Constraint Solving by Quantum Annealing

Philippe Codognet
CNRS / Sorbonne U. / U. of Tokyo
Japanese-French Laboratory for Informatics
JFLI - IRL 3527, Tokyo, JAPAN

Main Idea

- Two main paradigms in Quantum Computing :
  - Gate-model
  - Quantum annealing
- Investigate the use of Quantum Annealing (QA) for Constraint Satisfaction Problems
- QUBO is the input format for QA Systems
  - Quadratic Unconstrained Binary Optimization
- Implementation on quantum computer
  - D-Wave Systems Advantage (5000 qubits)

Modeling Combinatorial Optimization and Constraint Satisfaction Problems in QUBO

- “simple” problems :
  - A. Lucas, *Ising Formulations of Many NP Problems*, Frontiers in Physics, vol. 2, pp. 5, Feb 2014
  - Fred Glover, Gary Kochenberger &Yu Du, *A Tutorial on Formulating and Using QUBO Models,* arxiv.org 2019
- More complex problems from OR or CP such as:
  - Quadratic Assignment Problem
  - N-queens
  - Costas Arrays
  - Magic Squares

What is QUBO exactly ?

- QUBO = Quadratic Unconstrained Binary Optimization
- Consider *n* Boolean variables $x_1,\ldots, x_n$

  and a quadratic expression over $x_1,\ldots, x_n$ to minimize:

  $\Sigma_{i=1}^{n} q_i x_i + \Sigma_{i=1}^{n} {q'}_i x_i^2 + \Sigma_{i,j}\, q_{ij} x_i x_j$
- Equivalent to :

$$\Sigma_{i\leq j}\, q_{ij} x_i x_j$$

- QUBO in matrix format: *minimize* y = $x^{t} Q x$

  where x is a vector of binary decision variables

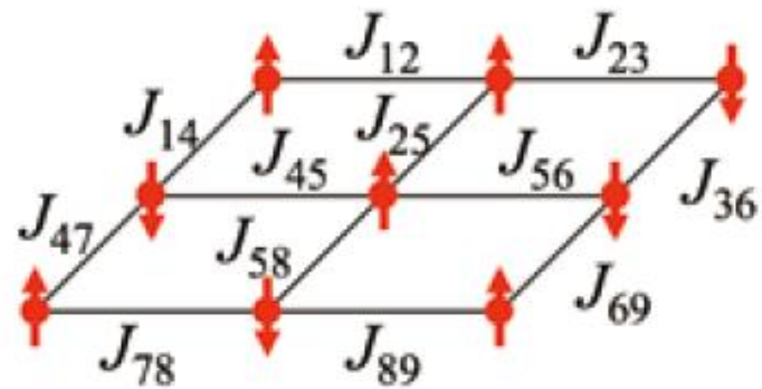
  and Q is a square matrix of constants

The Ising Model

- The **Ising model** is a mathematical model of ferromagnetism in statistical mechanics

- atomic "spins“:
  either +1 or −1

- 2D Lattice :
  Hamiltonian
  (energy function)
  is expressed by ***H***



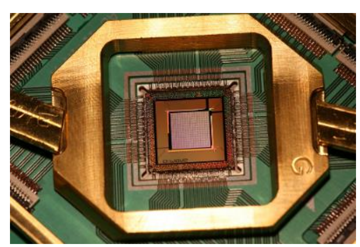
$$H = -\sum_{\langle i,j \rangle} J_{ij}\sigma_i\sigma_j - \sum_j h_j\sigma_j$$

[from Yamaoka 2019]

