QUANTUM FUTURE ACADEMY

Lecture: Quantum Annealing

2025 I DR. DENNIS WILLSCH

QUANTUM ANNEALING

Motivation and Contents

➢ **Main quantum computing paradigms:**

Digital (gate-based) Quantum Computers

Quantum Simulators

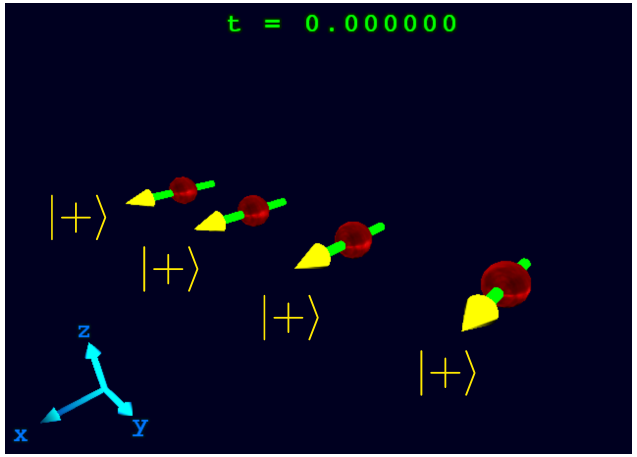
Quantum Annealers

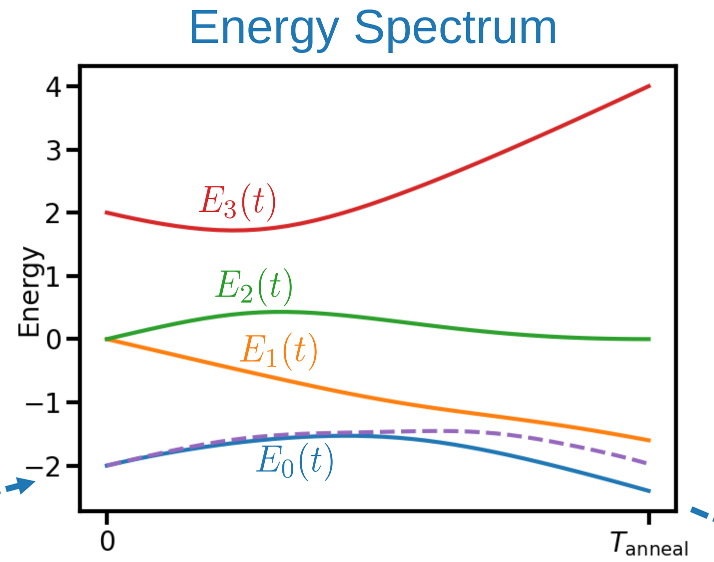
➢ Today, quantum annealers are the QC technology with the largest QPUs (~ 5000 physical qubits)

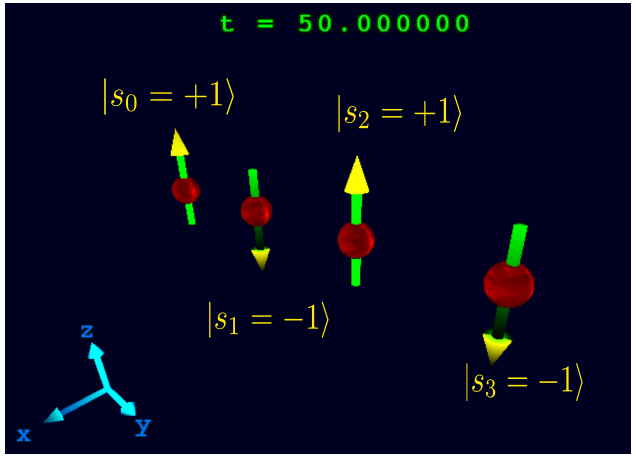
1. **What** do these devices do? → **Quantum Annealing and the Adiabatic Theorem**
2. **Which problems** can they solve? → **QUBO & Ising Problems**
3. **How** do we program them? → **D-Wave Ocean SDK**
4. **Bonus**: Relation between QA and digital QC → **Finding parameters for QAOA**
5. **Advanced Topics**: Coherent / Non-Equilibrium / Quasistatic Annealing

QUANTUM ANNEALING

Overview







Start of Annealing Process:

$$|\psi(0)\rangle = |+\cdots+\rangle = |+\rangle^{\otimes N}$$

Known initial state

Time

$$i\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle = H(t)\,|\psi(t)\rangle$$

Hamiltonian

End of Annealing Process:

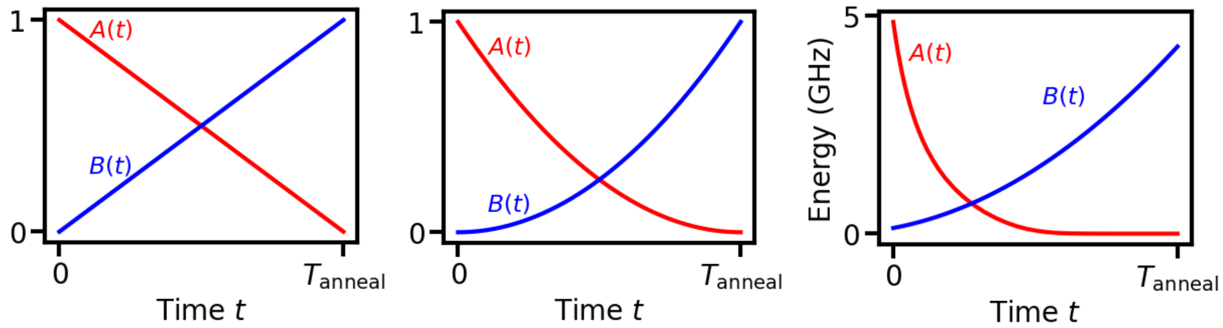
$$|\psi(T_{\text{anneal}})\rangle \overset{?}{\approx} |s_0 s_1 \cdots\rangle$$

Unknown solution to our problem

QUANTUM ANNEALING

What is the Hamiltonian?

$$H(t) = A(t)\underbrace{\left(-\sum_i \sigma_i^x\right)}_{H_{\mathrm{init}}} + B(t)\underbrace{\left(\sum_i h_i\sigma_i^z + \sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z\right)}_{H_{\mathrm{problem}}}$$



Linear Annealing Scheme

Quadratic Annealing Scheme

JUPSI Annealing Scheme

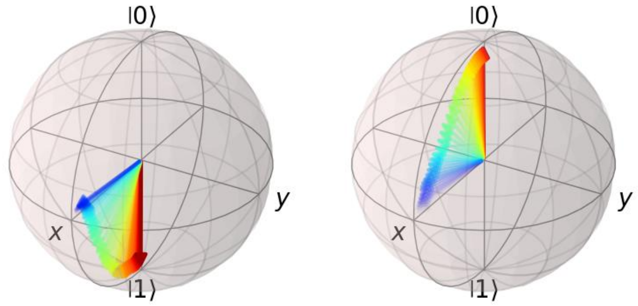
- System evolves from initial Hamiltonian H_{init} to final Hamiltonian H_{problem}
- Initial state $|+\rangle^{\otimes N}$ is the **ground state** of H_{init}

  eigenstate with lowest eigenvalue

- **Goal:** Final state shall be the ground state of H_{problem}
- Example for two qubits:

$$H(t) = A(t)(-\sigma_0^x - \sigma_1^x) + B(t)(h_0\sigma_0^z + h_1\sigma_1^z + J_{01}\sigma_0^z\sigma_1^z)$$

```
solve_tdse_qa(
    annealing_time=100,
    h0=-0.8, h1=-1.2, J01=2
)
```



