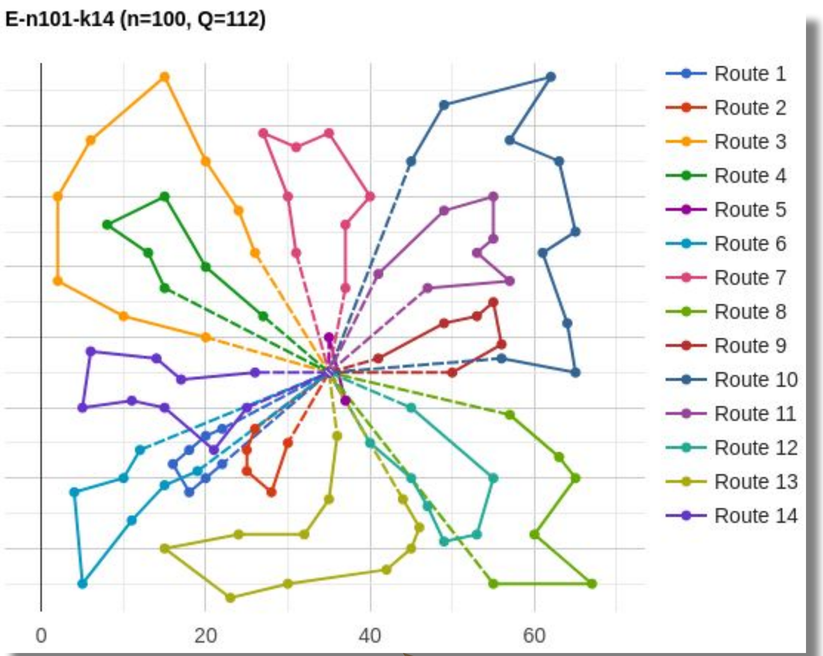


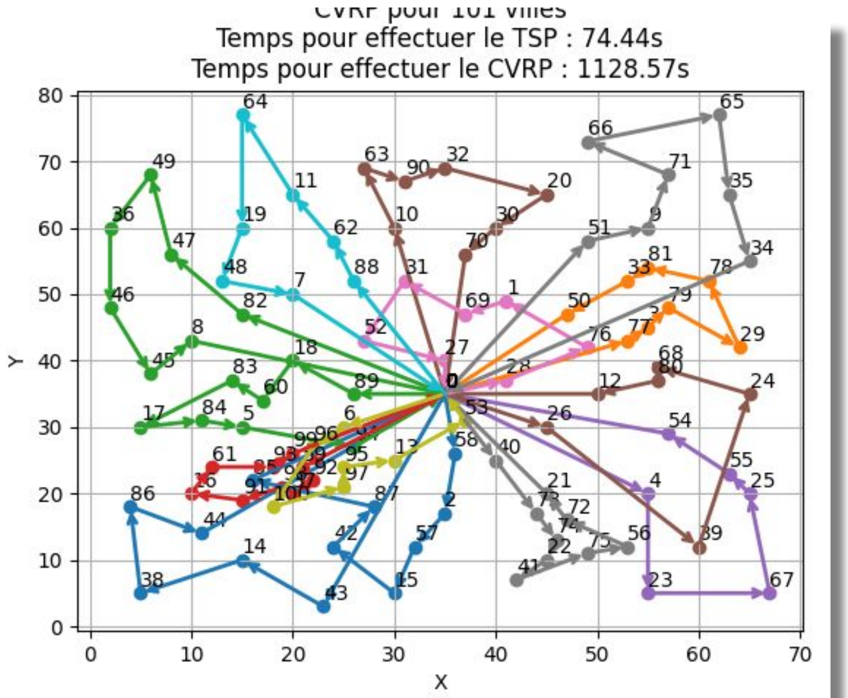



```
Quantum Resolution: 557.4777622059781
Optimal Resolution: 538.9580599403185
Leap IDE /workspace/DWAVEQuantum $
```

Set E (Christofides and Eilon, 1969)

E-n101-k14.sol
E-n101-k14.vrp

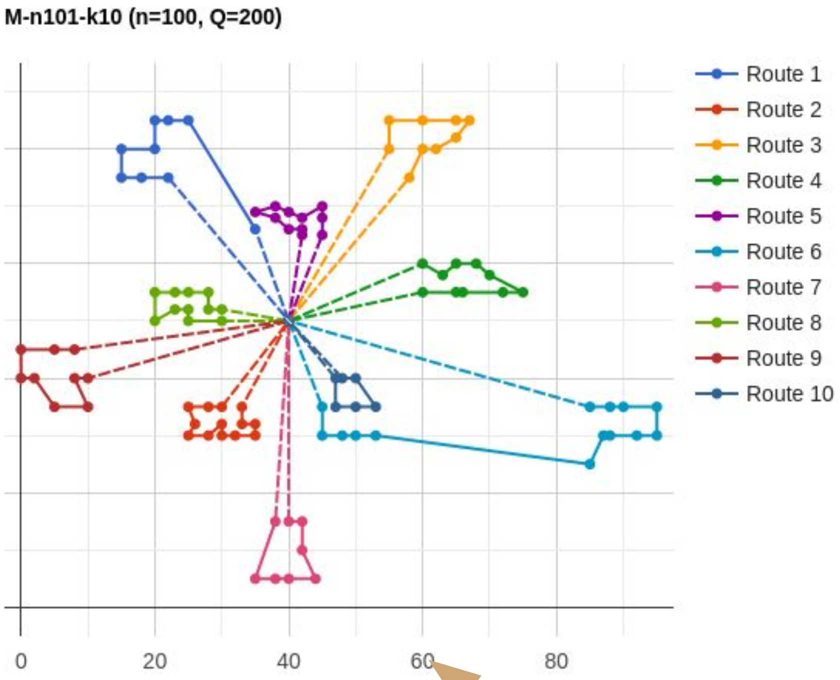


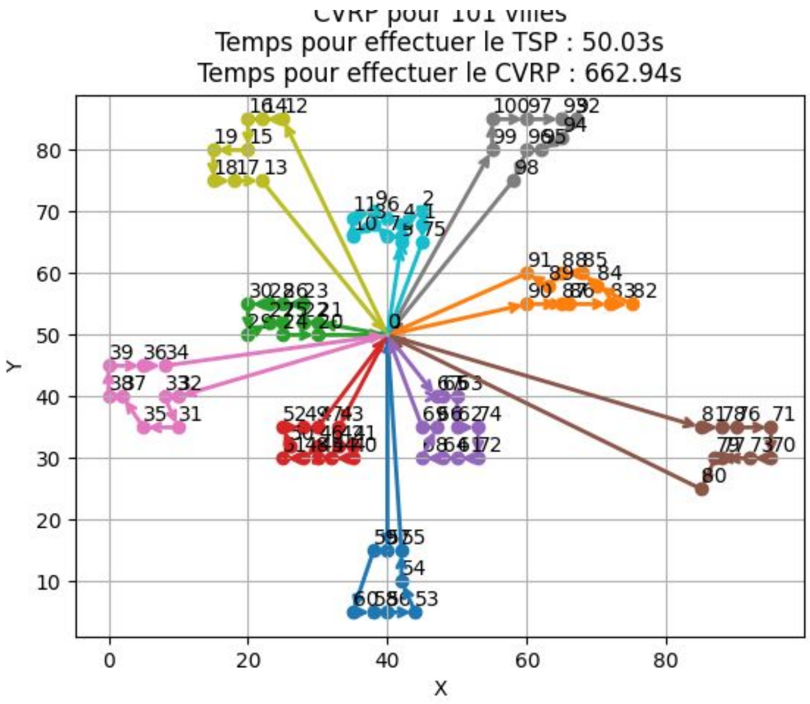



```
Quantum Resolution: 1215.7843157365835
Optimal Resolution: 1082.6501622071194
```