J_{ij} = coupling strength between i and j, h_j = bias on i

Solving techniques for QUBO

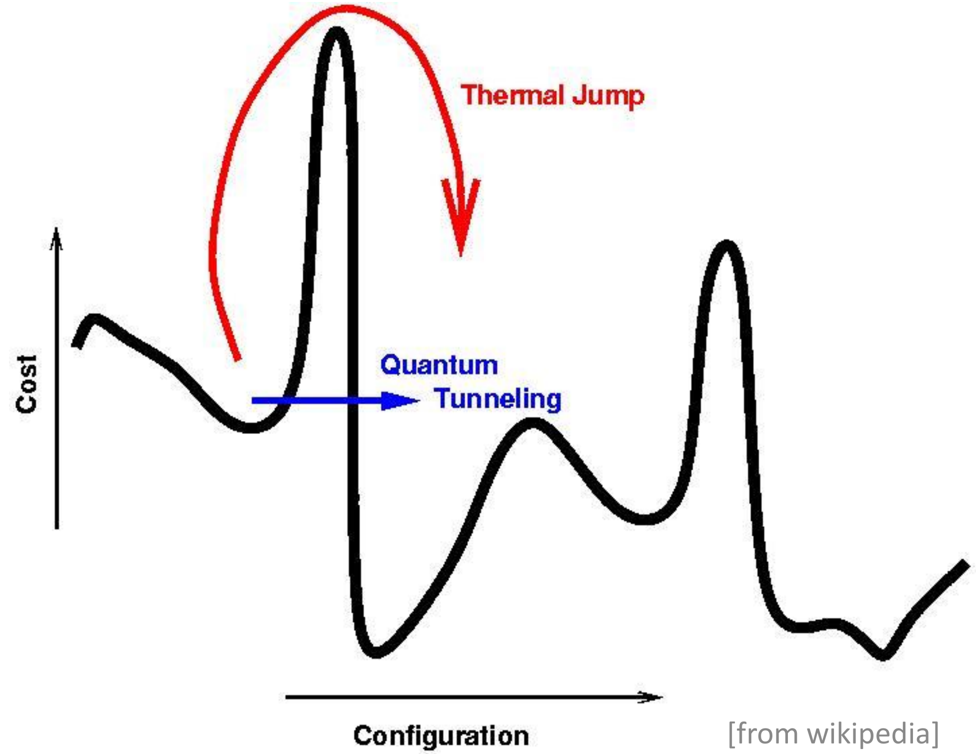
- Software:
  - CPLEX (Branch-and-Cut quadratic integer optimizer)
  - specialized version of Tabu Search
    (e.g. algorithms by F. Glover, J-K. Hao *et al*.)
- Hardware:
  - CMOS Digital Annealing (« quantum-inspired »)
    Fujitsu DAU
  - Quantum Annealing
    D-Wave
  - Others (prototypes)



Quantum Annealing

- Introduced by [Kadowaki & Nishimori 1998]
- Use quantum tunneling effect to escape local minima instead of thermal jump

- Quantum tunneling
  = quantum phenomenon
  where a wavefunction
  can propagate through
  a potential barrier.



[from wikipedia]

QUBO… Unconstrained ?

- In QUBO, the « U » stands for « unconstrained »
- Model only have an objective function to optimize

- Could we also extend that in order to represent some problems with « constraints » ?
- **Yes**: → add constraints as « penalties » in the objective function (to minimize)

- Similar to constraint-based local search…

Constraints as penalties

- Penalties are chosen so that they are equal to zero for feasible solutions (i.e. satisfying the constraints) and some positive value otherwise, e.g.:
  - constraint $x + y \leq 1$ leads to penalty: xy
  - constraint $x = y$ leads to penalty: $x + y - 2xy$
- Penalties are compositional
  - by addition
  - With penalty coefficient (weights) for each constraint

The « one-hot » Constraint

- Often used in Boolean models

  e.g. graph *n*-coloring, each node *i* has one color *j*

  $$\Sigma_{j=1}^{n} x_{ij} = 1$$

- Penalty in objective function in QUBO :

  $$(\Sigma_{j=1}^{n} x_{ij} - 1)^2 \quad = \quad -\Sigma_{j=1}^{n} x_{ij} + 2\, \Sigma_{j<k}\, x_{ij} x_{ik} + 1$$