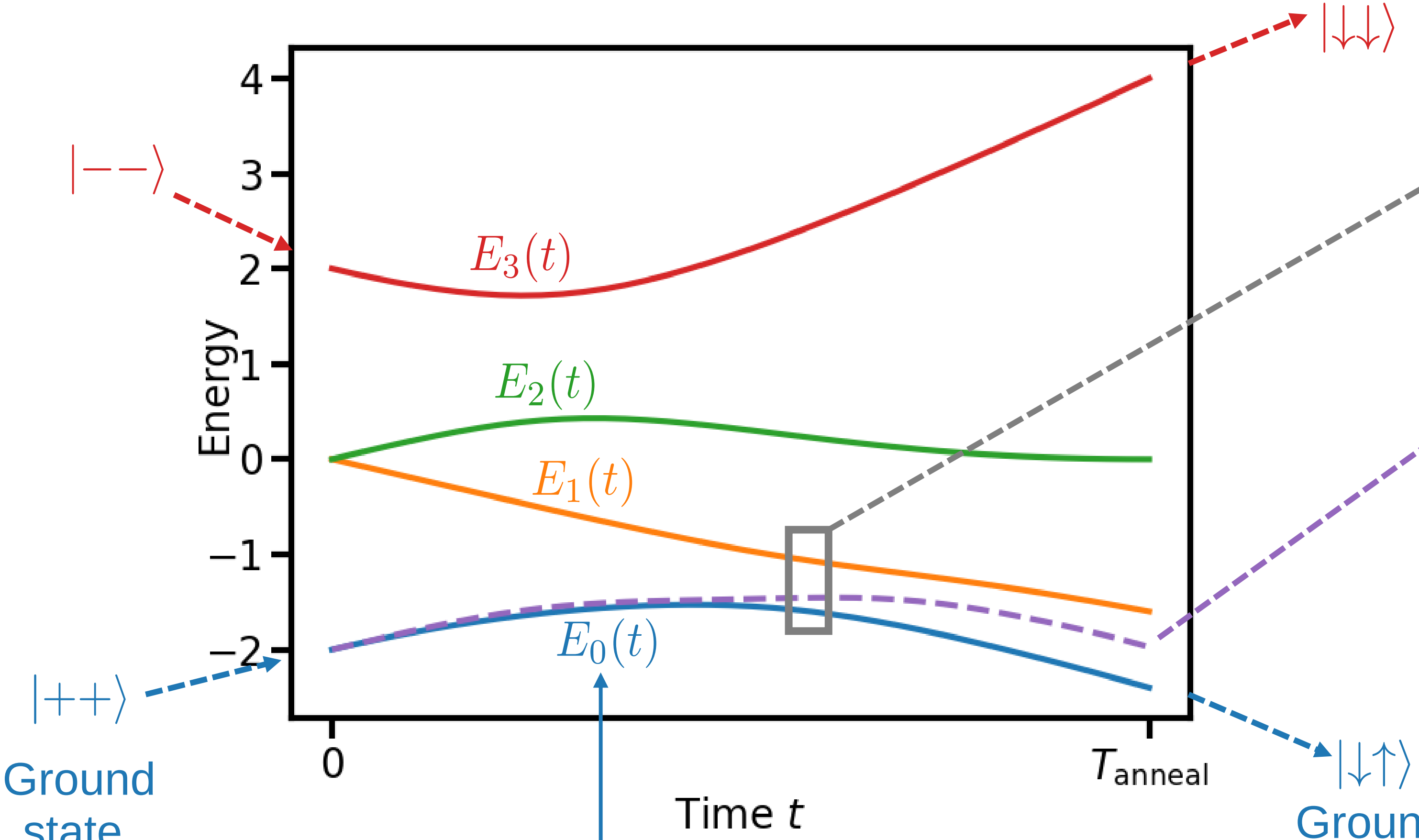
https://tinyurl.com/qc25-lecture

QC_Exercise_06

QUANTUM ANNEALING

$$H(t) = A(t)(-\sigma_0^x - \sigma_1^x) + B(t)(h_0\sigma_0^z + h_1\sigma_1^z + J_{01}\sigma_0^z\sigma_1^z)$$

Two-Qubit Example



Ground state at t = 0

Ground state of $H(t)$ at each intermediate time t

Ground state at t = T

- **Minimum Energy Gap**

$$\Delta E = \min_{t\in[0,T_{\mathrm{anneal}}]} (E_1(t) - E_0(t))$$

- **Intermediate Energy Expectation Value**

$$E(t) = \langle\psi(t)|H(t)|\psi(t)\rangle$$

- **Adiabatic Theorem:** If the time evolution is **slow enough**, the system always stays in the ground state of $H(t)$

QUANTUM ANNEALING

Adiabatic Theorem: What does it state?

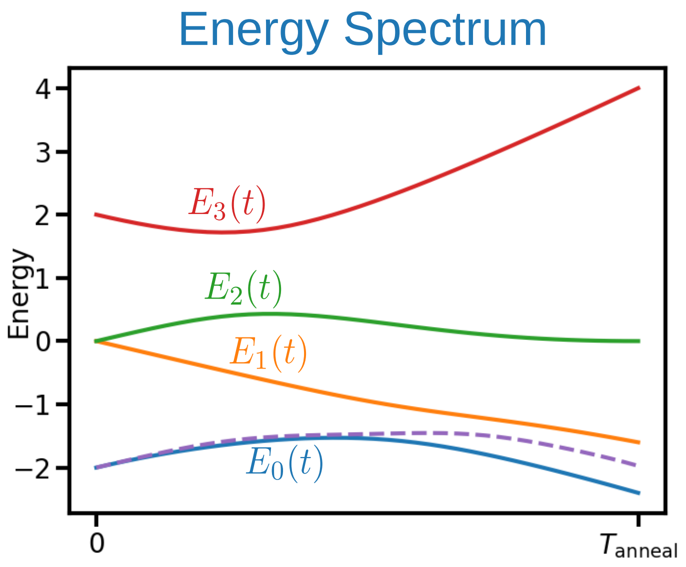
- **Traditional statement**: If

Intuition: Hamiltonian should change slowly in time (as measured by the gap)

$$\forall m \neq n : \quad \max_{t\in[0,T_{\mathrm{anneal}}]} \frac{|\langle E_m(t)|\frac{\mathrm{d}H}{\mathrm{d}t}|E_n(t)\rangle|}{(E_m(t)-E_n(t))^2} \ll 1$$

Intuition: Energy gap should be large

then a system initialized in $|E_n(0)\rangle$ will stay in the eigenstate $|E_n(t)\rangle$



Otherwise it would drive fast Rabi oscillations between $|E_m(t)\rangle$ and $|E_n(t)\rangle$

- Careful analysis shows an **additional condition** is needed:
  - Hamiltonian must not contain oscillating terms with frequencies $\omega \approx \langle E_m(t) - E_n(t)\rangle$ $\longleftarrow \langle\cdot\rangle = \frac{1}{t}\int_0^t \cdot\,\mathrm{d}t$

Amin 2009, https://doi.org/10.1103/PhysRevLett.102.220401

- In the past decades, many **more quantitative** adiabatic theorems have been proven (but much less intuitive)

Albash & Lidar, RMP 90, 015002 (2018), https://doi.org/10.1103/RevModPhys.90.015002

QUANTUM ANNEALING

$$H(t) = \underbrace{A(t)\left(-\sum_i \sigma_i^x\right)}_{H_{\text{init}}} + \underbrace{B(t)\left(\sum_i h_i \sigma_i^z + \sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z\right)}_{H_{\text{problem}}}$$

Which Type of Problems are Solved?

- **Goal:** Final state shall be the ground state of H_{problem}
  - Equivalent to the solution of the **Ising Problem**

$$\min_{s_i=\pm 1}\left(\sum_{i=0}^{N-1} h_i s_i + \sum_{i<j}^{N-1} J_{ij} s_i s_j\right)$$

  - Equivalent to the solution of the **QUBO Problem**

$$\min_{x_i=0,1}\left(\sum_{i\le j}^{N-1} x_i Q_{ij} x_j\right)$$

Quadratic
Unconstrained
Binary
Optimization

Example $\min(2s_0 - 3s_1) = -5$ ← $(s_0, s_1) = (-1, +1)$

$\min(4x_0 - 6x_1) + 1 = -5$ ← $(x_0, x_1) = (0, 1)$

$$s_i = 2x_i - 1$$
$$x_i = \frac{1+s_i}{2}$$

Example $\min(-s_0 s_1) = -1$ ← $(s_0, s_1) = (-1, -1)$, $(s_0, s_1) = (+1, +1)$

Antiferro-magnetic coupling

$\min(-4x_0x_1 + 2x_0 + 2x_1) - 1 = -1$ ← $(x_0, x_1) = (0, 0)$, $(x_0, x_1) = (1, 1)$

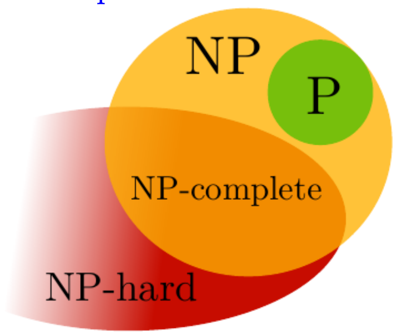
$x_i = 1 \Leftrightarrow s_i = +1$	
$x_i = 0 \Leftrightarrow s_i = -1$	

"QA convention"

QUANTUM ANNEALING

$$H(t) = A(t)\underbrace{\left(-\sum_i \sigma_i^x\right)}_{H_\text{init}} + B(t)\underbrace{\left(\sum_i h_i\sigma_i^z + \sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z\right)}_{H_\text{problem}}$$

Which Type of Problems are Solved?



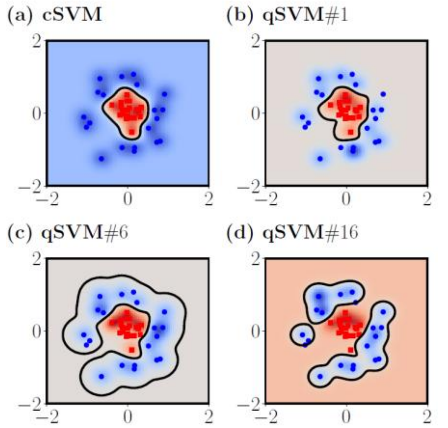
- **Goal:** Final state shall be the ground state of H_problem
  - Equivalent to the solution of the **Ising Problem**

$$\min_{s_i=\pm 1}\left(\sum_{i=0}^{N-1} h_i s_i + \sum_{i<j}^{N-1} J_{ij} s_i s_j\right)$$

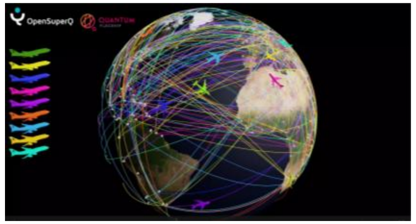
  - Equivalent to the solution of the **QUBO Problem**

$$\min_{x_i=0,1}\left(\sum_{i\le j}^{N-1} x_i Q_{ij} x_j\right)$$

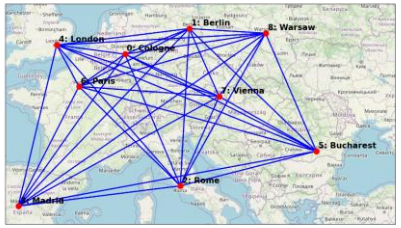
Quadratic
Unconstrained
Binary
Optimization



- Ising / QUBO are **NP-hard** → Can in principle map any NP problem to Ising / QUBO



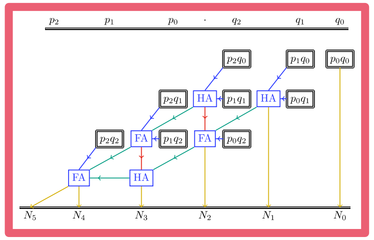
Tail Assignment



Traveling Salesman

Remote Sensing

Garden Optimization



Integer Factorization

Quantum Machine Learning

- **But:** Not guaranteed to find a quantum advantage!

