APPLICATIONS ON QUANTUM ANNEALERS AT FZJ

JUNIQ: Jülich Unified INfrastructure for Quantum Computing

DR. DENNIS WILLSCH

WE COMPUTE WITH QUANTUM COMPUTERS

supported by Bundesministerium für Bildung und Forschung
Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen

CONTENTS

1. Airline Scheduling
2. Traveling Salesman
3. Garden Optimization
4. Factorization
5. 2-Satisfiability
6. QSVM
7. QBM

optimization problem
with constraints
[increasing complexity]

constraints only

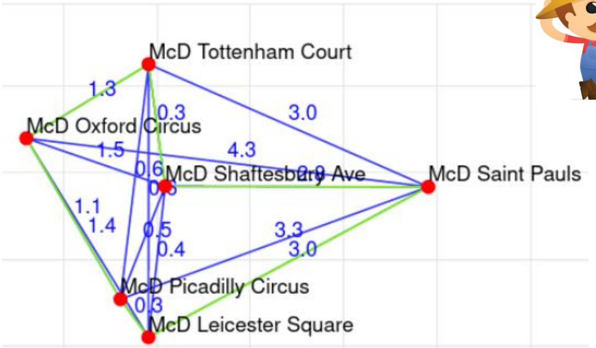
constraints only

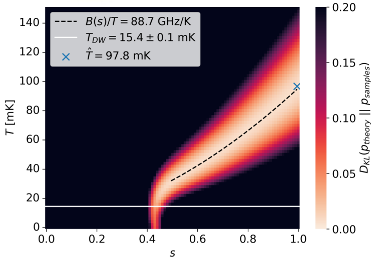
optimization only [basically]

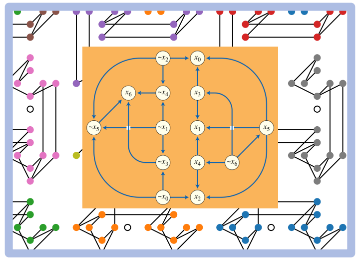
sampling problem

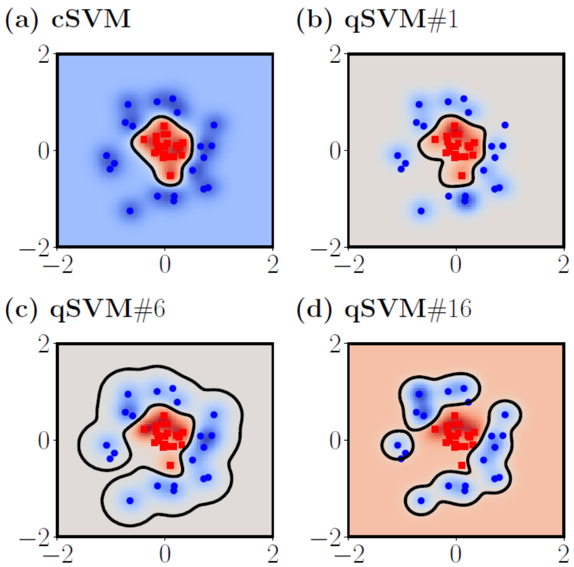
JUNIQ Rolling Call:

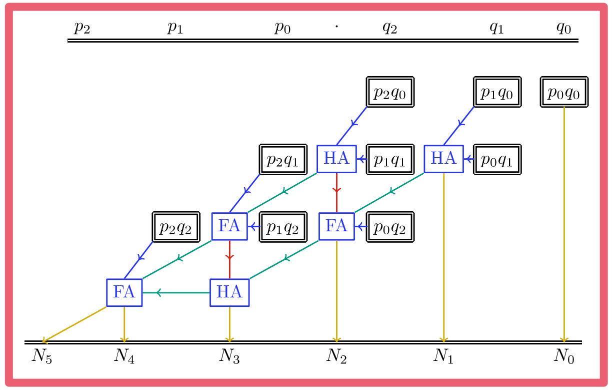
or search for
"FZJ JUNIQ ACCESS"