- Also for « at-most-one » constraint $\Sigma_{i=1}^{n} x_i \leq 1$

  In QUBO : $(\Sigma_{i=1}^{n} x_i - \frac{1}{2})^2$

  Corresponding Penalty: $\Sigma_{i<j} x_i x_j$

The « *all-different* » Constraint

- In Constraint Programming : *all_different([x_1,…x_n])*
  - e.g. for N-queens, Magic Square, Costas Array Problem, …

- In Boolean models we need n^2 variables x_{ij} such that:

$$x_{ij} = 1 \text{ iff } x_i \text{ has value } j$$

- *All_different* can be translated by $2n$ constraints:

$$\forall i \in \{1, n\} \ \Sigma_{j=1}^{n} x_{ij} = 1$$
$$\forall j \in \{1, n\} \ \Sigma_{i=1}^{n} x_{ij} = 1$$

« All-different » in QUBO

- Therefore in QUBO, $2n$ penalties of the form:

$$\forall i \in \{1, n\} \;\; (\Sigma_{j=1}^{k} x_{ij} - 1)^2$$
$$\forall j \in \{1, n\} \;\; (\Sigma_{i=1}^{k} x_{ij} - 1)^2$$

- Adding all expressions together gives the overall penalty:

$$\sum_{k=1}^{n} \sum_{i<j} x_{ik}x_{jk} + \sum_{k=1}^{n} \sum_{i<j} x_{ki}x_{kj} - \sum_{i=1,j=1}^{n} x_{ij}$$

The Quadratic Assignment Problem in QUBO

- **Problem:** n locations $\{1,\ldots,n\}$, n facilities $\{1,\ldots,n\}$,

flow matrix F = (f_{ij}) , distance matrix D = (d_{ij}), variables x_{ik}

x_{ik} = 1 if facility *i* assigned to location k, x_{ik} = 0 otherwise

- **QUBO:**

Objective function
$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij}d_{kl}x_{ik}x_{jl}$$

To ensure that each facility is assign to a different location, we must add two sets of One-Hot constraints

$$\forall i \in \{1,\cdots,n\} \ \ (\sum_{j=1}^{n} x_{ij}-1)^2 \ , \ \forall j \in \{1,\cdots,n\} \ \ (\sum_{i=1}^{n} x_{ij}-1)^2$$

More explicitly, we can use the penalty associated with an `all-different` constraint

$$p(\sum_{k=1}^{n}\sum_{i<j} x_{ki}x_{kj} + \sum_{k=1}^{n}\sum_{i<j} x_{ik}x_{jk} - \sum_{i=1,j=1}^{n} x_{ij})$$

Magic Square (CSPLIB 019)

- Place all the numbers in $\{1, 2, . . . , n^2\}$ on a $n \times n$ square s.t. sum of the numbers in all rows, columns and two diagonals are equal

52	73	7	64	21	15	35	98	99	44
91	58	25	6	66	19	41	79	84	43
31	60	62	11	5	26	29	68	36	74
100	40	2	3	20	61	65	86	24	88
4	38	14	76	87	71	16	80	53	97
34	22	85	89	82	18	77	69	47	56
8	9	57	67	50	78	42	10	96	70
90	1	13	39	46	33	81	49	27	59
83	30	48	12	51	45	55	92	28	23
95	93	63	32	72	17	94	75	37	54

- $2n + 2$ linear constraints:

(n rows, n columns, 2 diagonals)

e.g. column j: $\Sigma_{i=1}^{n} \, x_{ij} \;=\; M$

(with $M = \frac{n}{2}(n^2{+}1)$)

- And one *all_different* constraint

Magic Square in QUBO

- Variables: x_{ij}^{k} = 1 if variable x_{ij} has value k, x_{ij}^{k} = 0 otherwise
- Linear constraint on column *j*: $\sum_{i=1}^{n}(\sum_{k=1}^{n} kx_{ij}^{k}) = \frac{n}{2}(n^2+1)$
- Penalty in QUBO:

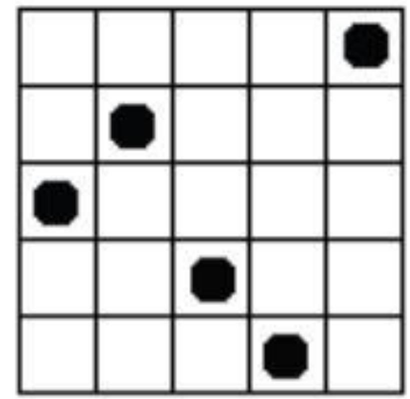
$$\sum_{i} k(k - n(n^2+1))x_{ij}^{k} + 2 \sum_{n*i+k<n*i'+k'} kk' x_{ij}^{k} x_{i'j}^{k'}$$

- All-different constraint:

$$\sum_{i=1,j=1}^{n} \sum_{\{k<k'\}} x_{ij}^{k} x_{ij}^{k'} + \sum_{k=1}^{n} \sum_{\{n*i+j<n*i'+j'\}} x_{ij}^{k} x_{i'j'}^{k} - \sum_{i=1,j=1,k=1}^{n} x_{ij}^{k}$$

- Penalty coefficients should be added…

Costas Array Problem (CAP)



- Proposed by Costas in the 60's
- Sonar / Radio applications
- Active community for 50 years
- Constructive methods exist
  - but not valid for all n
- Very combinatorial, solutions are very scarce
  - n=29 has only 164 solutions (23 unique) out of 29!
- still several open problems …
  - Does costas arrays exist for n=32 ?

Modeling CAP in QUBO (1)

- Variables: x_{ij} = 1 if mark on (i, j) , x_{ij} = 0 otherwise
- Basic constraints:
  - Only one mark per line: $\sum_{j=1}^{N} x_{ij}$ = 1
  - Only one mark per column: $\sum_{i=1}^{N} x_{ij}$ = 1
- Thus penality: $\sum_{i=1}^{N}\left(\sum_{j=1}^{N} x_{ij} - 1\right)^2 + \sum_{j=1}^{N}\left(\sum_{i=1}^{N} x_{ij} - 1\right)^2$

  = $\sum_{k=1}^{n} \sum_{i<j} x_{ik}x_{jk} + \sum_{k=1}^{n} \sum_{i<j} x_{ki}x_{kj} - \sum_{i=1,j=1}^{n} x_{ij}$
- Objective function, ensuring the Costas property if =0 :

$$\sum_{i=1,h=1,k=1,l=1,j=i+1,m=1}^{N} x_{ih}\, x_{i+k,l}\, x_{jm}\, x_{j+k,m+l-h}$$

- Quartic expression… can be quadratized by adding 2 ancillary variables per monomial (basic scheme)
- Thus $n^3(n-1)^3$ new variables !

From HOBO to QUBO

- HOBO = Higher Order Binary Optimization
  i.e. with monomial more than quadratic
- Example: $x_1 x_2 x_3$ is not a quadratic term
- Several « quadratization » scheme exists
  - [Rosenberg 1976] : replace $x_1 x_2$ by new variable y and add penalty function $3y + x_1 x_2 - 2x_1 y - 2x_2 y$
  - [Ishikawa 2011] for positive monomials
  - [Kolmogorov 2014] for negative monomials
  - [Boros, Crama et al. 2017] pairwise covers

- Pairwise covers seems a good compromise in practice…

Modeling CAP in QUBO (2)

- $d_{ij}^{kl} = 1$ iff mark on (i, j) and mark on $(k, l), k > i$
- Thus $n^2 \times n(n-1)/2$ new variables (+ n^2 x_{ij})
- Costas property: no two d_{ij}^{kl} with same differences between indices ($\mathrm{k} - \mathrm{i}$ and $l - j$)
- Objective function to minimize :