References and more information:
https://doi.org/10.34734/FZJ-2025-02577

QUANTUM ANNEALING

$$H(t) = A(t)(-\sigma_0^x - \sigma_1^x) + B(t)(h_0\sigma_0^z + h_1\sigma_1^z + J_{01}\sigma_0^z\sigma_1^z)$$

Programming a D-Wave QPU: 2-Qubit Example

```
from dwave.system import DWaveSampler, EmbeddingComposite
import dwave_networkx as dnx
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

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

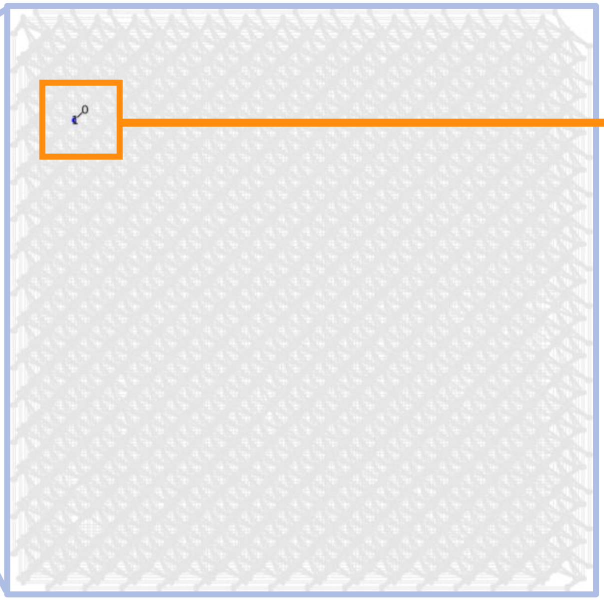
```
result = solver.sample_ising(
    h={0: -0.8, 1: -1.2},
    J={(0,1): 2},
    num_reads=1000,
    annealing_time=20,
    return_embedding=True
)
```

```
result.to_pandas_dataframe()
```

	0	1	chain_break_fraction	energy	num_occurrences
0	-1	1	0.0	-2.4	859
1	1	-1	0.0	-1.6	141

Solution on JUPSI

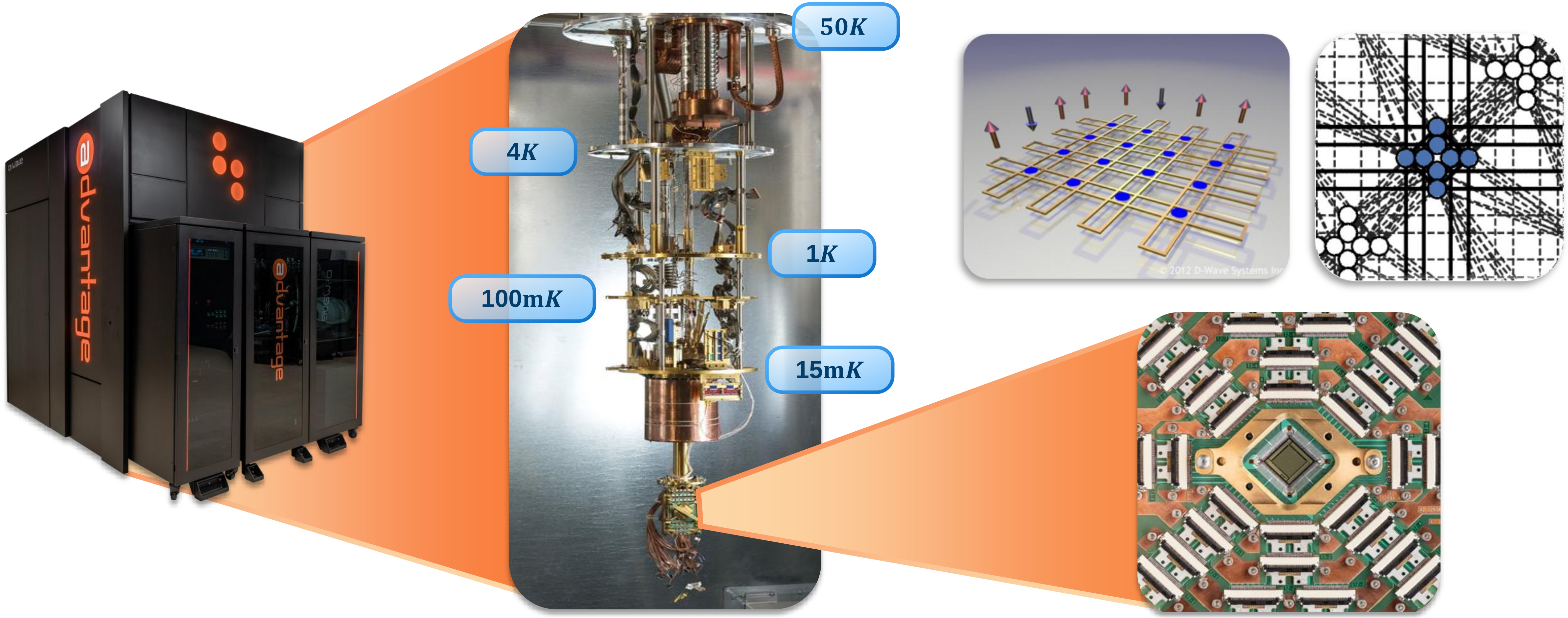
```
plt.figure(figsize=(10,10))

dnx.draw_pegasus_embedding(
    sampler.to_networkx_graph(),
    result.info['embedding_context']['embedding'],
    show_labels = True,
    crosses = True,
    node_size = 20
)
```



These two physical qubits were selected for our problem Hamiltonian

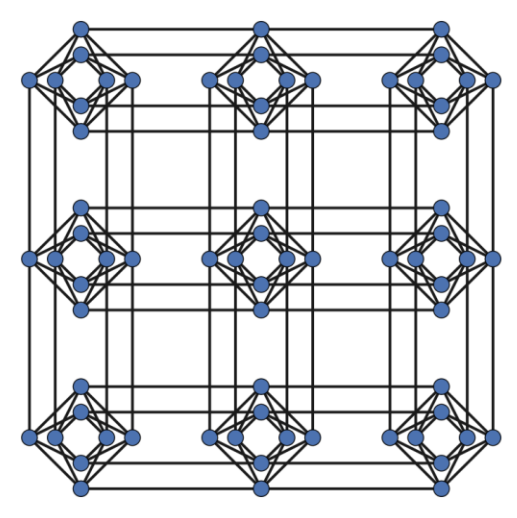
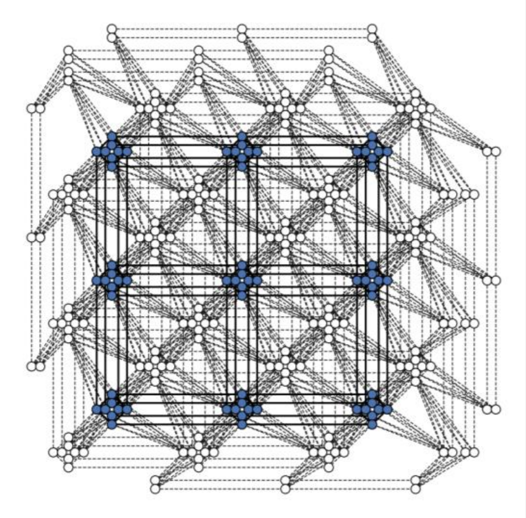
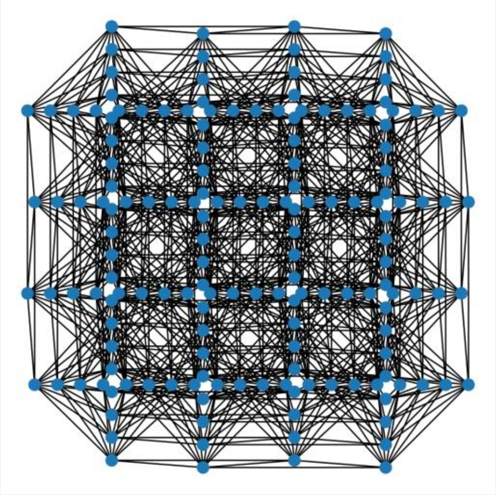
QUANTUM ANNEALING

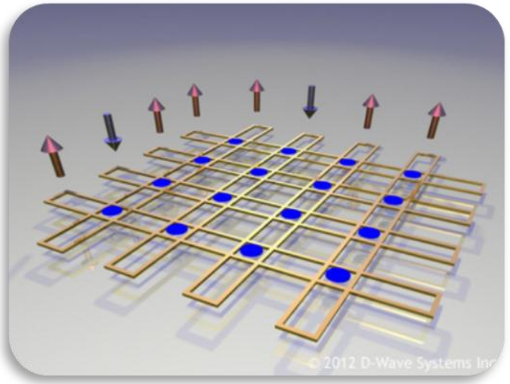
What is inside?

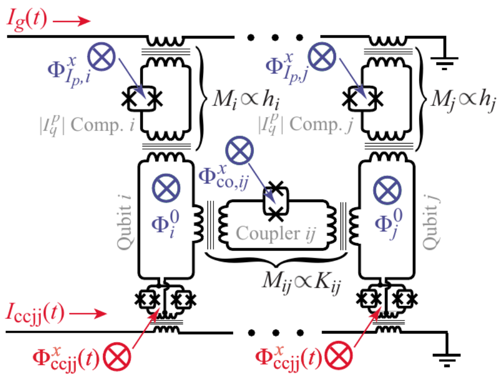


QUANTUM ANNEALING

What is inside?

QPU:	DW2000Q	Advantage	Advantage2
Graph:	Chimera	Pegasus	Zephyr
#Qubits:	~2000	~5600	~4600
#Couplers:	~6000	~40000	~42000
Connectivity:	**6**	**15**	**20**
Generation:	previous	current	current/future
			

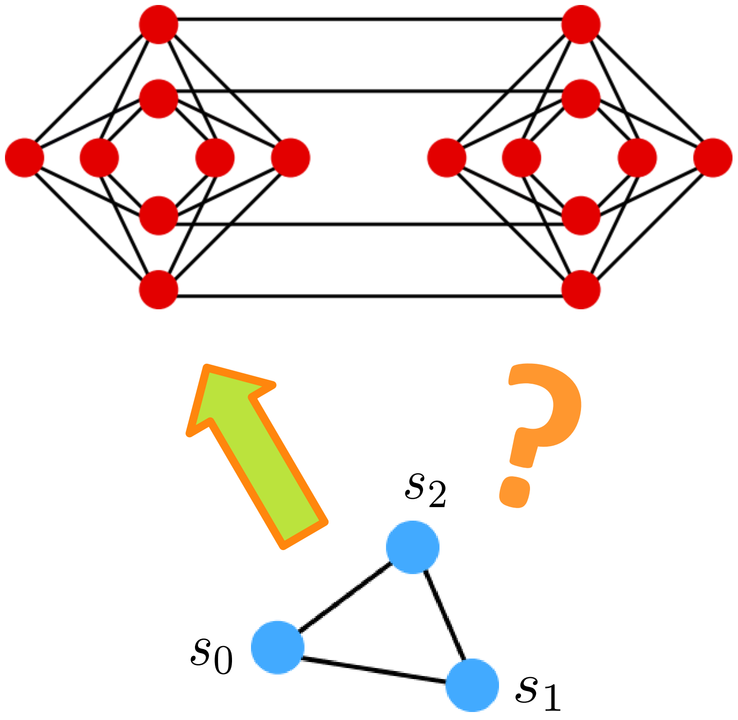




see reference slide at the end

QUANTUM ANNEALING

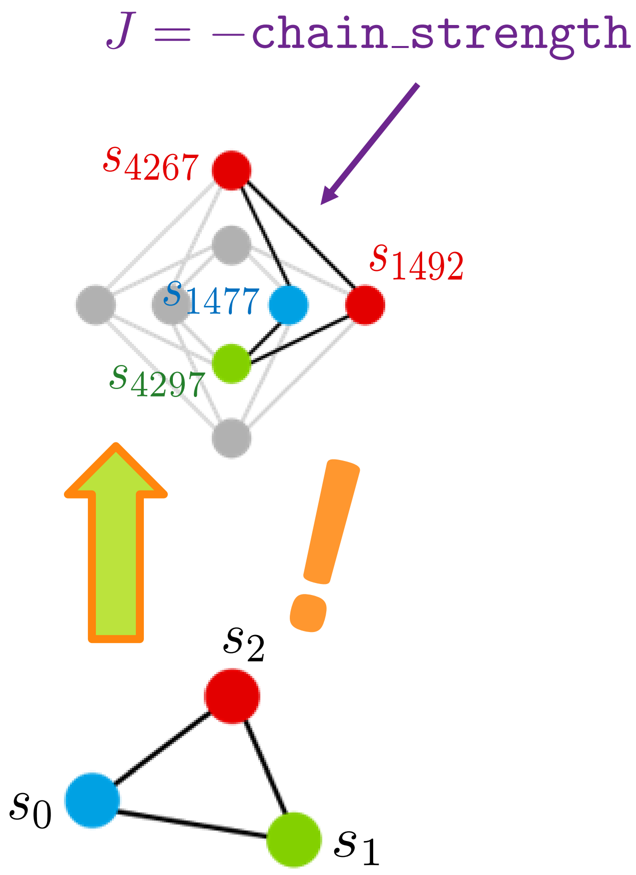
Embedding: What if my problem does not fit on the QPU?



Represent
logical qubit
s_2

by a **chain** of
physical qubits