Set M (Christofides, Mingozzi and Toth, 1979)

M-n101-k10.sol
M-n101-k10.vrp

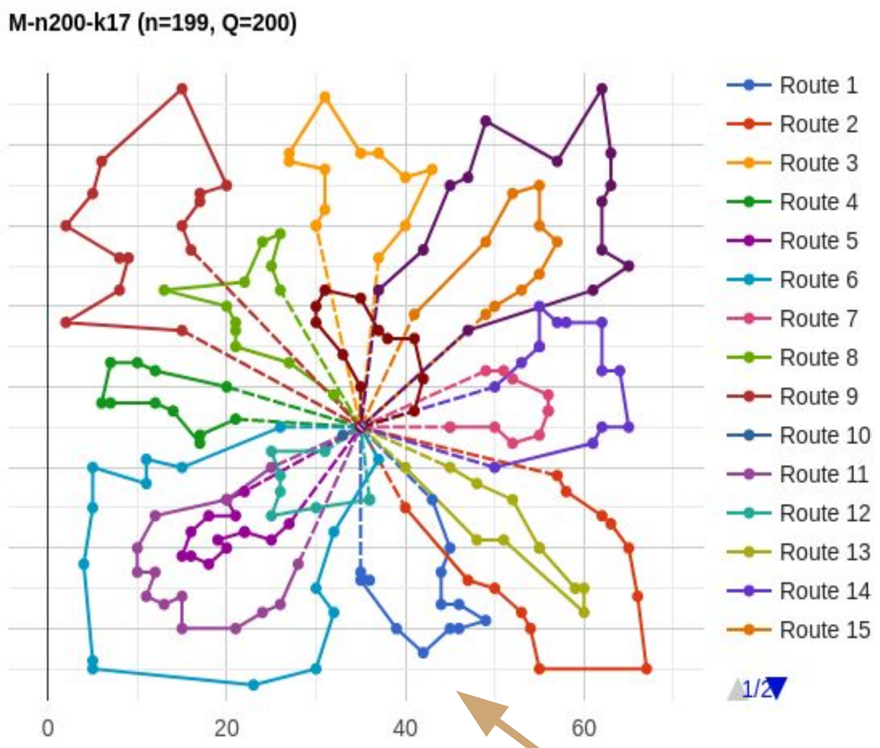


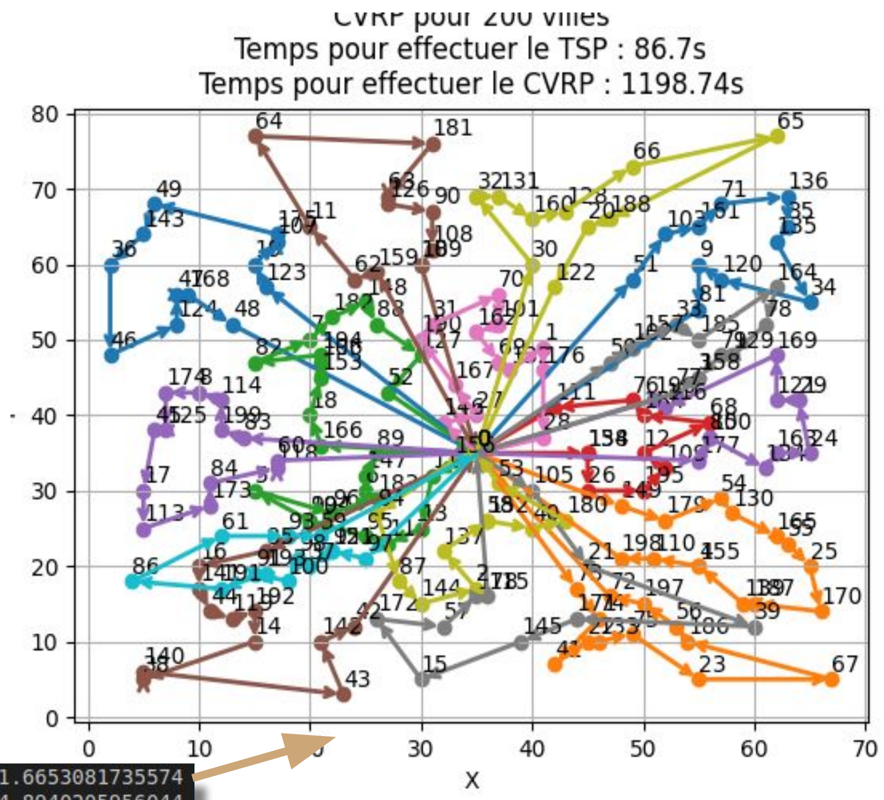



```
Quantum Resolution: 828.9368669428341
Optimal Resolution: 819.8108473050879
```

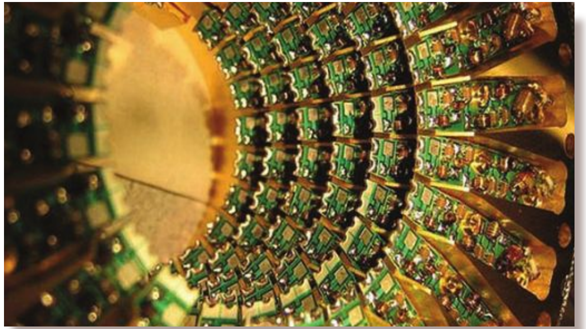
Set M (Christofides, Mingozzi and Toth, 1979)

M-n200-k17.sol
M-n200-k17.vrp






```
Quantum Resolution: 1501.6653081735574
Optimal Resolution: 1294.8940205956044
```



Outline

Summary,
General Results, and
Conclusions

Not a relaxing... relaxation

$$\min \quad \sum_{r\in R} z_{n_r} \tag{1}$$

s.t.