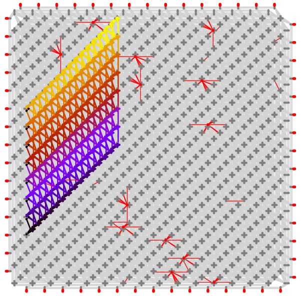










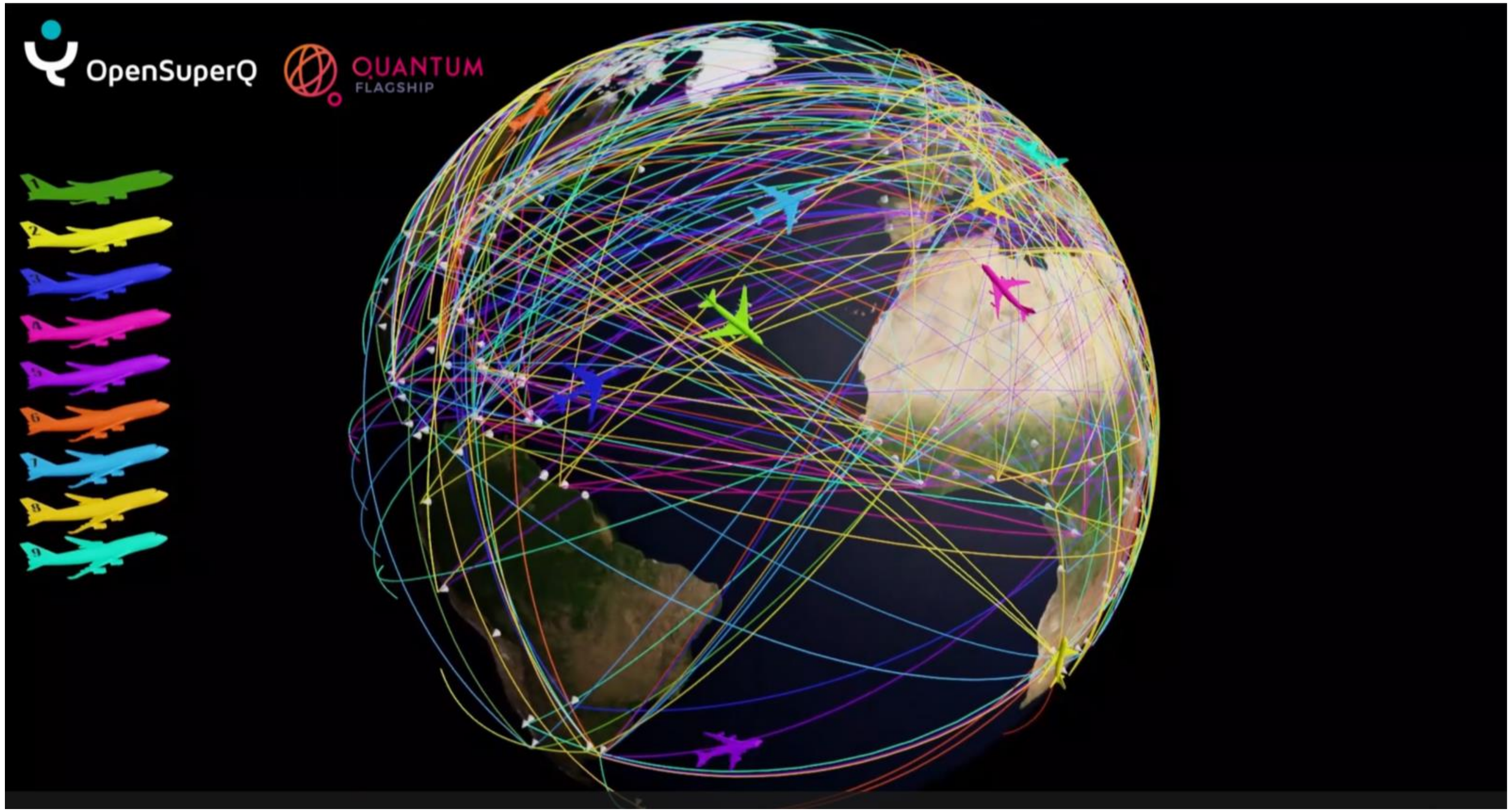


Willsch et al. (2022)
QINP **21**, 141
arXiv:2105.02208

TAIL ASSIGNMENT PROBLEM

Application: Airline scheduling

Find optimal flight schedule such that each flight is covered exactly once



<u>In collaboration with:</u>
Marika Svensson
Jeppesen and
Chalmers University of Technology

BOEING

OpenSuperQ QUANTUM FLAGSHIP

Willsch et al. (2022)
QINP **21**, 141
arXiv:2105.02208

TAIL ASSIGNMENT PROBLEM

Problem specification

Find optimal flight schedule such that each flight is covered exactly once

	Flight 0	Flight 1	Flight 2	Flight 3	Flight 4	Flight 5	⋯	Flight 469	Flight 470	Flight 471
Route 0	0	1	0	0	0	0		0	0	0
Route 1	0	0	0	0	1	0		0	0	0
⋮										
Route 10	0	0	0	0	0	1		0	0	1
Route 11	0	0	1	0	0	0		0	0	0
Route 12	0	0	0	1	0	0		1	0	0
Route 13	0	0	1	0	0	0		0	0	0
Route 14	0	0	0	1	0	0		0	0	0
Route 15	0	0	0	0	1	0		0	0	0
Route 16	0	1	0	0	0	0		0	1	0
⋮										
Route 37	0	0	0	0	0	1		0	0	0
Route 38	0	0	0	1	0	0		0	0	0
Route 39	0	0	0	1	0	0		0	0	0

Willsch et al. (2022)
QINP **21**, 141
arXiv:2105.02208

TAIL ASSIGNMENT PROBLEM

Mathematical formulation

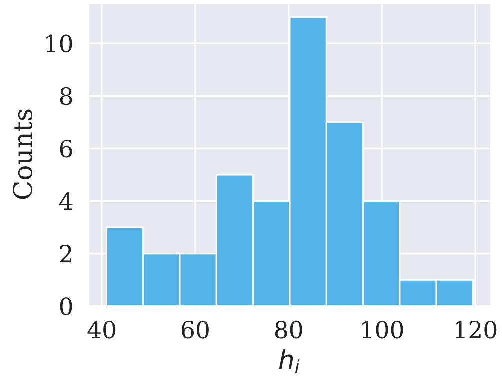
	Flight 0	Flight 1	Flight 2	Flight 3	Flight 4	Flight 5	⋯	Flight 469	Flight 470	Flight 471
Route 0	0	1	0	0	0	0		0	0	0
Route 1	0	0	0	0	1	0		0	0	0
⋮										
Route 10	0	0	0	0	0	1		0	0	1
Route 11	0	0	1	0	0	0		0	0	0
Route 12	0	0	0	1	0	0		1	0	0
Route 13	0	0	1	0	0	0		0	0	0
Route 14	0	0	0	1	0	0		0	0	0
Route 15	0	0	0	0	1	0		0	0	0
Route 16	0	1	0	0	0	0		0	1	0
⋮										
Route 37	0	0	0	0	0	1		0	0	0
Route 38	0	0	0	1	0	0		0	0	0
Route 39	0	0	0	1	0	0		0	0	0

- Linear assignment problem

$$\begin{array}{ll} \text{minimize} & \vec{f}^T\vec{x} \\ \text{subject to} & A\vec{x} = \vec{b} \end{array}$$

encodes cost of assigning airplanes to routes

exact cover problem: each flight covered once

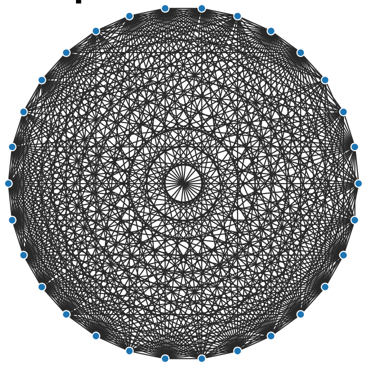


- Reformulation as Ising problem

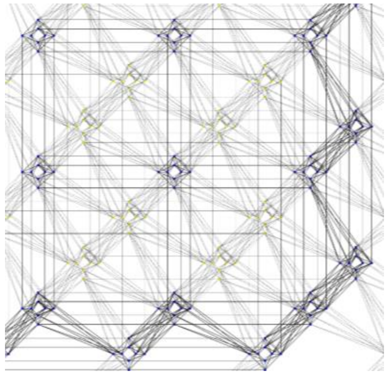
$$\text{QUBO} \leftrightarrow \text{Ising}: \quad x_i = (1+s_i)/2$$

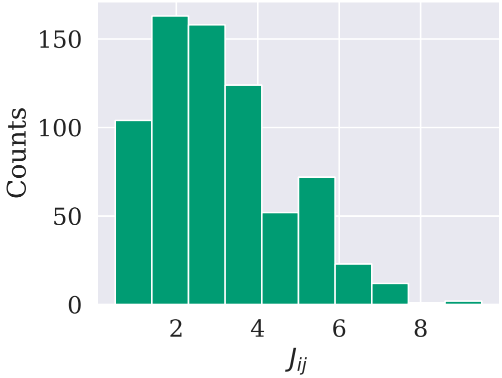
$$\min_{x_i=0,1}\left(\left(A\vec{x}-\vec{b}\right)^2 + \lambda\vec{f}^T\vec{x}\right) = \min_{s_i=\pm1}\left(\sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j + \text{const}\right)$$