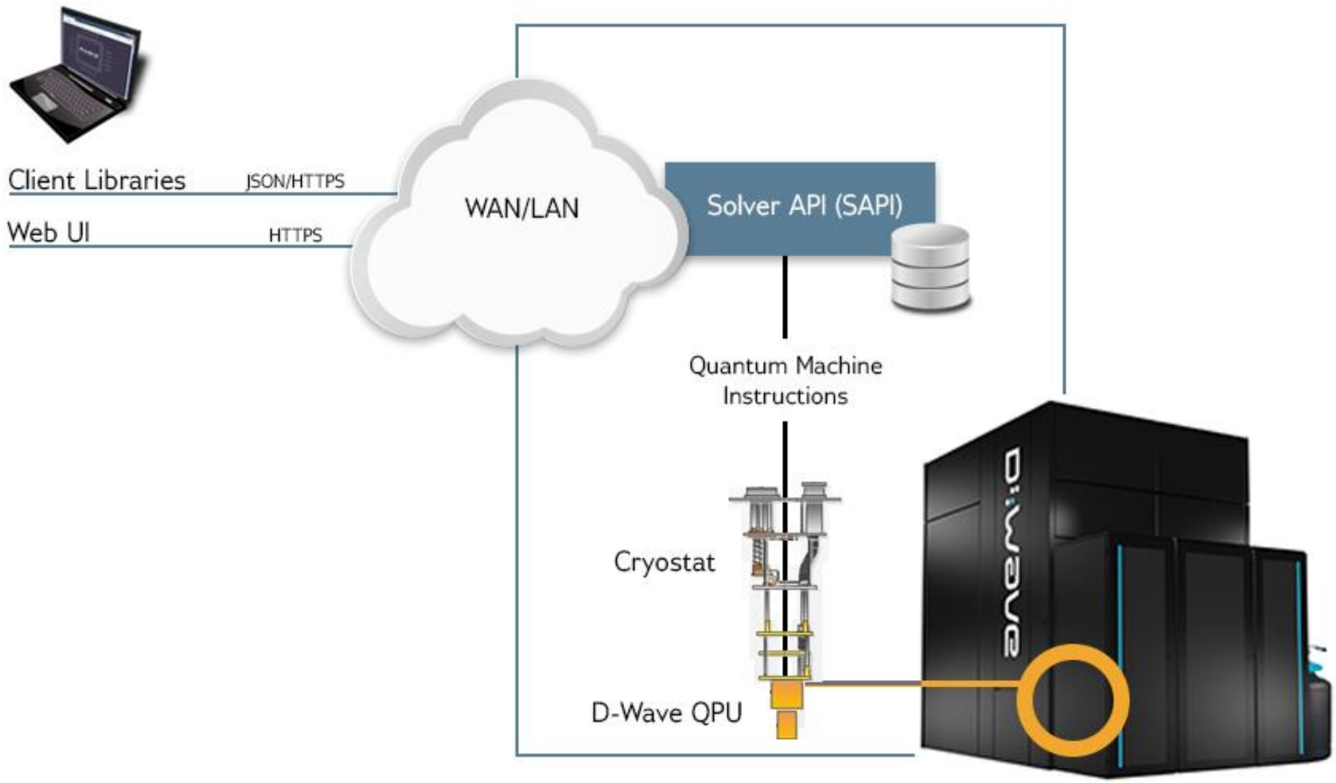
$$\sum_{\dots} d_{ij}^{kl}\, d_{ab}^{a+k-i,b+j-l} = 0$$

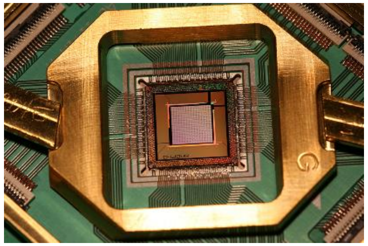
Problem size

Problem name	CAP	Magic Square
size	n=number of rows (and of columns)	n=number of rows (and of columns)
integer variables	n	n^2
QUBO variables	$n^2 + \frac{n^3(n-1)}{2}$	n^4
n=5	275	625
n=10	4600	10000
n=12	9648	20736