$$x_e^{end} = x_{e+1}^{begin}, \quad e \in RE_r, r \in R : e = 1..n_r - 1, \tag{2}$$

$$x_e^{end} \geq x_e^{begin} + d_e, \quad e \in E, \tag{3}$$

$$x_e^{begin} \geq \tau_e^{begin}, \quad e \in E : i_e = 1, \tag{4}$$

$$x_e^{end} - \tau_e^{end} \leq z_e, \quad e \in E, \tag{5}$$

$$\sum_{t\in T_s} q_{et} = 1, \quad e \in SE_s, s \in S, \tag{6}$$

$$q_{et} + q_{\hat{e}t} - 1 \leq \lambda_{e\hat{e}} + \gamma_{e\hat{e}}, \quad e, \hat{e} \in SE_s, t \in T_s, s \in S : e < \hat{e}, \tag{7}$$

$$x_{\hat{e}}^{begin} - x_e^{end} \geq \Delta_s^M \gamma_{e\hat{e}} - M(1 - \gamma_{e\hat{e}}), e < \hat{e} \text{ in } SE_s, s \in S, o_e \neq o_{\hat{e}}, \tag{8}$$

$$x_{\hat{e}}^{begin} - x_e^{end} \geq \Delta_s^F \gamma_{e\hat{e}} - M(1 - \gamma_{e\hat{e}}), e < \hat{e} \text{ in } SE_s, s \in S, o_e = o_{\hat{e}}, \tag{9}$$

$$x_e^{begin} - x_{\hat{e}}^{end} \geq \Delta_s^M \lambda_{e\hat{e}} - M(1 - \lambda_{e\hat{e}}), e < \hat{e} \text{ in } SE_s, s \in S, o_e \neq o_{\hat{e}}, \tag{10}$$

$$x_e^{begin} - x_{\hat{e}}^{end} \geq \Delta_s^F \lambda_{e\hat{e}} - M(1 - \lambda_{e\hat{e}}), e < \hat{e} \text{ in } SE_s, s \in S, o_e = o_{\hat{e}}, \tag{11}$$

$$\lambda_{e\hat{e}} + \gamma_{e\hat{e}} \leq 1, \quad e, \hat{e} \in SE_s, s \in S : e < \hat{e}, \tag{12}$$

$$x_e^{begin} \geq w_t^{end} q_{et} - M\alpha_e^{w_t}, \quad e \in SE_s, t \in T_s, s \in S, \tag{13}$$

$$x_e^{end} \leq w_t^{begin} q_{et} + M(1 - \alpha_e^{w_t}), \quad e \in SE_s, t \in T_s, s \in S, \tag{14}$$

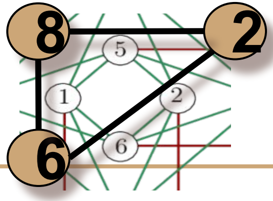
$$x_e^{begin}, x_e^{end}, z_e \geq 0, \quad e \in E, \tag{15}$$

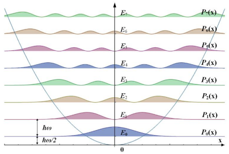
$$\gamma_{e\hat{e}}, \lambda_{e\hat{e}}, \in \{0,1\}, \quad e, \hat{e} \in SE_s, s \in S : e < \hat{e}, \tag{16}$$

$$q_{et}, \alpha_e^{w_t} \in \{0,1\}, \quad e \in SE_s, t \in T_s, s \in S, \tag{17}$$

Conclusions, Current and Future Works

- Hardware Heuristic
  - not properly fulfils Adiabatic Theorem
- QUBO & ISING
  - Not a relaxing relaxation...
  - What does the machine actually do for the relaxation process?
- Promising technology
  - Advantage for tailored problems
  - An Anneal in microseconds,
  - Different solutions?
- Hardware Problem:
  - Low connectivity ;

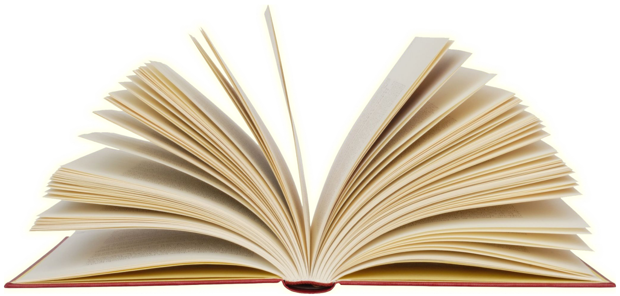




- Work on Combinatorial Optimization Problems such as:
  - Max Cut (more)
  - Max independent set problem
  - Max Clique problem
- Compare future results with quadratic problems solved on classical machines

- Daniel Vert. Étude des performances des machines à recuit quantique pour la résolution de problèmes combinatoires. Thèse 2021.
- Vert, D., Sirdey, R., & Louise, S. (2021). Benchmarking Quantum Annealing Against “Hard” Instances of the Bipartite Matching Problem. *SN Computer Science*, *2*(2), 1-12.

Outline



References & Co

- Some Help to install D-Wave Ocean API
- Several types of references:
  - Books, Vidéos, Conf', Papers, Courses,
  - From vulgarisation to theoretical.

References of this presentation (order of appearance)

Shor, P. W. (1994, November). Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science.* Ieee. (pp. 124-134).
or: Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, *41*(2), 303-332.

Grover, L. K. (1996, July). A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (pp. 212-219).

Farhi, E., Goldstone, J., & Gutmann, S. (2014). A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*.

Kadowaki, T., & Nishimori, H. (1998). Quantum annealing in the transverse Ising model. *Physical Review E*, *58*(5), 5355.

Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671-680.

Daniel Vert. Étude des performances des machines à recuit quantique pour la résolution de problèmes combinatoires. Thèse.

Vert, D., Sirdey, R., & Louise, S. (2021). Benchmarking Quantum Annealing Against “Hard” Instances of the Bipartite Matching Problem. *SN Computer Science*, *2*(2), 1-12.

Festa, P., P. M. Pardalos, M. G. C. Resende, C. C. Ribeiro. 2002. Randomized heuristics for the max-cut problem. Optim. Methods Software 7 1033–1058.

De Santis, M., Festa, P., Liuzzi, G., Lucidi, S., & Rinaldi, F. (2016). A nonmonotone grasp. Mathematical Programming Computation, 8(3), 271-309

Nguyen, V.H., Minoux, M. Linear size MIP formulation of Max-Cut: new properties, links with cycle inequalities and computational results. Optim Lett 15, 1041–1060 (2021). https://doi.org/10.1007/s11590-020-01667-z.

David Garvin, Nec Australia at Qubits conf. 2021-10: Solving the Last Mile Resupply Problem | D-Wave Qubits 2021 https://www.youtube.com/watch?v=BM23FpwlAYk&t=1s.

Christofides, N., & Eilon, S. (1969). An algorithm for the vehicle-dispatching problem. *Journal of the Operational Research Society*, *20*(3), 309-318.

Christofides, N., Mingozzi, A., & Toth, P. (1979). Loading problems. *N. Christofides and al., editors, Combinatorial Optimization*, 339-369.

Other Research Papers

- Smelyanskiy, Vadim N., et al. "A near-term quantum computing approach for hard computational problems in space exploration." arXiv preprint arXiv:1204.2821 (2012).
- Hamerly, R., Inagaki, T., McMahon, P.L., Venturelli, D., Marandi, A., Onodera, T., Ng, E., Langrock, C., Inaba, K., Honjo, T. and Enbutsu, K., 2019. Experimental investigation of performance differences between coherent Ising machines and a quantum annealer. *Science advances*, *5*(5), p.eaau0823.
  - Here the MaxCut problem is solved on DW2Q has 2048 qubits (lets try with a 5K!)
- Barahona, Francisco; Grötschel, Martin; Jünger, Michael; Reinelt, Gerhard (1988). "An Application of Combinatorial Optimization to Statistical Physics and Circuit Layout Design". *Operations Research*. **36** (3): 493–513.
- Barahona. "On the Computational Complexity of Ising Spin Glass Models." J. Phys. A 15 (1982), pp. 3241–3253.
  - About the NP-Hard complexity of Ising and QUBO Problems
- Mengoni, Riccardo, Daniele Ottaviani, and Paolino Iorio. "Breaking RSA Security With A Low Noise D-Wave 2000Q Quantum Annealer: Computational Times, Limitations And Prospects." *arXiv preprint arXiv:2005.02268* (2020).
- King, James, et al. "Benchmarking a quantum annealing processor with the time-to-target metric." arXiv preprint arXiv:1508.05087 (2015).
- Analytical and numerical evidence suggests that quantum annealing outperforms simulated annealing under certain conditions. Heim, B., Rønnow, T. F., Isakov, S. V., & Troyer, M. (2015). Quantum versus classical annealing of Ising spin glasses. Science, 348(6231), 215-217.
- Albash, Tameem, Victor Martin-Mayor, and Itay Hen. "Temperature scaling law for quantum annealing optimizers." Physical review letters 119.11 (2017): 110502.