- 25 - 40 qubit almost fully-connected (“clique”) Ising problems



embedding needed





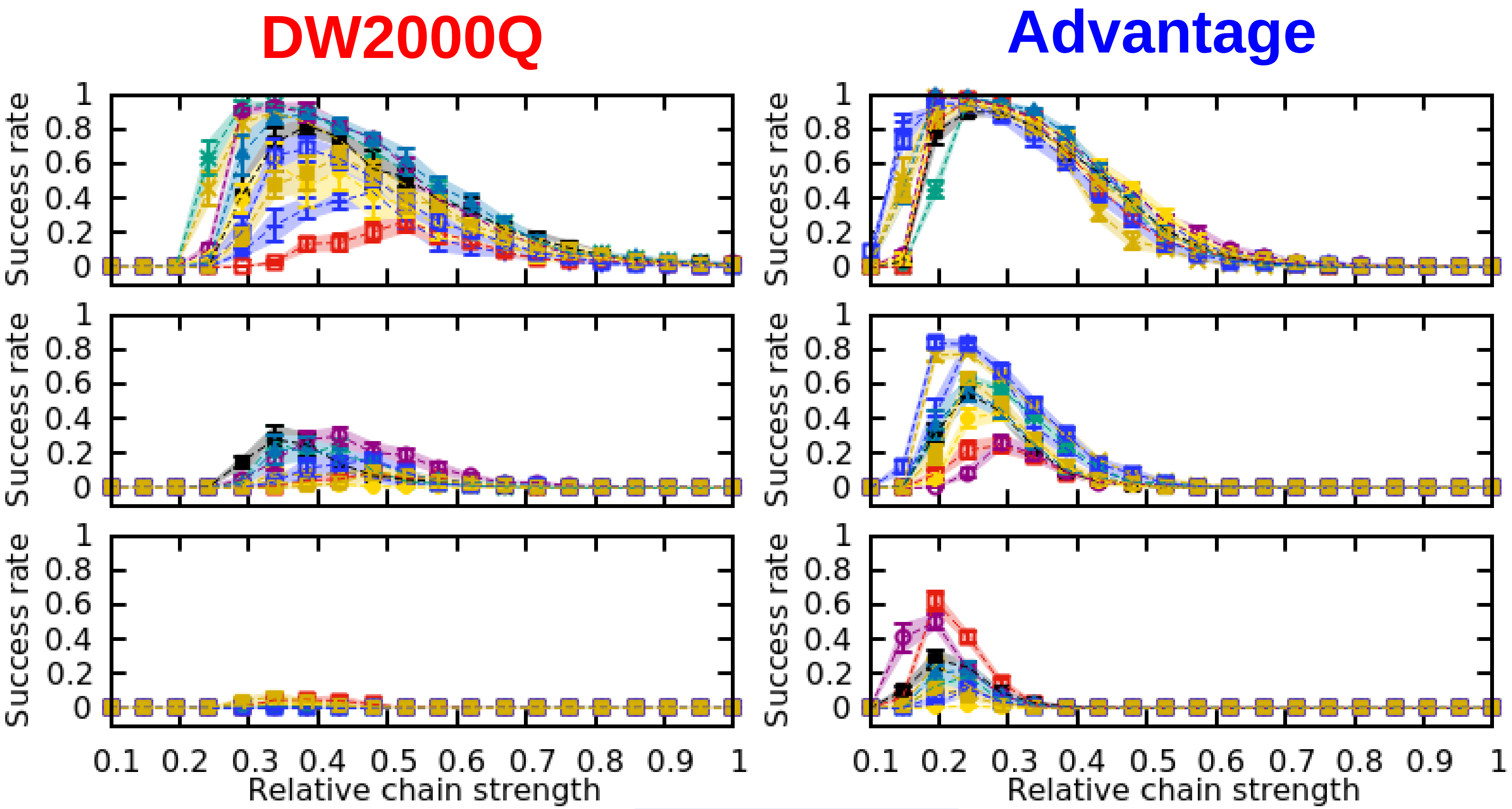
JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

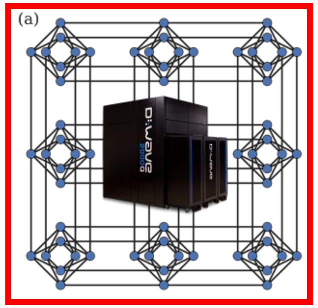
Willsch et al. (2022)
QINP **21**, 141
arXiv:2105.02208

TAIL ASSIGNMENT: RESULTS

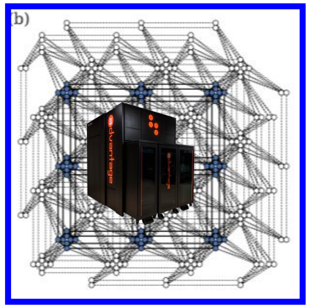
30 - 40 qubit problems (90% nonzero couplers)

Scan of 10 different embeddings and 20 relative chain strengths:





vs.



Relative chain strength
= CS / max(|hi|, |Jij|)

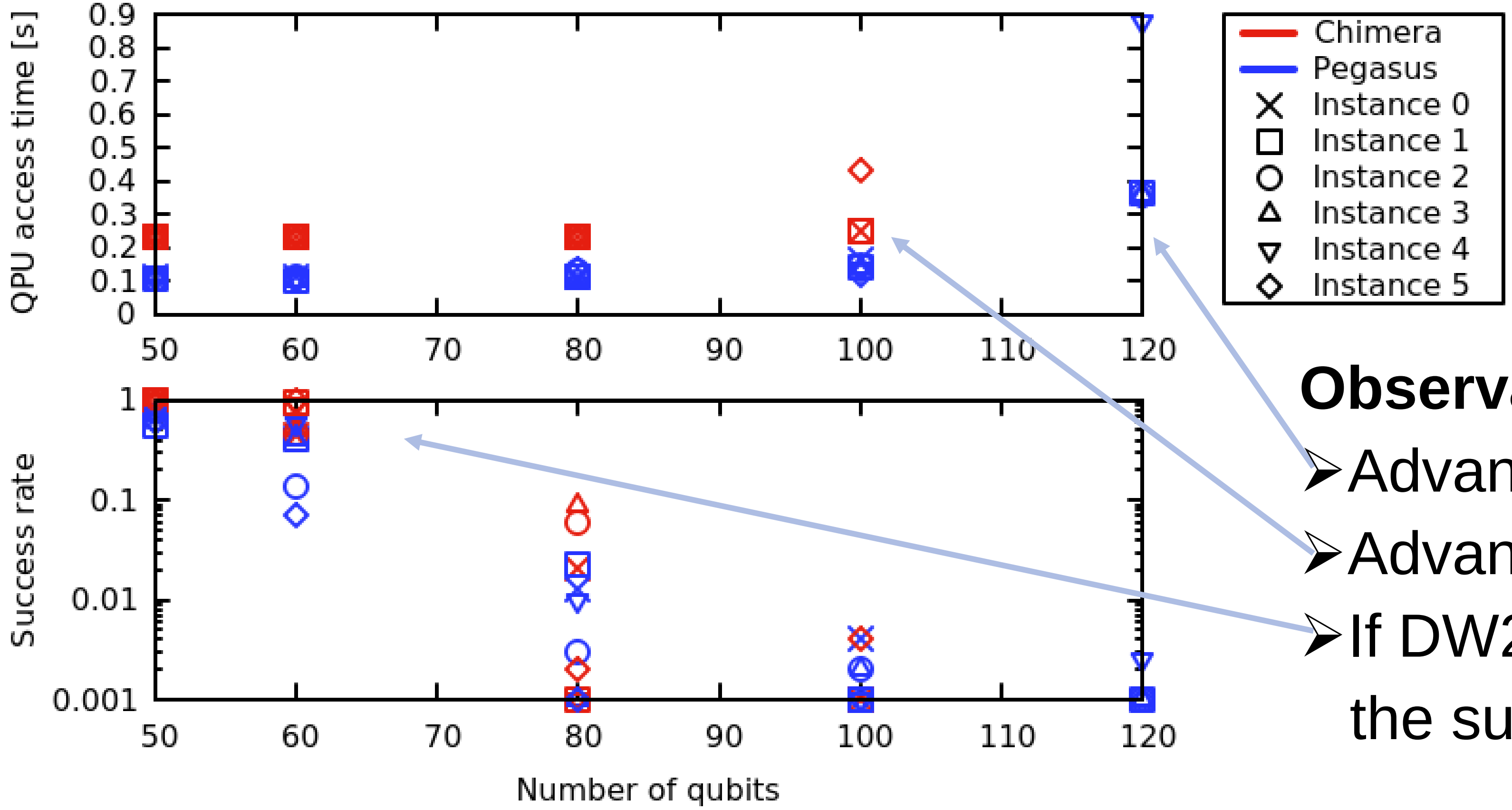
JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