```
embedding = {
    0: [1477],
    1: [4297],
    2: [4267, 1492],
}
```



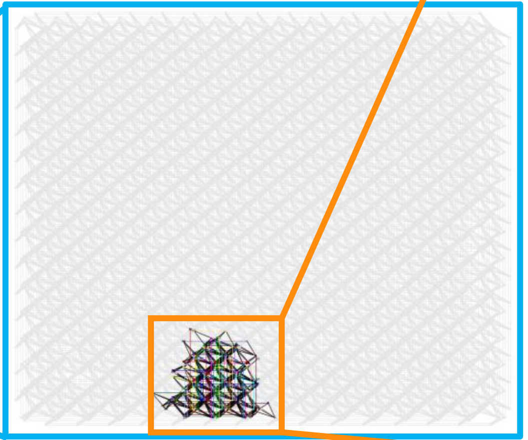
QUANTUM ANNEALING

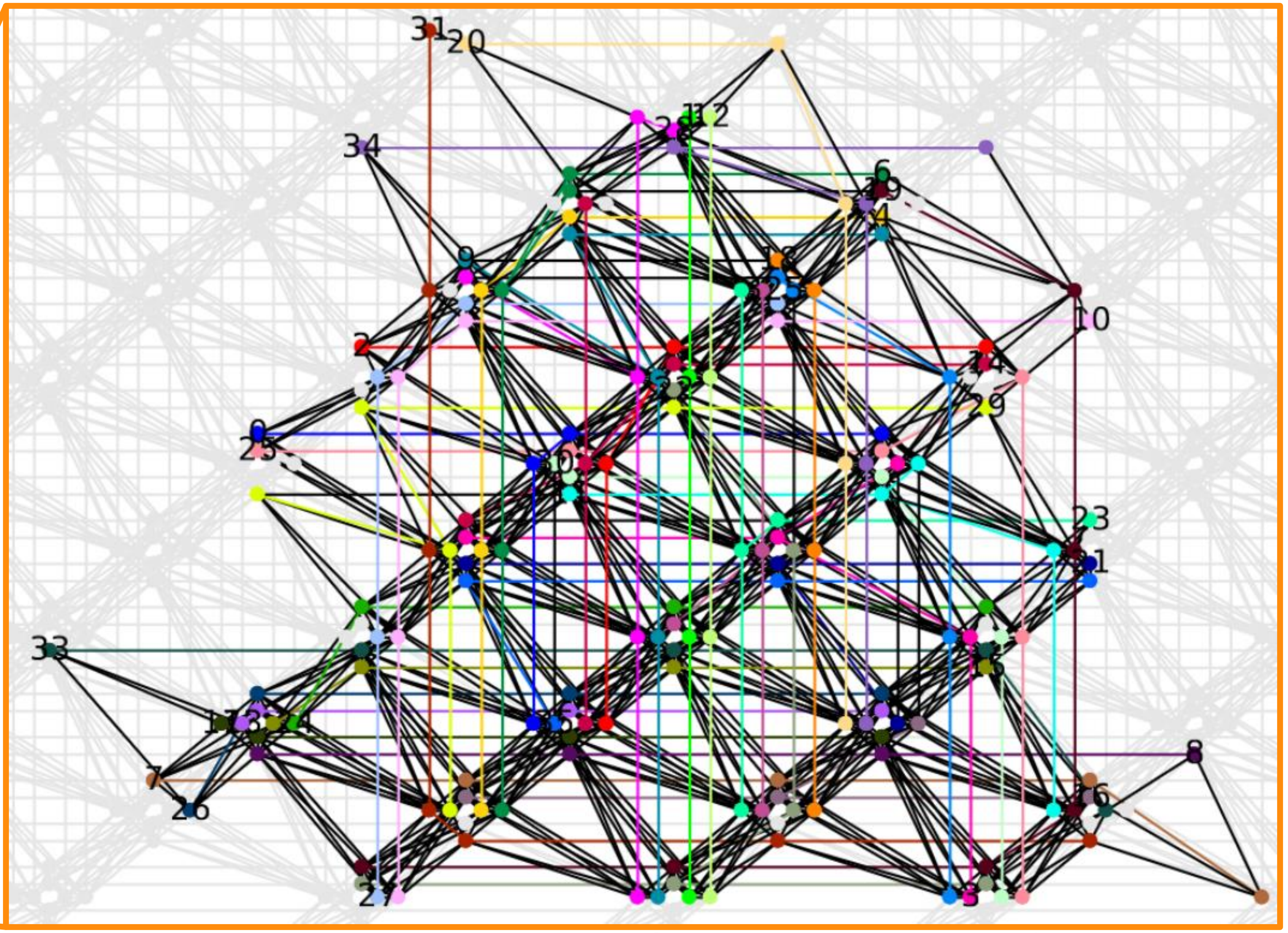
Embedding: Visualization

```python
result.info['embedding_context']
```

```
{'embedding': {0: (5344, 5346, 1094, 5345, 1093),
  1: (1181, 1184, 1183, 1182),
  2: (5285, 5287, 1139, 1138, 5286),
              ⋮
 34: (5150, 5152, 5151, 1289, 1288, 1287),
 35: (1109, 5452, 5451, 5450)},
'chain_break_method': 'majority_vote',
'embedding_parameters': {},
'chain_strength': 30}
```