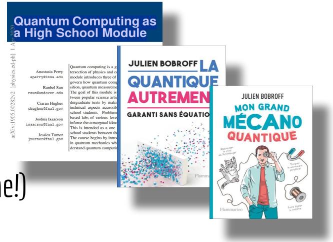
Literature: Some Results on a 1K and 2K qubits Machine

- D-Wave 2X again (1K qubits) runs up to **100M times faste**r than an optimized implementation of the quantum Monte Carlo algorithm **on a single core**.
    - Denchev, V.S., Boixo, S., Isakov, S.V., Ding, N., Babbush, R., Smelyanskiy, V., Martinis, J. and Neven, H., 2016. What is the computational value of finite-range tunneling?. Physical Review X, 6(3), p.031015.
- D-Wave newsletter of the **18/02/2021** about results on 2K machine:
    - "Today D-Wave marks a major milestone on the journey to quantum advantage in a new peer-reviewed paper published in *Nature Communications*. The new research uses a D-Wave lower noise system to demonstrate **3 million times speed-up over classical alternatives in a real-world problem**. [...]"
    - The Paper: King, A.D., Raymond, J., Lanting, T. et al. Scaling advantage over path-integral Monte Carlo in quantum **simulation of geometrically frustrated magnets**. Nature Communication 12, 1113 (2021).
    - **They compared the QA against the best known classical simulation algorithm for this problem (PIMC).**
    - To know which CPU is used in comparison (also the number of threads used), we need to go in the supplementary materials (they use a light **i7-8650U on one core**! To compare, if it is still possible, and if the algo is fully parallelizable, the Fujitsu A64FX will win since it has more than 7 millions of (ARM) cores)!
- Be careful with all the announcements in the media...and also from the companies!
- But still... waiting for results on **Advantage**(5k qubits and 15 connectivity)

General popularization works

(from the easiest to the hardest)

- Article (100p): Quantum Computing as a High School Module https://arxiv.org/abs/1905.00282
- Book (fr): La quantique autrement. Julien Bobroff.
- Book (fr): Mon grand mécano quantique. Julien Bobroff.



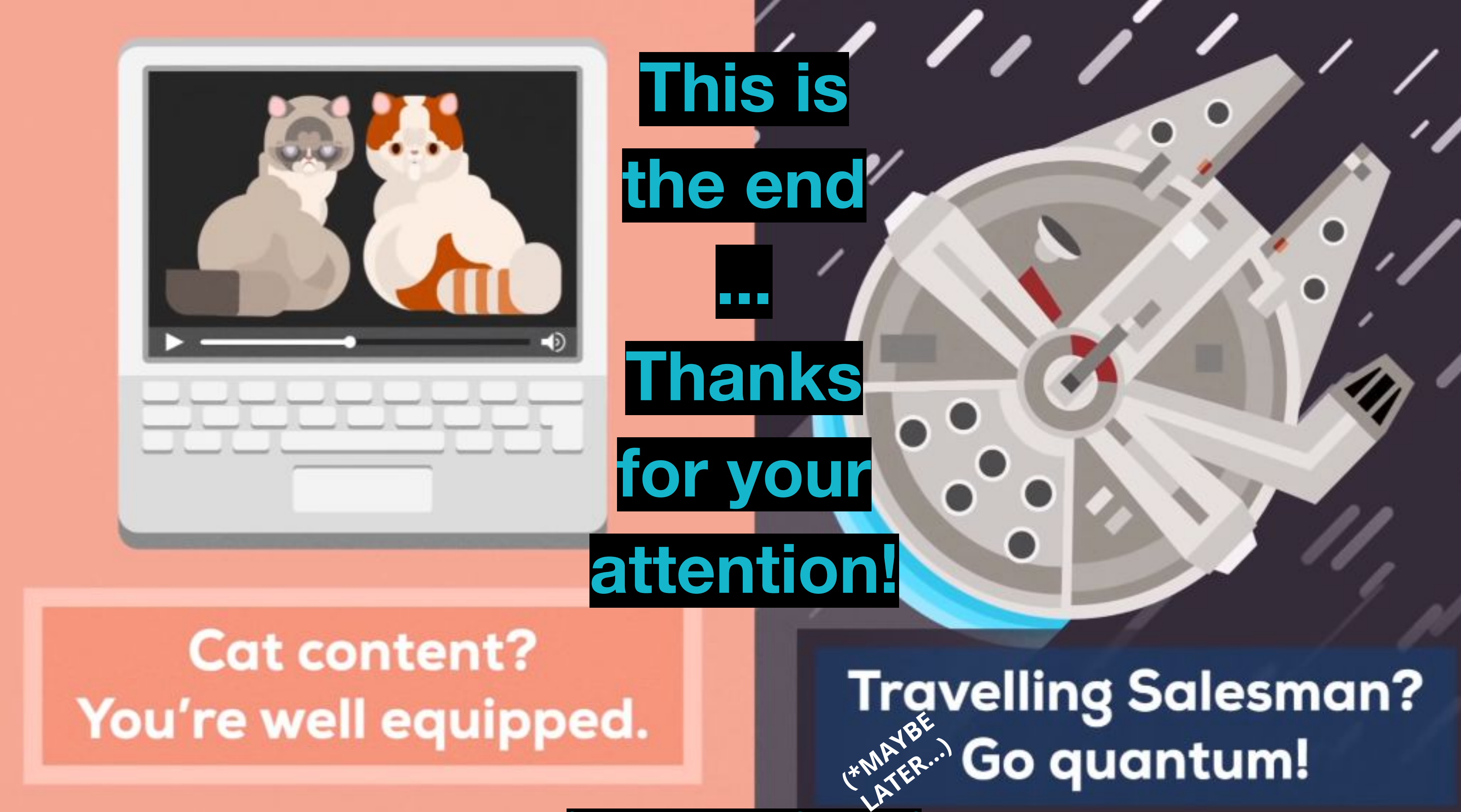
- Video: "Les Ordinateurs Quantiques Expliqués - Limites de la technologie humaine" (so well done!) https://www.youtube.com/watch?v=JhHMJCUmq28&ab_channel=Kurzgesagt%E2%80%93InaNutshell
- Website article: "You don't need to be a mathematician to master quantum computing" https://towardsdatascience.com/you-dont-need-to-be-a-mathematician-to-master-quantum-computing-161026af8878
- Slides: US Department of energy: How about quantum computing. Bert de Jong https://cs.lbl.gov/assets/CSSSP-Slides/20190624-deJong.pdf
- Videos: Understanding Quantum Mechanics. Sabine Hossenfelder. https://www.youtube.com/watch?v=XJSfgE9LUJw&list=PLwgQsqtH9H5djIfFhXE6We207beTgUnyL
- Book: Quantum computation and quantum information. Nielsen & Chuang