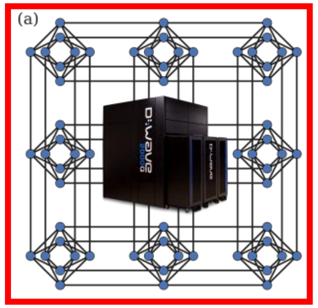
Willsch et al. (2022)
QINP **21**, 141
arXiv:2105.02208

TAIL ASSIGNMENT: RESULTS

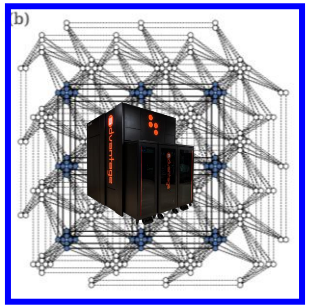
50-120 qubits: larger but sparser problems (20% nonzero couplers)

The fastest successful runs that reproducibly gave a solution:





vs.



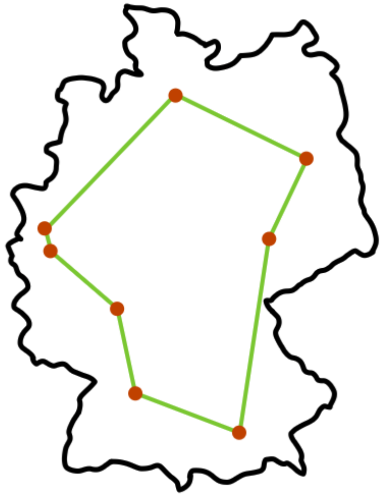
Observations:

- Advantage solves larger problems
- Advantage solves problems faster
- If DW2000Q can solve a problem, the success rate is sometimes higher

Nemes et al., https://tsp-quantum.netlify.app/ (2022)
Montanez-Barrera et al., QST 9, 025022 (2024)
Tsakalias, JUSER:1038364 (2024)

TRAVELING SALESMAN PROBLEM

QUBO formulation



$$\min_{x_k=0,1} \vec{x}^T Q \vec{x}$$

Cost to travel from city i to city j

$$\min_{x_{it}=0,1} \left\{ \lambda \sum_{i=0}^{n-1} \sum_{\substack{j=0 \\ j\neq i}}^{n-1} \sum_{t=0}^{n-1} c_{ij} x_{it} x_{j(t+1)} + \sum_{i=0}^{n-1} \left(\sum_{t=0}^{n-1} x_{it} - 1 \right)^2 + \sum_{t=0}^{n-1} \left(\sum_{i=0}^{n-1} x_{it} - 1 \right)^2 \right\}$$

time steps

cities

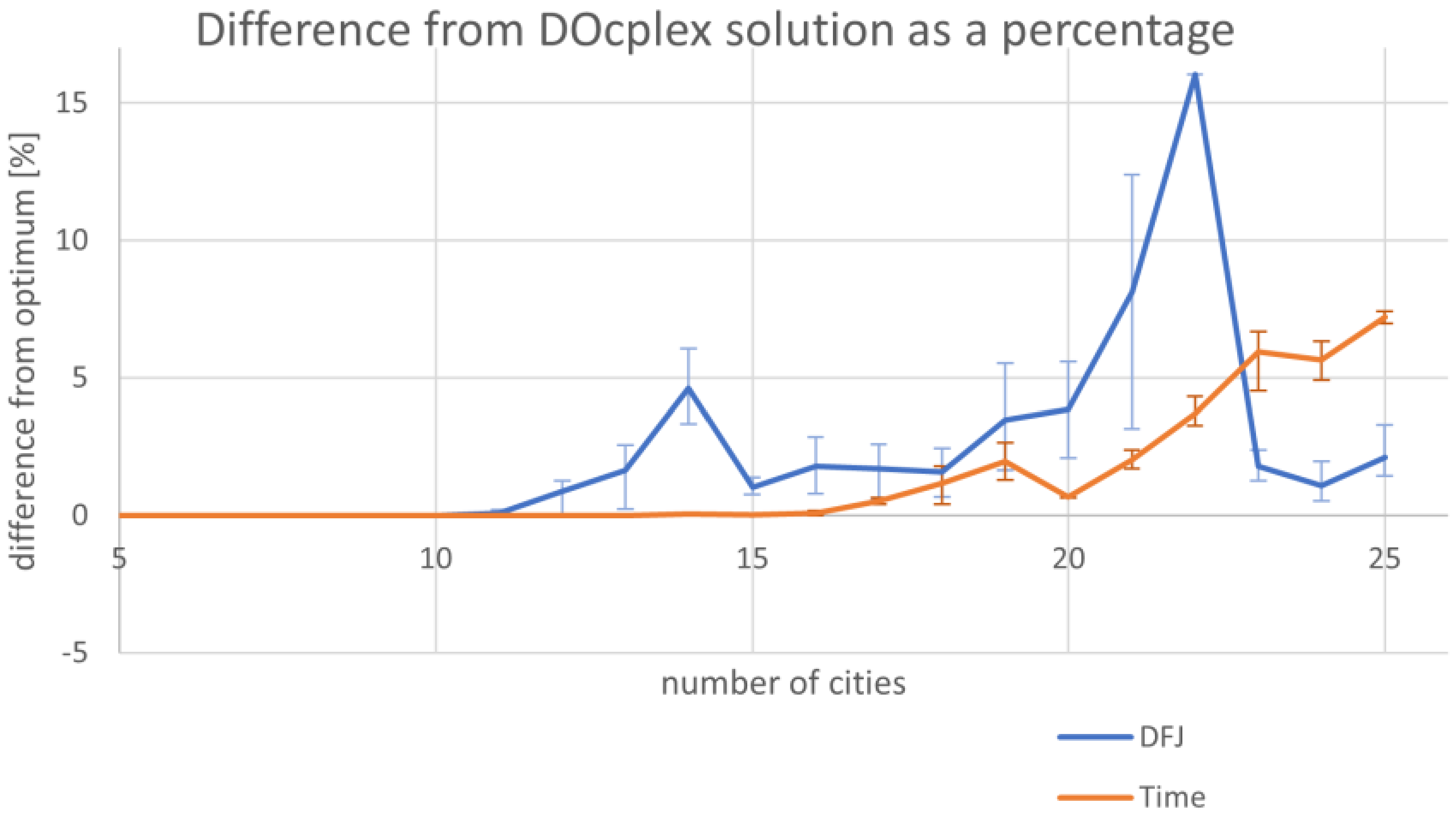
	0	1	2	3
A	1	0	0	0
B	0	0	0	1
C	0	1	0	0
D	0	0	1	0