```python
plt.figure(figsize=(30,25))

dnx.draw_pegasus_embedding(
    sampler.to_networkx_graph(),
    emb = result.info['embedding_context']['embedding'],
    show_labels = True,
    crosses = True,
    node_size = 20,
)
```



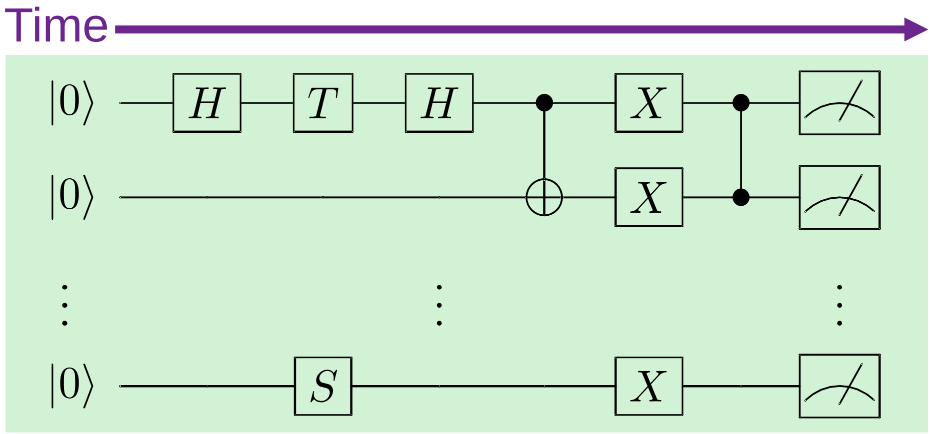


QUIZ

QUANTUM ANNEALING

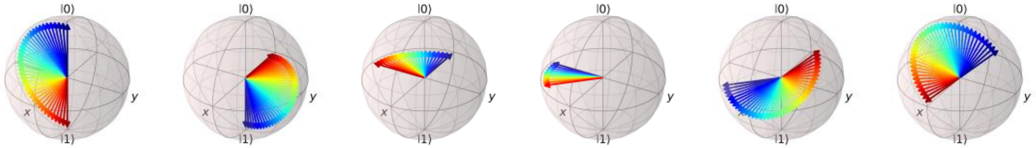
Comparison to Gate-Based Quantum Computing

- **Gate-Based Quantum Computing**



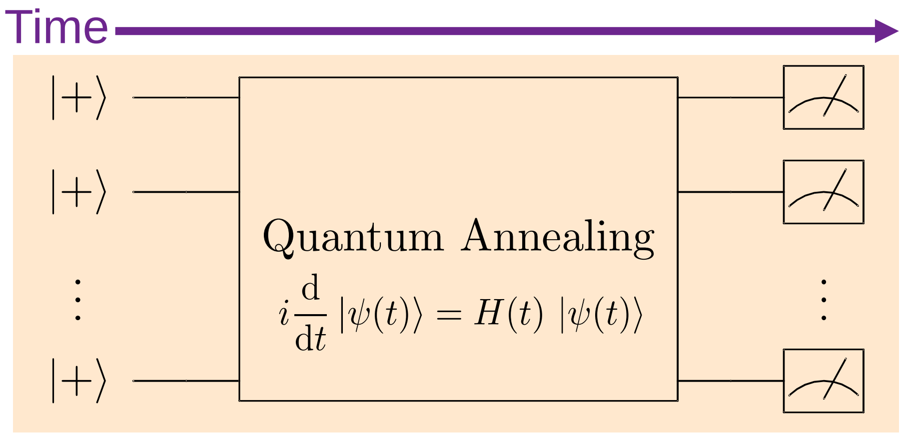
- Each gate:

  Individual, programmable time evolution



- Initial state: $|0\rangle^{\otimes N}$

- **Quantum Annealing**



- Only one "gate": Intuition: Simpler to manufacture

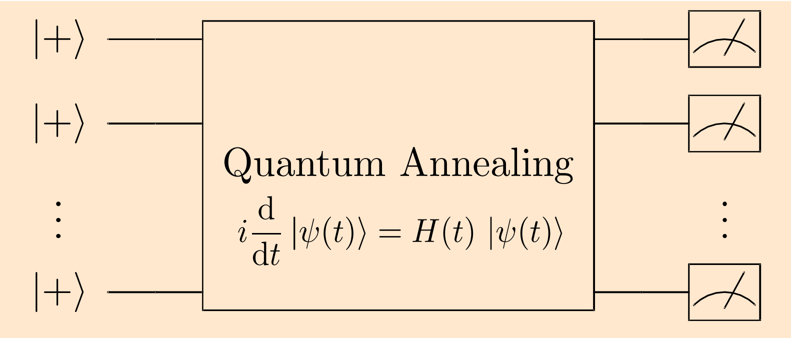
  Natural time evolution according to

$$H(t) = A(t)\left(-\sum_i \sigma_i^x\right) + B(t)\left(\sum_i h_i\sigma_i^z + \sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z\right)$$

- Initial state: $|+\rangle^{\otimes N}$

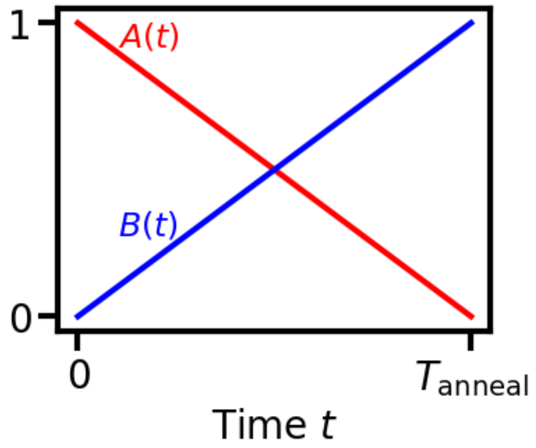
QUANTUM ANNEALING

Discretizing Quantum Annealing



- Quantum Annealing Hamiltonian

$$H(t) = A(t)\underbrace{\left(-\sum_i \sigma_i^x\right)}_{H_M} + B(t)\underbrace{\left(\sum_i h_i\sigma_i^z + \sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z\right)}_{H_W} = -A(t)H_M + B(t)H_W$$



- Time evolution operator for one time step τ:

$$e^{-i\,H(t)\,\tau} = e^{-i\,(-A(t)H_M + B(t)H_W)\,\tau} \approx e^{i\,A(t)\,H_M\,\tau}\,e^{-i\,B(t)\,H_W\,\tau}$$

Suzuki-Trotter product formula
(1st-order decomposition of matrix exponential,
ok for small time step τ)

- Total time evolution discretized in p time steps $(t_1, t_2, \ldots, t_p)$ of size τ

$$|\psi(T)\rangle \approx e^{iA(t_p)\tau H_M}\,e^{-iB(t_p)\tau H_W}\cdots e^{iA(t_1)\tau H_M}\,e^{-iB(t_1)\tau H_W}\,|+\cdots+\rangle$$

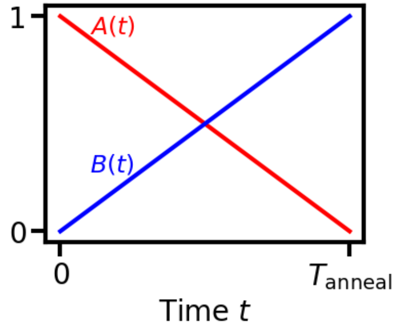
← "Trotterization"

Montañez-Barrera & Michielsen, npj 11, 113 (2025)
https://doi.org/10.1038/s41534-025-01082-1

QUANTUM ANNEALING

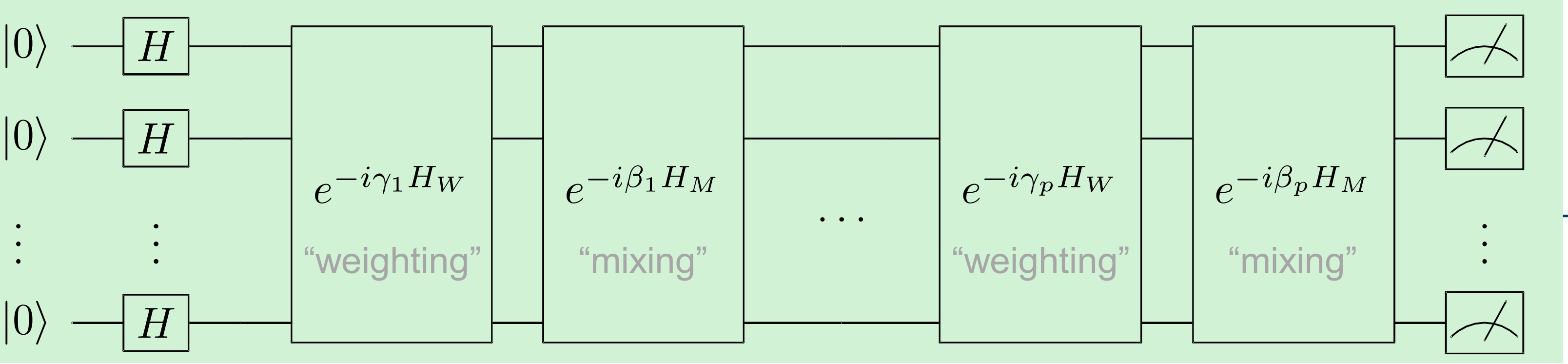
Discretizing Quantum Annealing → Finding good QAOA parameters

- Quantum Annealing time evolution — discretized in p time steps $(t_1, t_2, \ldots, t_p)$ of size τ



$$|\psi(T)\rangle \approx e^{iA(t_p)\tau H_M}\, e^{-iB(t_p)\tau H_W} \cdots e^{iA(t_1)\tau H_M}\, e^{-iB(t_1)\tau H_W}\, |+\cdots+\rangle$$