Courses

- Frederic Magniez, “Algorithmes Quantiques”, Collège de France, Chaire Informatique et sciences numériques 2020-2021. https://www.college-de-france.fr/site/frederic-magniez/p5045022425906061_content.htm
- “A practical introduction to quantum computing: from qubits to quantum machine learning and beyond (5/7)”
  - 7 courses from the CERN Quantum Technology Initiative.
  - (5/7): “Quantum algorithms for combinatorial optimization. Quantum adiabatic computing and quantum annealing. Introduction to D-Wave Leap. Quantum Approximate Optimization Algorithm.”
  - https://cds.cern.ch/record/2746545
- John Preskill lectures:
  - http://theory.caltech.edu/~preskill/ph229/
- Serge Haroche (collège de France lectures)
  - https://www.college-de-france.fr/site/serge-haroche/_course.htm

D-Wave ressource

- Leap:
    - https://cloud.dwavesys.com/leap/
- Qubo & Ising Models
    - https://docs.dwavesys.com/docs/latest/c_gs_3.html
- D-Wave examples on Github
    - https://github.com/dwave-examples
- Tutos (Videos and articles) for developers
    - https://www.dwavesys.com/practical-quantum-computing-developers
- Tuto for installing the Ocean API:
    - https://docs.ocean.dwavesys.com/en/stable/overview/install.html
- A tons of videos from D-Wave:
    - https://www.youtube.com/user/dwavesystems/playlists



This is the end ... Thanks for your attention!

(samuel.deleplanque@junia.com)

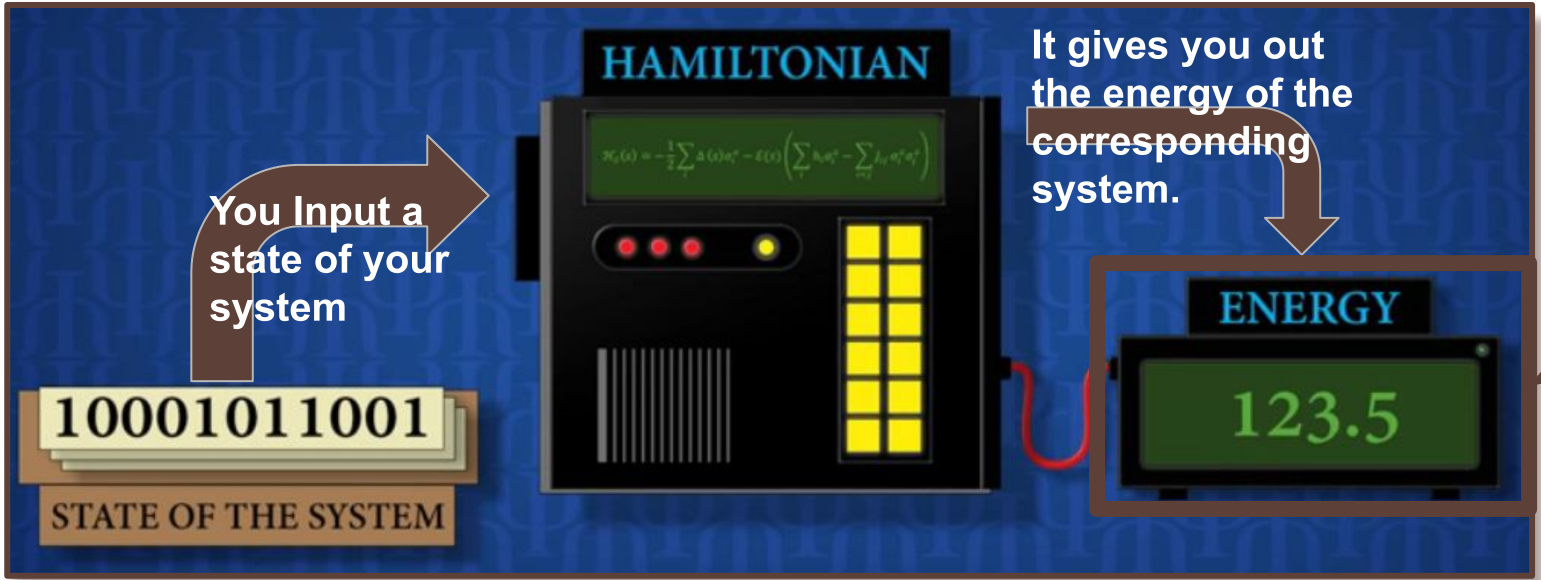
pic: https://www.youtube.com/watch?v=JhHMJCUmq28&ab_channel=Kurzgesagt%E2%80%93InaNutshell

Rejected slides...

The Hamiltonian during the quantum annealing....

Quantum Annealing

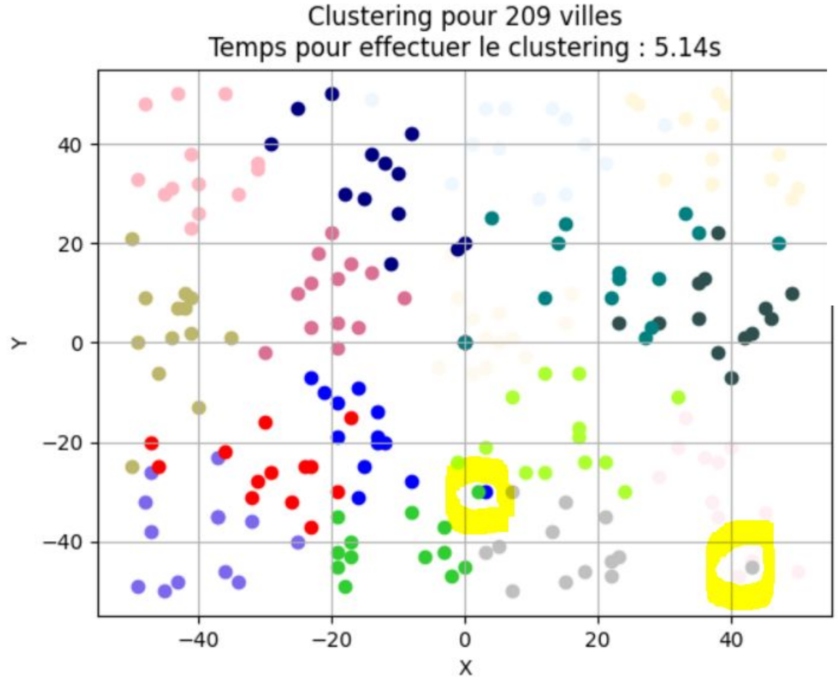
- The **Hamiltonian ($\mathcal{H}$)** is (here) a mathematical description of the energy of a physical system. (here we focus on the energy inside the QPU).