- Assign cities & times to qubits:
  - Qubit $x_{it} = 1$ means the traveler is at city i at time t
  - Count the cost c_{ij} if the traveler goes from i to j at some time step
  - Traveler must pass city i only once
  - Traveler can only be at one city at each time step
- Could simplify by fixing starting point → Number of qubits $(n-1)^2$
- Not the DFJ formulation = linear but exp. many inequality constraints

Nemes et al., https://tsp-quantum.netlify.app/ (2022)
Montanez-Barrera et al., QST 9, 025022 (2024)
Tsakalias, DOI:10.34734/FZJ-2025-01361 (2024)

TRAVELING SALESMAN PROBLEM

Results: 25 cities in Europe

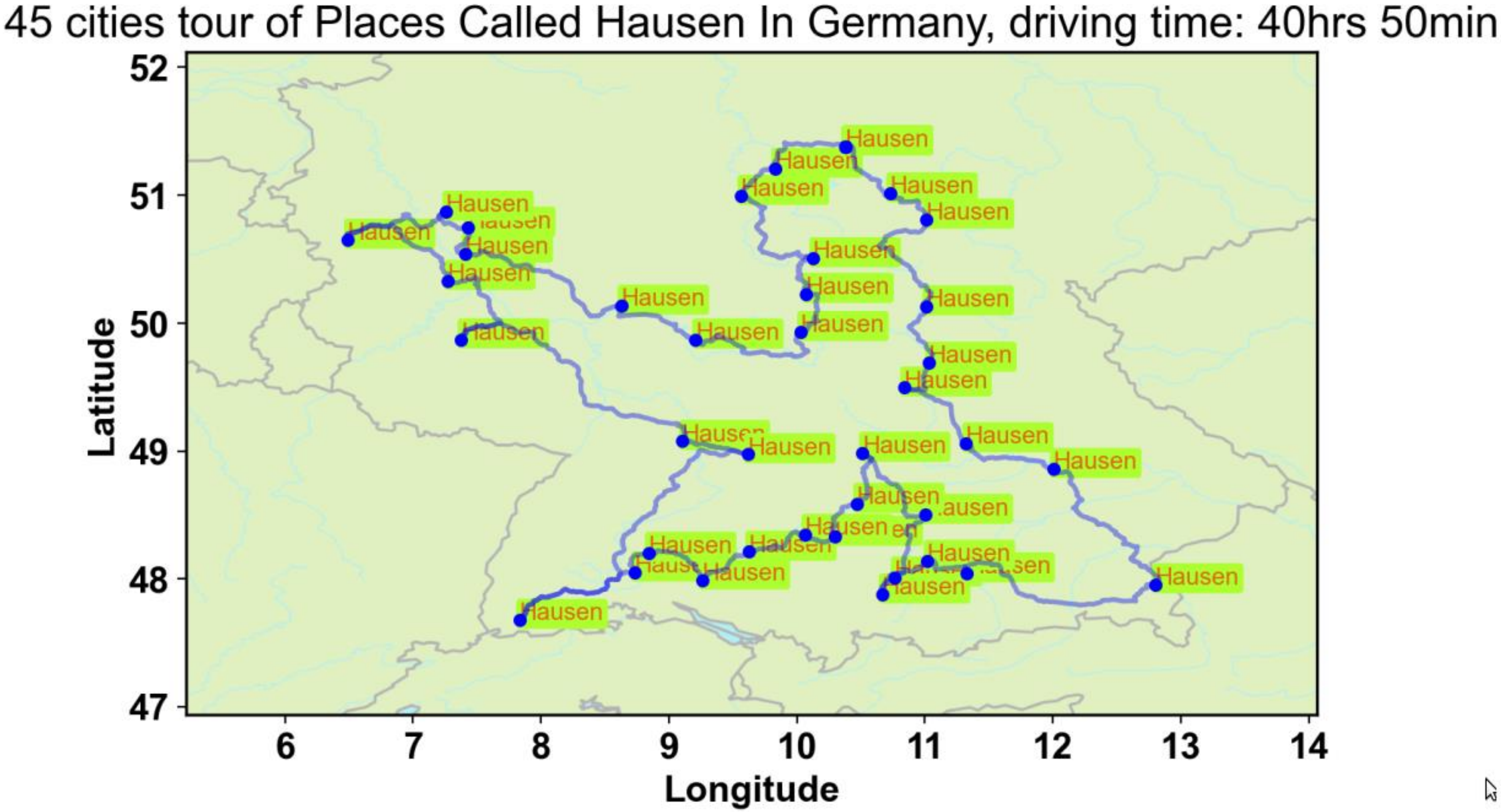


➢ Additional result: D-Wave performance comparable to human performance!

Nemes et al., https://tsp-quantum.netlify.app/ (2022)
Montanez-Barrera et al., QST 9, 025022 (2024)
Tsakalias, DOI:10.34734/FZJ-2025-01361 (2024)

TRAVELING SALESMAN PROBLEM

More results: 45 cities in Germany

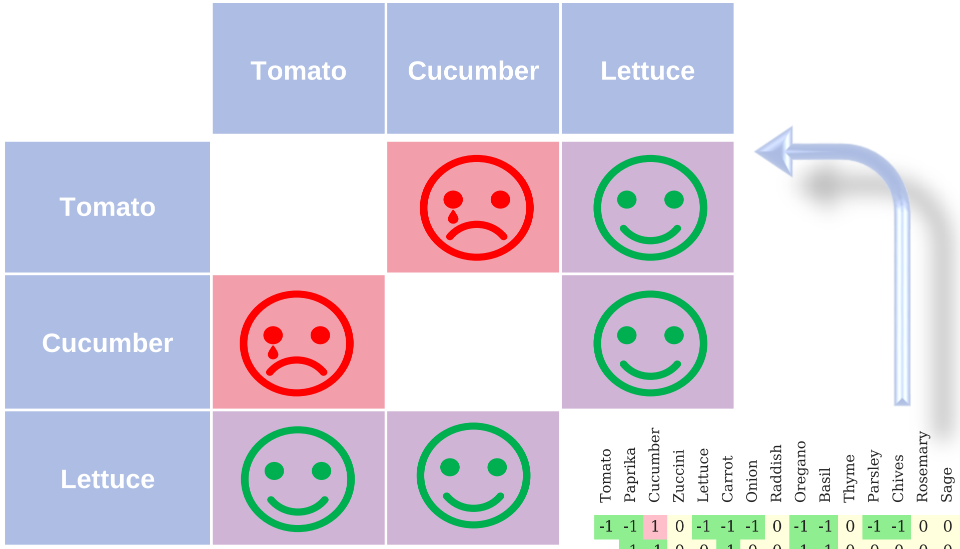


VEGETABLE GARDEN OPTIMIZATION

Gonzalez Calaza et al. (2021)
QINP **20**, 305
arXiv:2101.10827
https://jugit.fz-juelich.de/qip/garden-optimization-problem