- Remark:
  - Fujitsu DAU has 8192 bits
  - D-Wave Advantage has 5000 qubits
- In fact more qubits are needed for D-Wave…

The D-Wave System

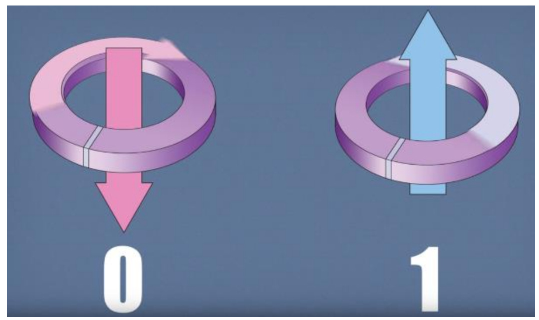




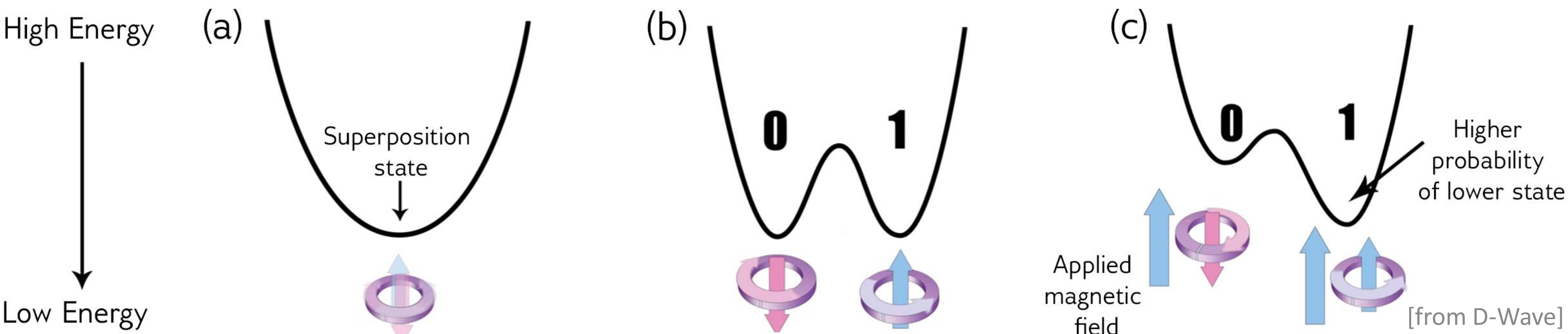
[from D-Wave]

Qubits in the D-Wave System

- Qubits are the lowest energy states of the superconducting loops that have a circulating current and corresponding magnetic field



- probability of falling into the 0 or the 1 state can be biased by applying an external magnetic field to the qubit (transverse field)



[from D-Wave]

QA in the D-Wave System

- Quantum Hamiltonian:

$$\mathcal{H}_{ising} = \underbrace{-\frac{A(s)}{2}\left(\sum_i \hat{\sigma}_x^{(i)}\right)}_{\text{Initial Hamiltonian}} + \underbrace{\frac{B(s)}{2}\left(\sum_i h_i \hat{\sigma}_z^{(i)} + \sum_{i>j} J_{i,j}\hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)}\right)}_{\text{Final Hamiltonian}}$$

where $\hat{\sigma}_{x,z}^{(i)}$ are Pauli matrices operating on a qubit q_i, and h_i and $J_{i,j}$ are the qubit biases and coupling strengths.