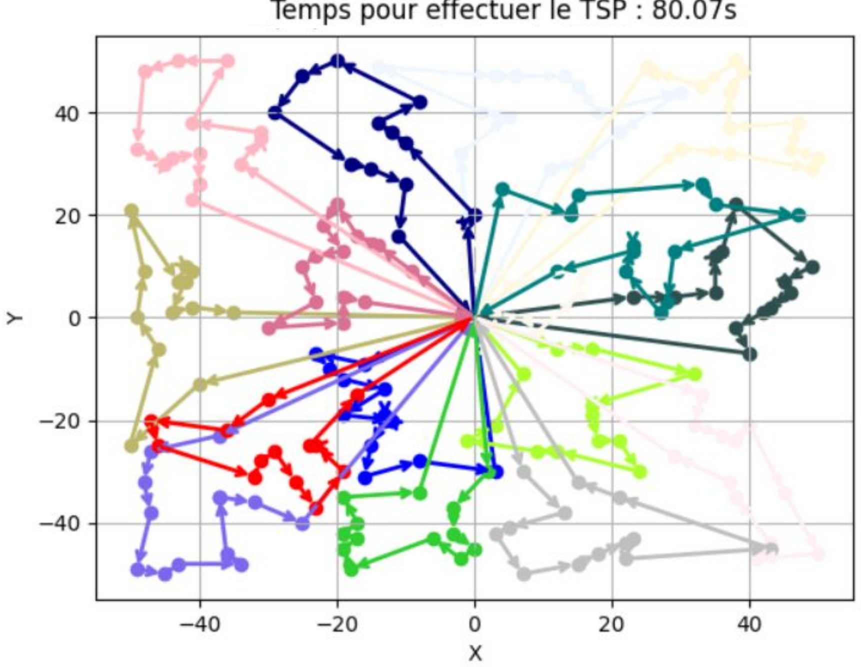


Goal? To Find the **minimum** energy state

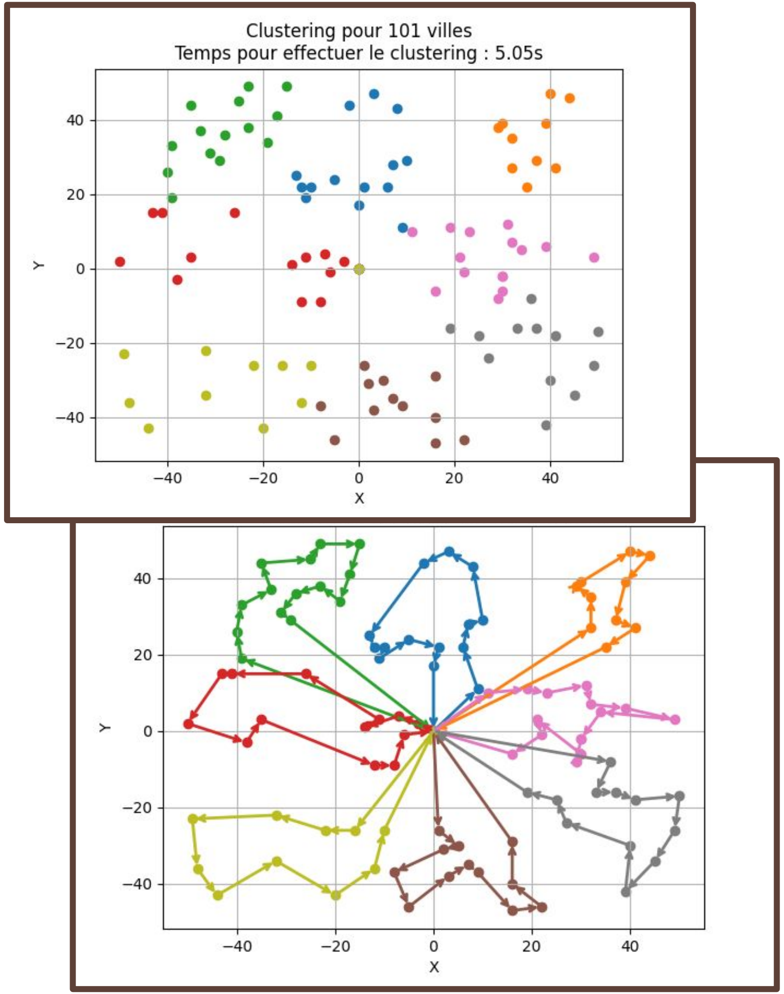
pics: https://www.youtube.com/watch?v=tnikftltqE0&t=125s