Overview

- Problem: Companion planting in polyculture vegetable gardens



Tomato	Paprika	Cucumber	Zuccini	Lettuce	Carrot	Onion	Raddish	Oregano	Basil	Thyme	Parsley	Chives	Rosemary	Sage	Dill	Coriander	Mint	
-1	-1	1	0	-1	-1	-1	0	-1	-1	0	-1	-1	0	0	0	-1	-1	Tomato
	-1	-1	0	0	-1	0	0	-1	-1	0	0	0	0	0	0	0	0	Paprika
		-1	1	-1	0	-1	-1	0	0	0	-1	0	0	0	-1	0	0	Cucumber
			-1	0	0	-1	0	0	0	0	-1	0	0	0	0	0	0	Zuccini
				-1	-1	-1	-1	0	0	0	1	0	0	-1	-1	-1	-1	Lettuce
					-1	-1	-1	0	0	0	0	-1	-1	-1	-1	0	-1	Carrot
						-1	0	0	0	0	-1	0	0	0	-1	0	0	Onion
							-1	0	0	0	-1	0	0	0	0	0	0	Raddish
								-1	-1	0	0	0	0	0	0	0	0	Oregano
									-1	0	0	0	0	0	0	-1	0	Basil
										-1	0	0	0	0	0	0	0	Thyme
											-1	-1	0	0	0	-1	0	Parsley
												-1	0	0	-1	1	0	Chives
													-1	0	0	0	0	Rosemary
														-1	0	0	0	Sage
															-1	-1	0	Dill
																-1	0	Coriander
																	-1	Mint

- Task: Find an optimal placement of plants in the garden considering the characteristics of their nearest neighbours

Gonzalez Calaza et al. (2021)
QINP **20**, 305
arXiv:2101.10827
https://jugit.fz-juelich.de/qip/garden-optimization-problem

VEGETABLE GARDEN OPTIMIZATION

QUBO formulation

$$\min_{x_k=0,1} \vec{x}^T Q \vec{x}$$

Tomato	Paprika	Cucumber	Zuccini	Lettuce	Carrot	Onion	Raddish	Oregano	Basil	Thyme	Parsley	Chives	Rosemary	Sage	Dill	Coriander	Mint	
-1	-1	1	0	-1	-1	-1	0	-1	-1	0	-1	-1	0	0	0	-1	-1	Tomato
	-1	-1	0	0	-1	0	0	-1	-1	0	0	0	0	0	0	0	0	Paprika
		-1	1	-1	0	-1	-1	0	0	0	-1	0	0	0	-1	0	0	Cucumber
			-1	0	0	-1	0	0	0	0	-1	0	0	0	0	0	0	Zuccini
				-1	-1	-1	-1	0	0	0	1	0	0	-1	-1	-1	-1	Lettuce
					-1	-1	-1	0	0	0	0	-1	-1	-1	-1	0	-1	Carrot
						-1	0	0	0	0	-1	0	0	0	-1	0	0	Onion
							-1	0	0	0	-1	0	0	0	0	0	0	Raddish
								-1	-1	0	0	0	0	0	0	0	0	Oregano
									-1	0	0	0	0	0	0	-1	0	Basil
										-1	0	0	0	0	0	0	0	Thyme
											-1	-1	0	0	0	-1	0	Parsley
												-1	0	0	-1	1	0	Chives
													-1	0	0	0	0	Rosemary
														-1	0	0	0	Sage
															-1	-1	0	Dill
																-1	0	Coriander
																	-1	Mint

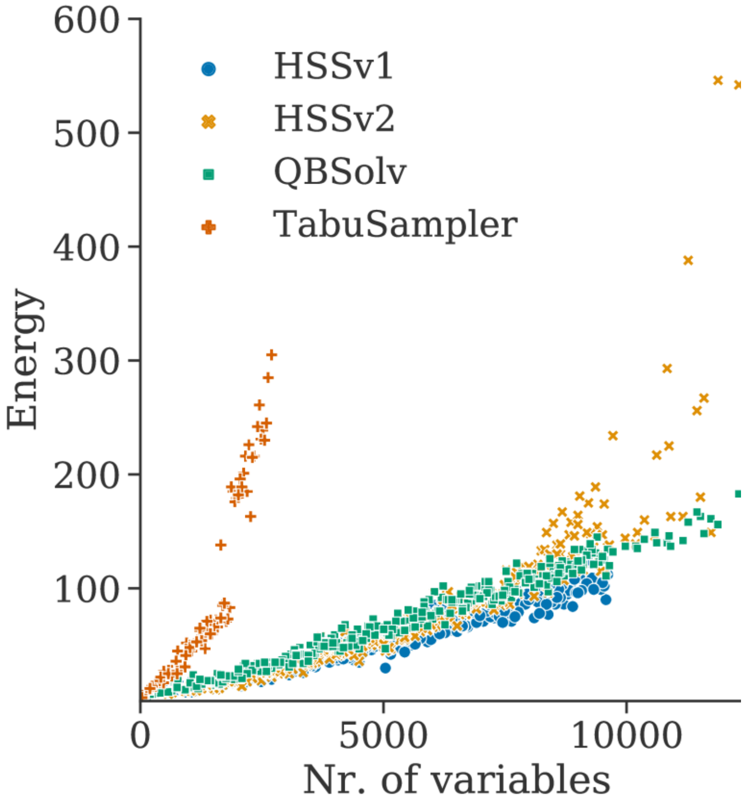
- Assign plants to pots in the garden:
  - Qubit $x_{ij} = 1$ means species j is placed in pot i
  - All plants should have a good relationship with their neighbors
  - Use all available plants and pots
  - Big plants should not shadow small ones

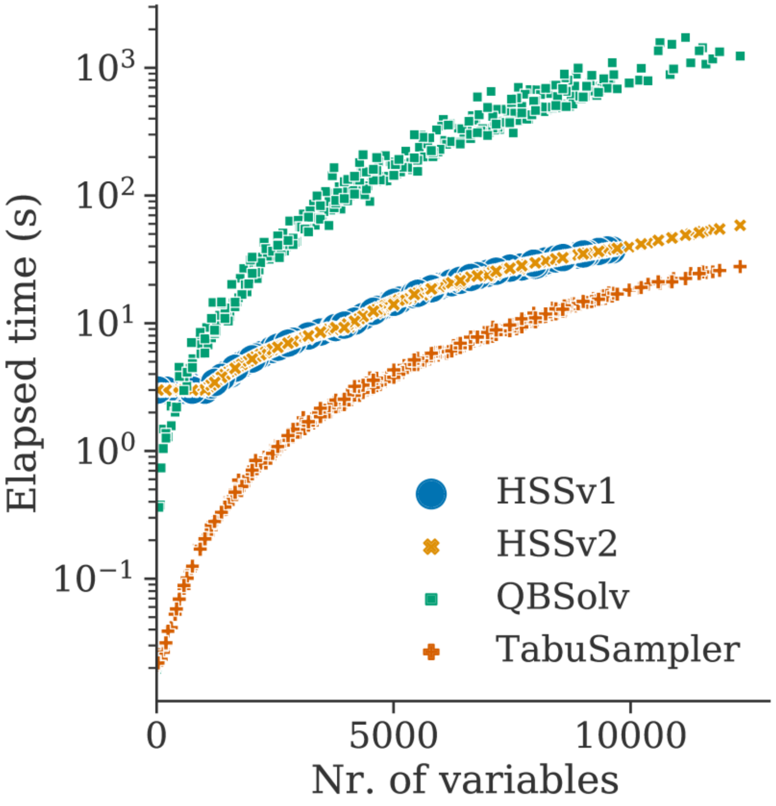
n: nr. of pots in the garden
t: nr. of plant species to place
J: connectivity between pots
C: relationship between species
c_j: nr. of plants of species ***j***
s_j: size of species ***j***