- Compare this with the QAOA circuit:



$|+\cdots+\rangle$ $\quad e^{-iB(t_1)\tau H_W}$ $\quad e^{iA(t_1)\tau H_M}$ $\quad e^{-iB(t_p)\tau H_W}$ $\quad e^{iA(t_p)\tau H_M}$

$\gamma_k = B(t_k)\tau$ goes to 1
$\beta_k = -A(t_k)\tau$ goes to 0

	$k=1$	$k=2$	$k=3$	$\cdots$	$k=p$
β_k	-0.9	-0.8	-0.7	$\cdots$	0
γ_k	0.1	0.2	0.3	$\cdots$	1

- So the QAOA can be motivated as **discretized Quantum Annealing**
  → and the **individual time steps** are the parameters to be optimized!

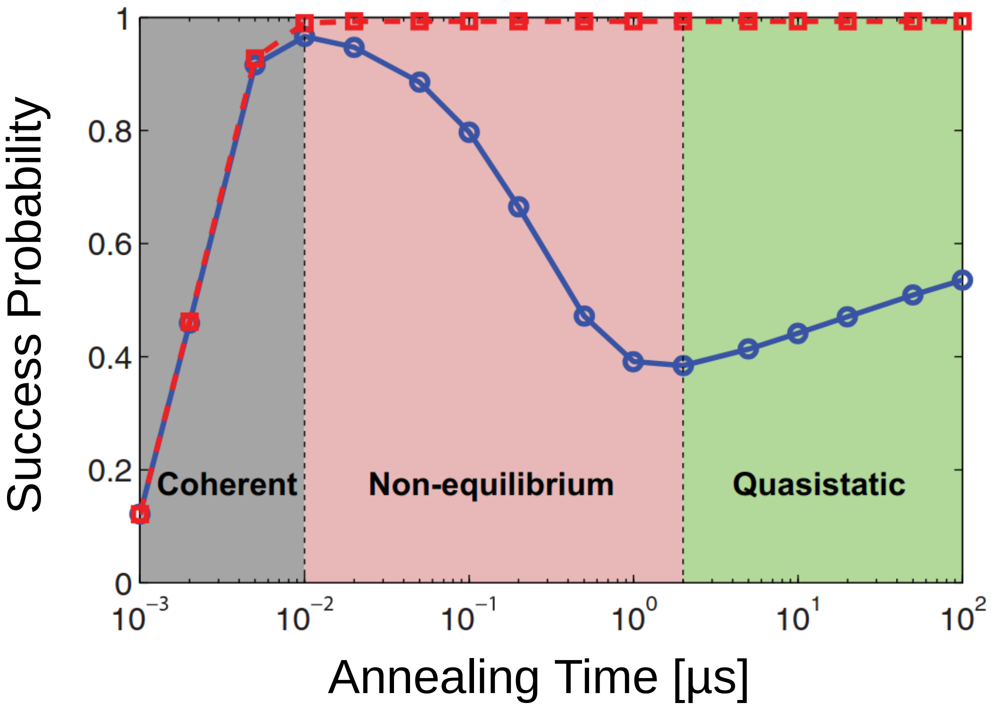
JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

QUANTUM ANNEALING

Advanced Topics: Annealing Regimes

Quantum System with 2^N possible states ⟷ Interaction ⟷ **Environment** at temperature T

- Recently, the **fast annealing** feature was introduced for D-Wave Advantage QPUs
- With fast annealing, we can anneal as quickly as **5 ns** (compared to the normal minimum 500 ns)
- One can show that for very short annealing, the success probability can **improve** again
  (and computation speed matters!)



Explanation:

- **Coherent regime:**
  - Anneal dominated by “quantum” dynamics: $i\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(t)\rangle = H(t)\,|\psi(t)\rangle$
  - Intuition: so fast that the system is not affected by environment
- **Non-equilibrium regime:**
  - Environment excites the system into higher-energy states
- **Quasistatic:**
  - System can relax to lower-energy states again (T is quite 🥶)

Amin, PRA 92, 052323 (2015), http://dx.doi.org/10.1103/PhysRevA.92.052323

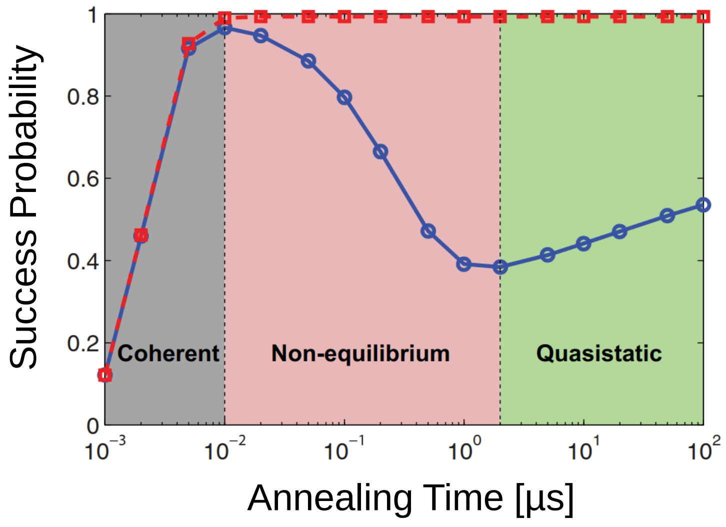
QUANTUM ANNEALING

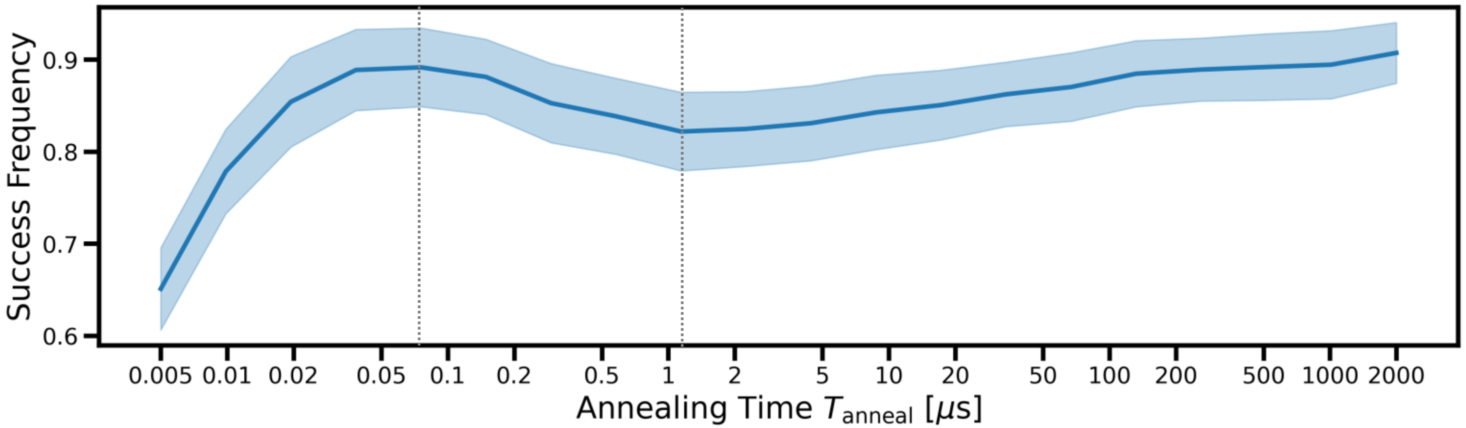
Experimental Results

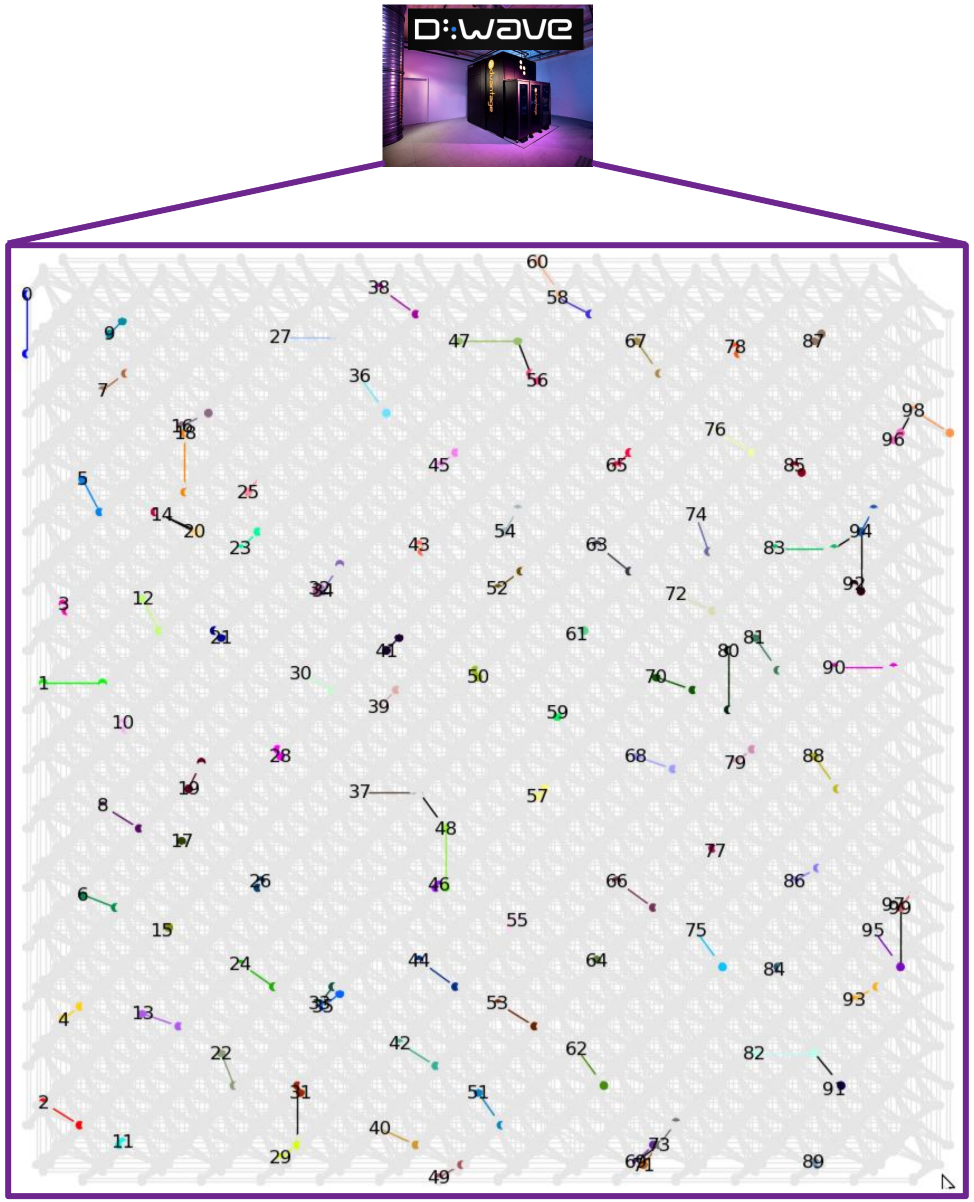
By

- **averaging** over many qubit pairs on the QPU
- **repeating** the experiment a few times

we can observe the theoretically expected behavior:







https://tinyurl.com/qc25-lecture

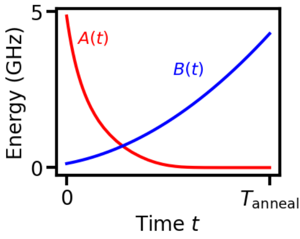
QC_07_Lecture_Fast-and-Reverse-Annealing

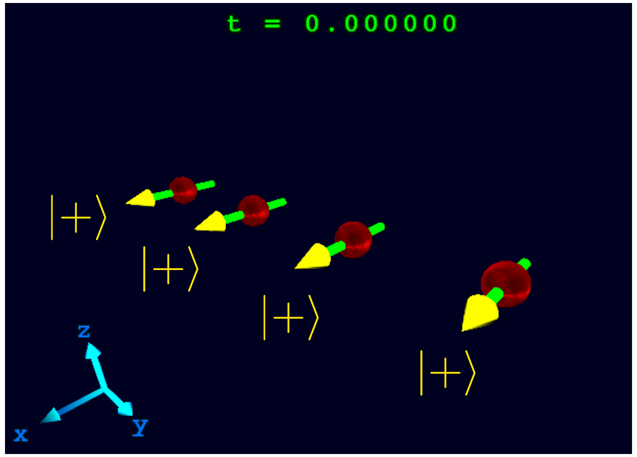
JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

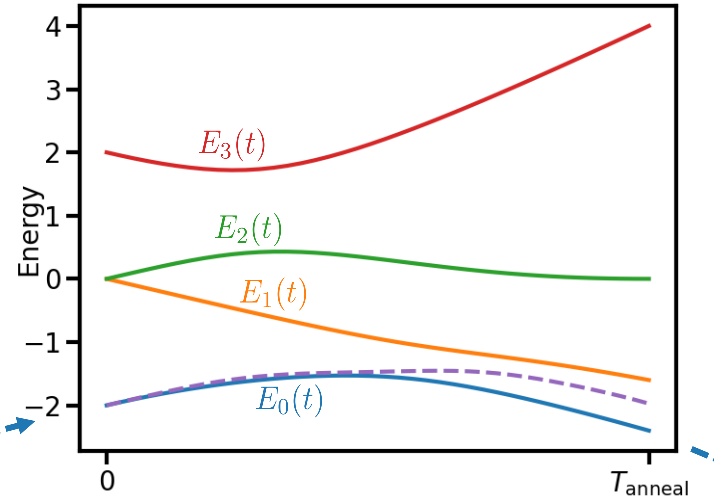
QUANTUM ANNEALING

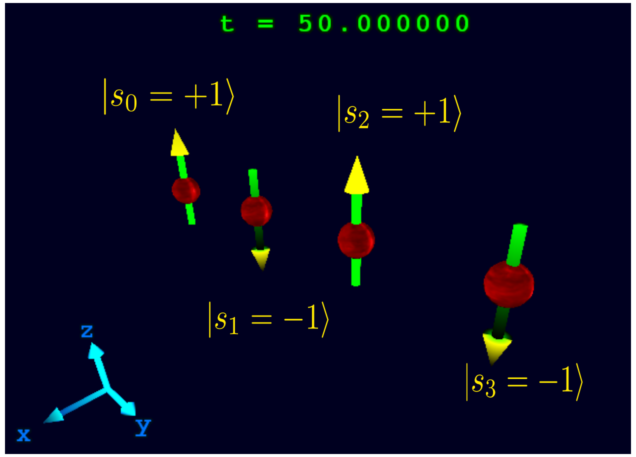
Summary