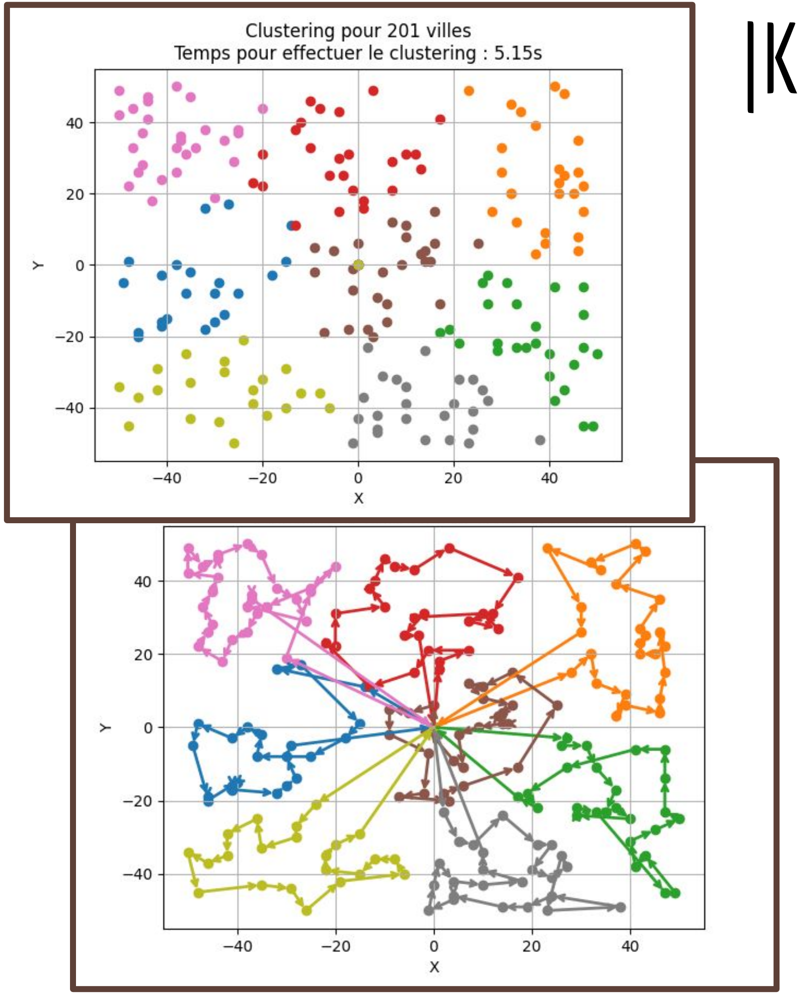
|K| =16 ; n = 208


Clustering pour 209 villes
Temps pour effectuer le clustering : 5.14s
Y
X

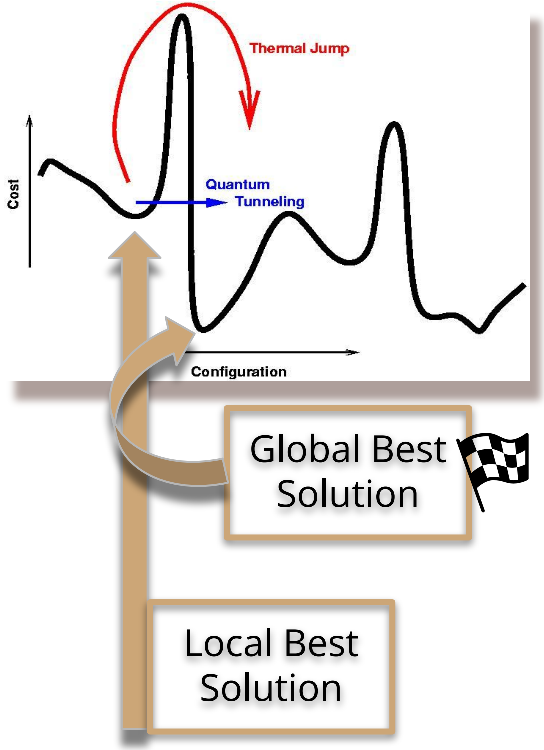

Temps pour effectuer le TSP : 80.07s
Y
X

|K| =8


Clustering pour 101 villes
Temps pour effectuer le clustering : 5.05s
X
Y


Clustering pour 201 villes
Temps pour effectuer le clustering : 5.15s
X
Y

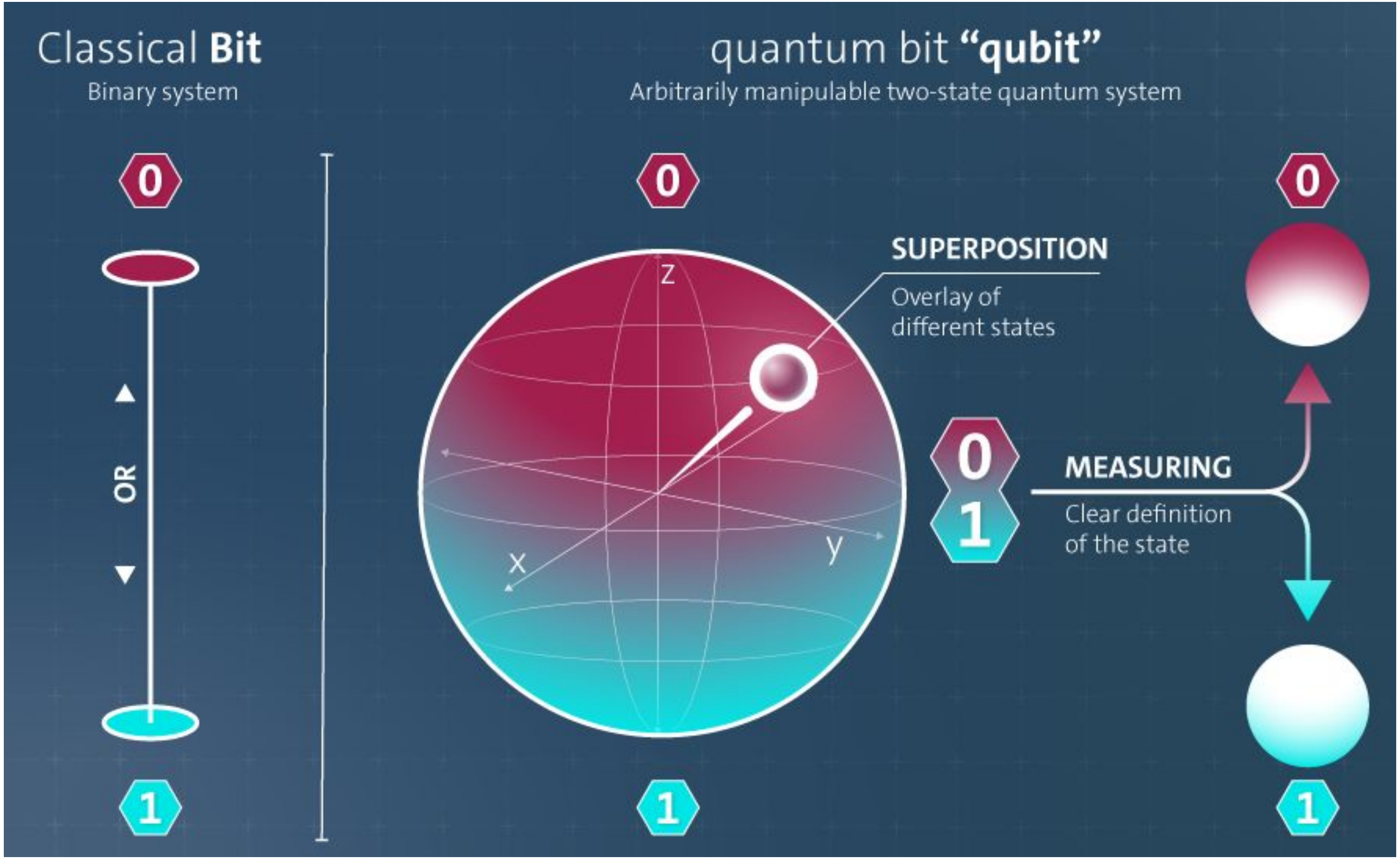
Simulated Annealing & Quantum annealing: 2 Metaheuristics



Analogy

While the temperature is directly related to the probability of moving to a new solution in simulated annealing, in quantum annealing, the strength of transverse field determines the quantum-mechanical probability to **change the amplitudes of all states in parallel**.

The advantage of QA compared to SA is due to the quantum mechanics allowing for **an additional escape route from local minima**. While SA must climb over energy barriers to escape traps, **QA can penetrate these barriers without any increase in energy**. This effect is known as **quantum tunneling**.

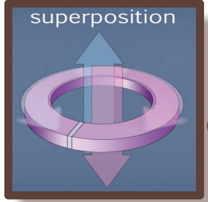

Classical Bit
Binary system
0
OR
1
quantum bit "qubit"
Arbitrarily manipulable two-state quantum system
0
z
x
y
SUPERPOSITION
Overlay of different states
0
1
MEASURING
Clear definition of the state
0
1
1

Future of the quantum annealing tech

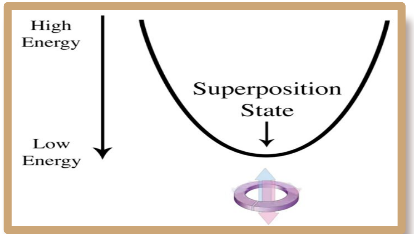
- Several Projects in Europe:
  - **AVaQus**
    - the European project developing the first superconducting quantum annealer
    - https://www.quantaneo.com/AVaQus-the-European-project-to-develop-the-first-superconducting-coherent-quantum-annealer_a479.html
  - **ATOS**
    - Quantum Annealing **Simulator**
    - Scope: Machine Learning
    - https://atos.net/en/2020/press-release_2020_07_07/atos-opens-up-a-new-path-to-quantum-annealing-simulation
  - **Qilimanjaro**

First Step

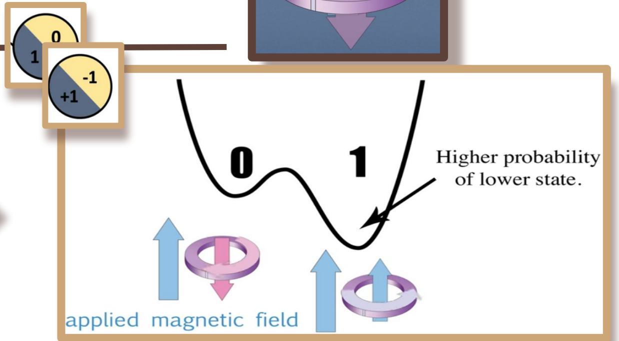
The Qubits start in a superposition state (**zero and one**).