Gonzalez Calaza et al. (2021)
QINP **20**, 305
arXiv:2101.10827
https://jugit.fz-juelich.de/qip/garden-optimization-problem

CLASSICAL AND HYBRID SOLVERS

Comparing energies and run times





Results:

- HSSv1: best but only up to 10k vars.
- HSSv2: same as v1 up to 2k vars., worse later.
- QBSolv: good but very slow.
- TabuSampler: returns unusable results extremely fast.

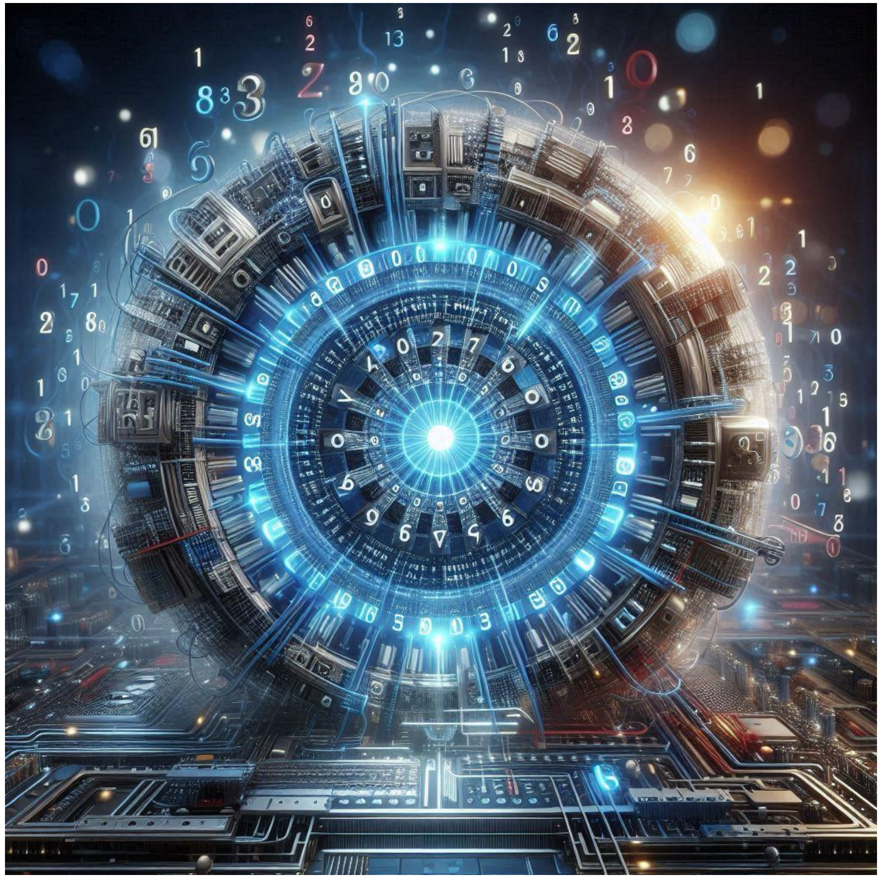
Conclusion:

- Hybrid solvers outperform classical solvers.

Willsch et al., Mathematics 11, 4222 (2023)
P. Hanussek, DOI:10.34734/FZJ-2024-05254 (2024)
Willsch et al., arXiv:2410.14397 (2024)

THE FACTORING PROBLEM

$$N = p \times q$$



FACTORING RECORDS

Willsch et al., Mathematics 11, 4222 (2023)
P. Hanussek, DOI:10.34734/FZJ-2024-05254 (2024)
Willsch et al., arXiv:2410.14397 (2024)

Digital QCs

Shor's Algorithm
$N = 15, 21, (35) \sim 2^{6}$
Martín-López et al., Nat. Photonics 6, 773 (2012)
Monz et al., Science 351, 1068 (2016)
Amico et al., Phys. Rev. A 100, 012305 (2019)

Careful: Oversimplified Shor's Algorithm
$N \sim 2^{20000}$
Smolin et al., Nature 499, 7457 (2013)
Gidney, algassert.com/post/2000 (2020)

Variational Quantum Factoring
$N = 1099551473989 \sim 2^{41}$
Anschuetz et al., arXiv:1808.08927 (2018)
Karamlou et al., npj Quantum Inf. 7, 156 (2021)

← Careful: preprocessed down to 3 qubits!

Schnorr's Algorithm
$N = 261980999226229 \sim 2^{48}$
Schnorr, cryptoeprint:2021/933 (2021)
Hegade and Solano, arXiv:2301.1105 (2023)

← seems not scalable? →
Ducas, github.com/lducas/SchnorrGate (2021)
Grebnev et al., IEEE Access 11, 134760 (2023)
Khattar and Yosri, arXiv:2307.09651 (2023)
Tesoro et al., arXiv:2410.16355 (2024)

Analog QCs

Direct Method
$N = 1042441 \sim 2^{20}$
Peng et al., Phys. Rev. Lett. 101, 220405 (2008)
Willsch et al., arXiv:2410.14397 (2024)

Multiplier Circuit
$N = 8219999 \sim 2^{23}$
Andriyash et al., Tech. Rep. 14-1002A-B (2016)
Jiang et al., Sci. Rep. 8, 17667 (2018)
Ding et al., Sci. Rep. 14, 3518 (2024)

Schnorr's Algorithm
$N = 845546611823483 \sim 2^{50}$
Schnorr, cryptoeprint:2021/933 (2021)
Yan et al., arXiv:2212.12372 (2022)
Wang et al., Chin. J. Comput. 47, 13015 (2024)

Simulated Digital QCs

Shor's Algorithm
$N = 549755813701 \sim 2^{39}$
Willsch et al., Mathematics 11, 4222 (2023)

Classical (Super)Computers

Number Field Sieve
$N = \text{RSA-250} \sim 10^{250} \sim 2^{829}$
Boudot et al., CRYPTO 2020, 62 (2020)

Schnorr's Algorithm
$N \sim 2^{100}$
Tesoro et al., arXiv:2410.16355 (2024)