$$H(t) = A(t)\underbrace{\left(-\sum_i \sigma_i^x\right)}_{H_{\text{init}}} + B(t)\underbrace{\left(\sum_i h_i\sigma_i^z + \sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z\right)}_{H_{\text{problem}}}$$









Time

Start of Annealing Process:

$$H(0) = H_{\text{init}}$$

$$|\psi(0)\rangle = |+\cdots+\rangle = |+\rangle^{\otimes N}$$

Known initial state

$$i\frac{\mathrm{d}}{\mathrm{d}t}|\psi(t)\rangle = H(t)\,|\psi(t)\rangle$$

Adiabatic Theorem

End of Annealing Process:

$$H(T_{\text{anneal}}) = H_{\text{problem}}$$

$$|\psi(T_{\text{anneal}})\rangle \overset{?}{\approx} |s_0 s_1\cdots\rangle$$

Ising:

$$\min_{s_i=\pm 1}\left(\sum_{i=0}^{N-1} h_i s_i + \sum_{i<j}^{N-1} J_{ij} s_i s_j\right)$$

THANK YOU FOR YOUR ATTENTION

JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

FOR REFERENCE

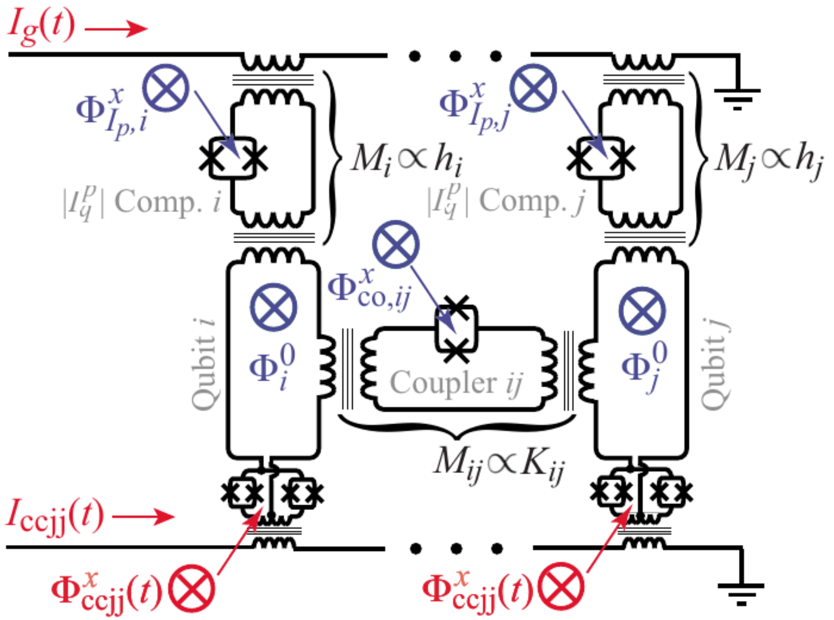
D-Wave Flux Qubit Hamiltonian

- Desired quantum annealing Hamiltonian:

$$H(s(t)) = \sum_i -\frac{A(s(t))}{2}\,\sigma_i^x + \sum_i \frac{B(s(t))}{2}\,h_i\,\sigma_i^z + \sum_{i<j} \frac{B(s(t))}{2}\,J_{ij}\,\sigma_i^z\sigma_j^z$$

- Given flux-qubit Hamiltonian:

$$H(s(t)) = \sum_i -\frac{\Delta_q(\Phi_{\mathrm{ccjj}}(s(t)))}{2}\,\sigma_i^x + \sum_i |I_p(\Phi_{\mathrm{ccjj}}(s(t)))|\,\left(\Phi_i^x(s(t)) - \Phi_i^{x,\mathrm{offset}}\right)\sigma_i^z$$
$$+ \sum_{i<j} |I_p(\Phi_{\mathrm{ccjj}}(s(t)))|^2\, M_{ij}(\Phi^x_{\mathrm{co},ij})\,\sigma_i^z\sigma_j^z$$



Harris, PRB 82, 024511 (2010), http://dx.doi.org/10.1103/PhysRevB.82.024511

Need fast cancellation current to obtain a constant h!

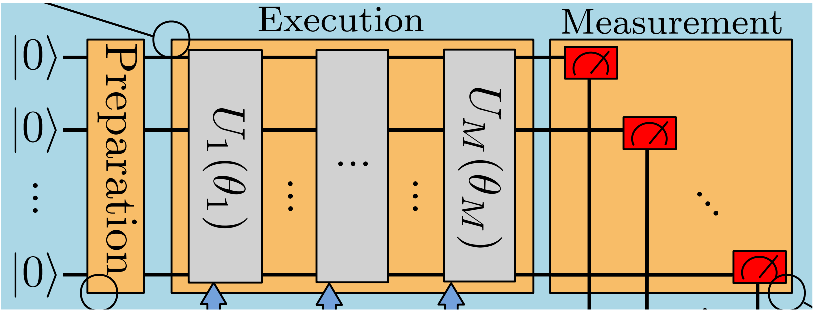
flux_bias

- Quantum annealing parameters from physically controllable magnetic fluxes and currents:

$$A(s(t)) = \Delta_q(\Phi_{\mathrm{ccjj}}(s(t)))$$

$$h_i = \frac{2}{B(s(t))}|I_p(\Phi_{\mathrm{ccjj}}(s(t)))|\,\left(\Phi_i^x(s(t)) - \Phi_i^{x,\mathrm{offset}}\right) = \frac{M_i(\Phi^x_{I_p,i})\,I_g(s(t)) - \Phi_i^{x,\mathrm{offset}}}{M_{\mathrm{AFM}}|I_p(\Phi_{\mathrm{ccjj}}(s(t)))|}$$

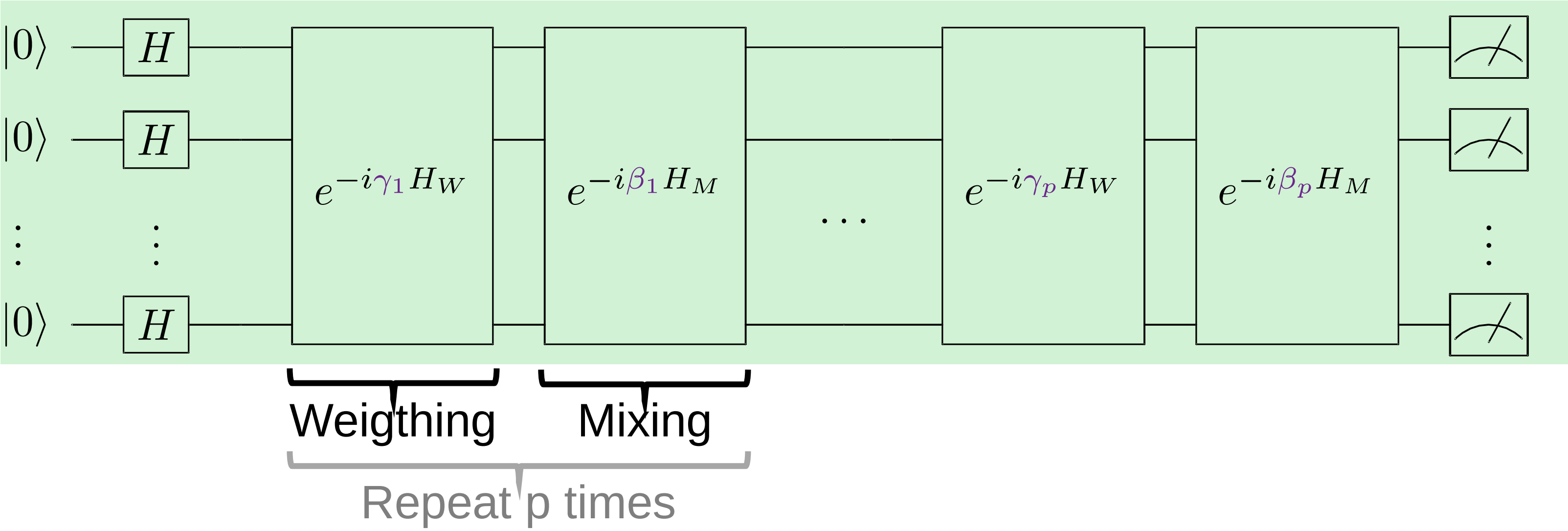
$$B(s(t)) = 2M_{\mathrm{AFM}}|I_p(\Phi_{\mathrm{ccjj}}(s(t)))|^2 \qquad J_{ij} = \frac{M_{ij}(\Phi^x_{\mathrm{co},ij})}{M_{\mathrm{AFM}}}$$



FOR REFERENCE

QAOA — Quantum Approximate Optimization Algorithm