Nonzero values of h_i and $J_{i,j}$ are limited to those available in the working graph

[from D-Wave]

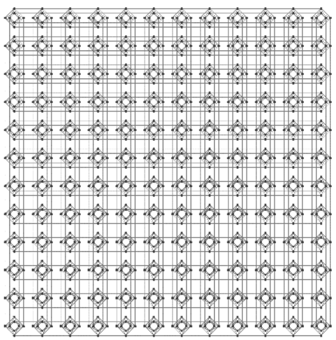
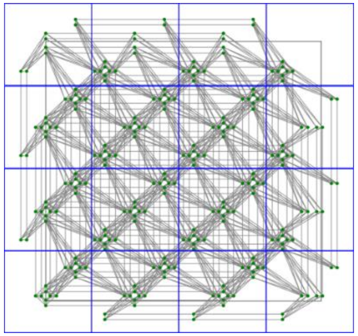
- By varying *A(s)* and *B(s)* over time the system moves from initial Hamiltonian to final Hamiltonian

D-Wave's QUBO solvers

- `DWaveSampler`:
  - Quantum hardware solver on the QPU
- `Hybrid Solver:` (version 1 and version 2)
  - decomposition-based, mixing classical and quantum executions
- classical computation solvers:
  - `Neal`: simulated annealing
  - `QBSolv`: decomposition-based tabu search

Minor Embedding

- Problems have to be cast to the actual hardware architecture (not fully-connected graph):

D-Wave 2000	D-Wave Advantage
2000 qubits	5000 qubits
 Chimera Architecture	 Pegasus Architecture
6-way connectivity	15-way connectivity

- Several physical qubits used to represent logical qubits: *chained* qubits for missing connections

Logical Qubits vs. Physical Qubits

- According to [Raymond 2021]:
  - D-Wave Advantage (5000 qubits) can encode complete graph up to 182 logical qubits (with maximal chain lengths of 17)
  - D-Wave 2000X (2000 qubits) can encode complete graph up to 64 logical (maximal chain length of 17)

- Drastically reduces problem size when using QA hardware, if full connectivity is needed, e.g. because of global constraints such as « one-hot » or « all-different »

Experiments on D-Wave QPU Solver

- 12x12 QAP problems (QAPLIB): Rou12, Had12

- Minor embedding possible
- but result is **infeasible solution:** not permutation
  - some qubit chains are broken…

- QPU solver cannot be used to solve a 12x12 QAP
- Need to use `Hybrid Solver` instead (mixing classical and quantum computations)

Quadratic Assignment Problem

QAP Problem	DAU	EO-QAP	D-WAVE Qbsolv	D-WAVE Hybrid Solver (30 sec.)	D-WAVE Hybrid Solver (200-300 sec.)
had12	3.392	120 (1 core) 0.00 (32cores)	613 1% from BKS	30 0.5% from BKS	200
rou12	0.448	0.013 (1 core)	135	30 3.4% from BKS	300

- DAU is Fujitsu's Digital Annealer Unit [Matsubara *et al.* 2019]
- EO-QAP is Extremal Optimization metaheuristic [Munera *et al.* 2016]
- D-WAVE Advantage (LEAP cloud access)
- D-WAVE QPU gives infeasible solution… (quantum computation time: 0.027s)

Constraint Satisfaction Problems

- Minor embedding for execution on QPU only for n=3
  Larger problems to be solved by `Qbsolv` or `Hybrid Solver`

- Magic Square:
  - n=4 solved by `Hybrid Solver` in 8 s., by `QBSolv` in 0.85 s.
  - For n ≥ 10 neither `Hybrid Solver` nor `QBSolv` can find a solution without any conflict

- Costas Arrays:
  - n=8 solved (1856 variables) in 4.89 s. by `Hybrid Solver` and 8.50 s. by `QBSolv`
  - N ≥ 10 cannot be solved to satisfiability by `Hybrid Solver` nor `QBSolv` (output solutions with ≈ 4 conflicts for n=10)

Conclusion

- QUBO: simple and expressive way to model combinatorial optimization or Constraint Satisfaction Problems
- QUBO: input language for quantum computers
- For problems with complex constraints, current QA hardware cannot yet compete with best heuristic methods
- But in the next years quantum hardware will improve, both in performance and size (number of qubits) …
- Competitive w.r.t. best classical methods in 5 years ?