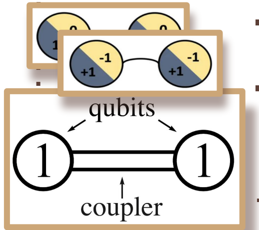


and

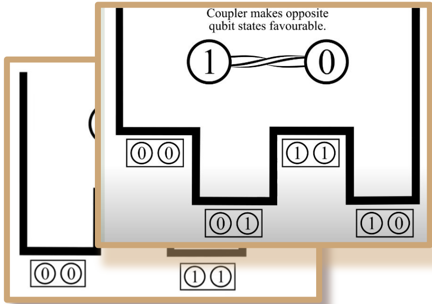


Control the probability of the state with a programmable external magnetic field



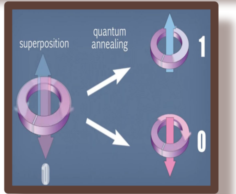


- **link the qubits together** by **couplers** (entanglement)
- The couplers define how a pair of qubits **ends up**, in the same or in opposite state, resp.: 0 & 0 ; 1 & 1 or, 1 & 0 ; 0 & 1.
- **the machine will "physically" make the equality (or the inequality)** "**energetically favorable**", i.e., to "low the energy" of those states.

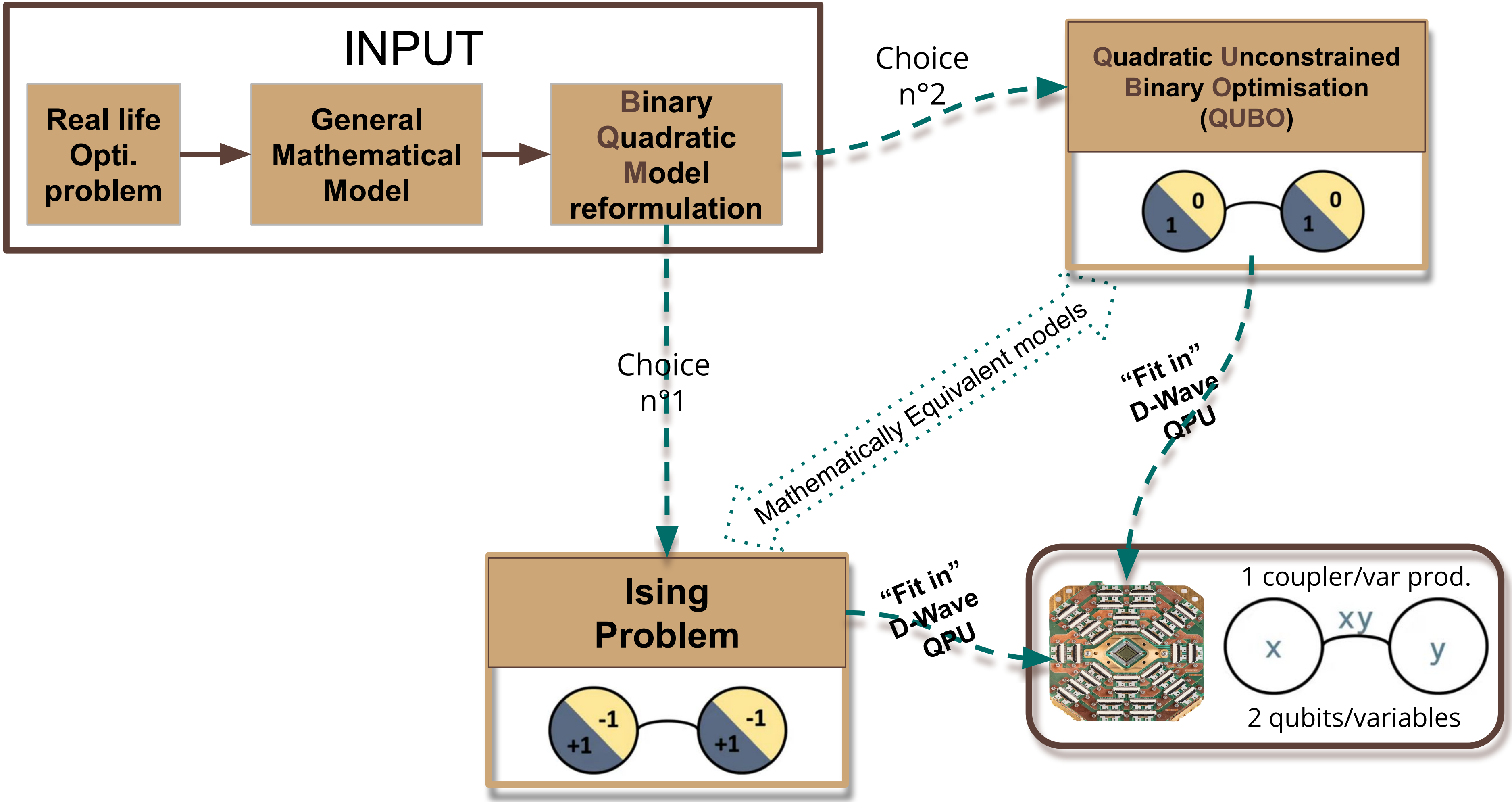


Last step

At the end, the measure of each qubit make them going to (with a high probability) either the **zero state or** the **one state**.

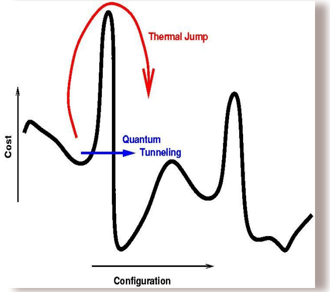


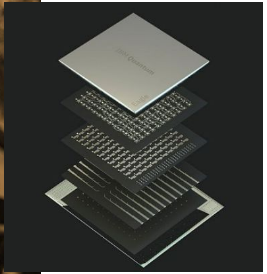
or

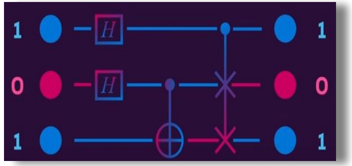

INPUT
Real life Opti. problem
General Mathematical Model
Binary Quadratic Model reformulation
Choice n°2
Quadratic Unconstrained Binary Optimisation (QUBO)
0
1
0
1
Choice n°1
Mathematically Equivalent models
"Fit in" D-Wave QPU
Ising Problem
-1
+1
-1
+1
"Fit in" D-Wave QPU
1 coupler/var prod.
x
xy
y
2 qubits/variables

5k qubits… versus 127 ???

Differences between a D-Wave machine and a gates quantum computer?







There is no match…they are not in the same competition

End of 2020, the **Advantage** D-Wave machine is released…

- A new **Q**PU with:
  - more than **5,000 qubits**
  - **35,000 couplers**
    - To link pair of qubits
- qubits technology:
  - Superconducting qubits (*SQUID*)
  - Metal used: **niobium**
    - We obtain quantum mechanical effects when the **niobium** becomes a superconductor (once it is cooled down to ≈0K).