- The **QAOA** is one of the most popular variational quantum algorithms
- The QAOA has 2p **variational parameters** $\vec{\beta} = (\beta_1, \ldots, \beta_p)$ and $\vec{\gamma} = (\gamma_1, \ldots, \gamma_p)$



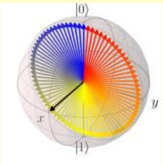
- It consists of p **QAOA layers** with 2 alternating steps:

Weighting Step

$$H_W = \sum_i h_i \sigma_i^z + \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z$$

Mixing Step

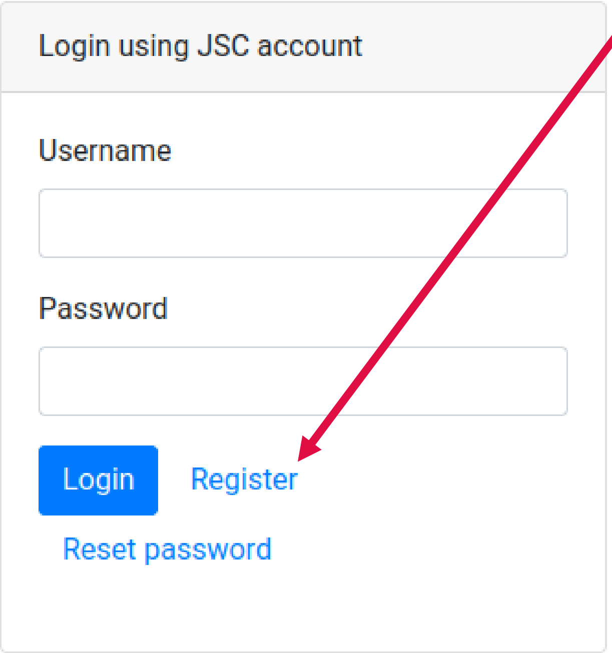
$$H_M = \sum_i \sigma_i^x$$



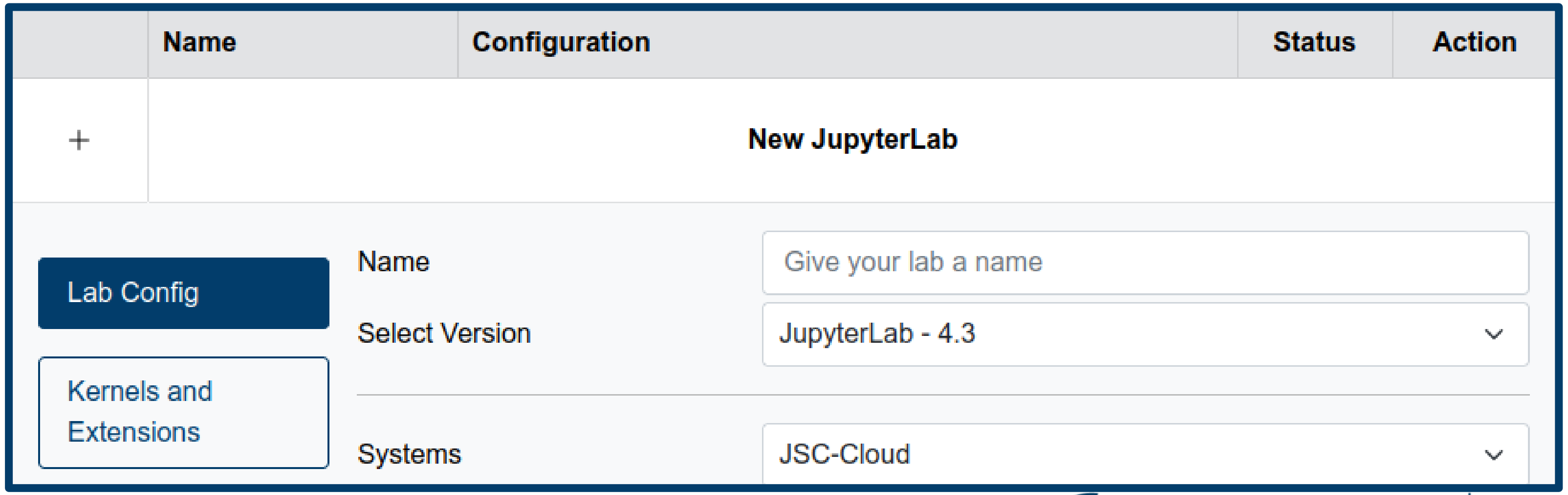
Reference and more information:
https://go.fzj.de/juniq-docs
https://go.fzj.de/juniq-getting-started

QUANTUM ANNEALING

Setup Access to JUNIQ Cloud



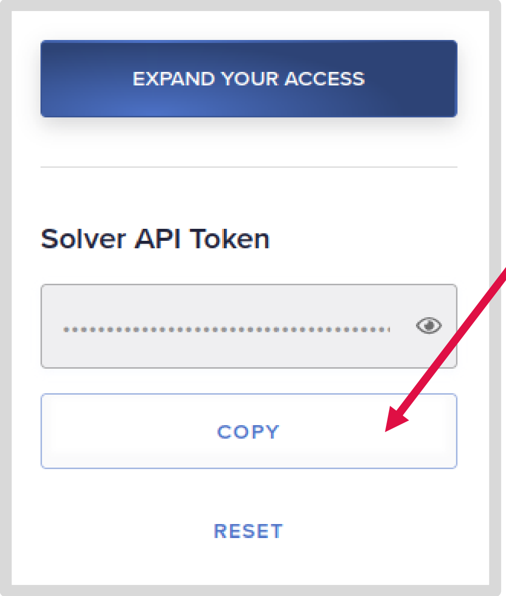
1. Create a **Judoor** account at https://judoor.fz-juelich.de/ (if you don’t have one yet).
2. Join the **juniq-lecture** project by submitting a join request at https://judoor.fz-juelich.de/projects/join/juniq-lecture/
3. Wait some time for us to admit you to the **juniq-lecture** project
4. Sign the usage agreement in Judoor under “Systems”
5. Login to JUNIQ at https://juniq.fz-juelich.de/ with your Judoor account.
6. Create a new Jupyter Lab on JSC-Cloud:



Reference and more information:
https://go.fzj.de/juniq-docs
https://go.fzj.de/juniq-getting-started

QUANTUM ANNEALING

Setup Access to the Quantum Annealer



1. Create **D-Wave Leap Account** at https://cloud.dwavesys.com/leap/
2. Follow the Leap email to join juniq-lecture on Leap (can take 2 days, check Spam/Junk)
3. When that's done, on the left side in D-Wave Leap, copy the juniq-lecture **API Token**

   NOTE: Never share your token!
4. In https://juniq.fz-juelich.de/ click on File → New → Terminal
5. Type ml Dwave and dwave config create and paste your juniq-lecture **API Token:**

```
jovyan@JSC-Cloud:~$ ml DWave
jovyan@JSC-Cloud:~$ dwave config create
Using the simplified configuration flow.
Try 'dwave config create --full' for more options.

Creating new configuration file: /home/jovyan/.config/dwave/dwave.conf
Updating existing profile: defaults
Solver API token [skip]: jV1M-[redacted]
Configuration saved.
```

6. Check with dwave ping or cat .config/dwave/dwave.conf whether it worked