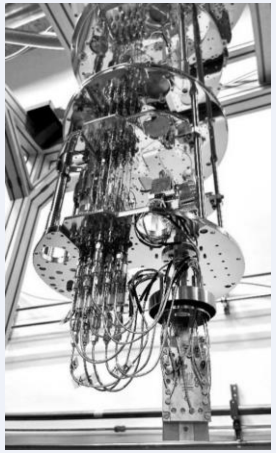
Willsch et al., Mathematics 11, 4222 (2023)
P. Hanussek, DOI:10.34734/FZJ-2024-05254 (2024)
Willsch et al., arXiv:2410.14397 (2024)

FACTORING RECORDS

Digital QCs

Shor's Algorithm
N = 15, 21, (35) ~ 2^{6}
Martín-López et al., Nat. Photonics 6, 773 (2012)
Monz et al., Science 351, 1068 (2016)
Amico et al., Phys. Rev. A 100, 012305 (2019)

Careful: Oversimplified Shor's Algorithm
N ~ 2^{20000}
Smolin et al., Nature 499, 7457 (2013)
Gidney, algassert.com/post/2000 (2020)



Variational Quantum Factoring
N = 1099551473989 ~ 2^{41}
Anschuetz et al., arXiv:1808.08927 (2018)
Karamlou et al., npj Quantum Inf. 7, 156 (2021)

← Careful: preprocessed down to 3 qubits!

Schnorr's Algorithm
N = 261980999226229 ~ 2^{48}
Schnorr, cryptoeprint:2021/933 (2021)
Hegade and Solano, arXiv:2301.1105 (2023)

← seems not scalable? →
Ducas, github.com/lducas/SchnorrGate (2021)
Grebnev et al., IEEE Access 11, 134760 (2023)
Khattar and Yosri, arXiv:2307.09651 (2023)
Tesoro et al., arXiv:2410.16355 (2024)

Analog QCs

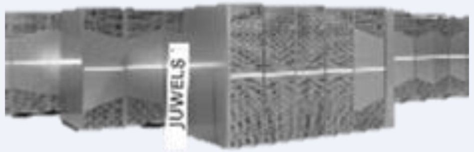
Direct Method
N = 1042441 ~ 2^{20}
Peng et al., Phys. Rev. Lett. 101, 220405 (2008)
Willsch et al., arXiv:2410.14397 (2024)

Multiplier Circuit
N = 8219999 ~ 2^{23}
Andriyash et al., Tech. Rep. 14-1002A-B (2016)
Jiang et al., Sci. Rep. 8, 17667 (2018)
Ding et al., Sci. Rep. 14, 3518 (2024)

Schnorr's Algorithm
N = 845546611823483 ~ 2^{50}
Schnorr, cryptoeprint:2021/933 (2021)
Yan et al., arXiv:2212.12372 (2022)
Wang et al., Chin. J. Comput. 47, 13015 (2024)

Simulated Digital QCs

Shor's Algorithm
N = 549755813701 ~ 2^{39}
Willsch et al., Mathematics 11, 4222 (2023)



Classical (Super)Computers

Number Field Sieve
N = RSA-250 ~ 10^{250} ~ 2^{829}
Boudot et al., CRYPTO 2020, 62 (2020)

Schnorr's Algorithm
N ~ 2^{100}
Tesoro et al., arXiv:2410.16355 (2024)

Willsch et al., Mathematics 11, 4222 (2023)
P. Hanussek, DOI:10.34734/FZJ-2024-05254 (2024)
Willsch et al., arXiv:2410.14397 (2024)

FACTORING ON ANALOG QCS

Mehtods

- ➢ Problem needs to be expressed as QUBO / Ising
- ➢ We investigated **3 approaches**:

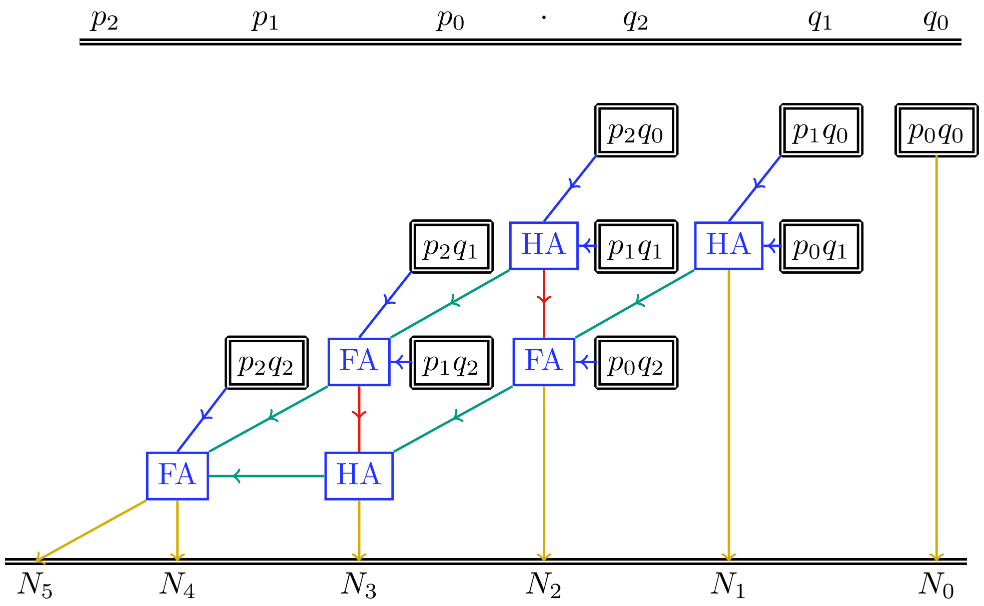
Direct Method

Peng et al., Phys. Rev. Lett. 101, 220405 (2008)

$$\min_{p,q} (N - pq)^2$$

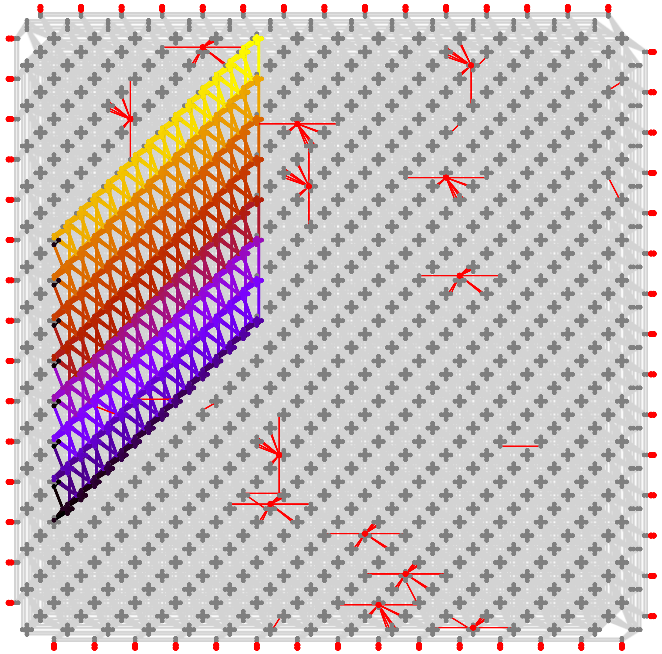
Multiplication Circuit Method

Andriyash et al, D-Wave TR 14-1002A-B (2016)



Controlled Full-Adder Method

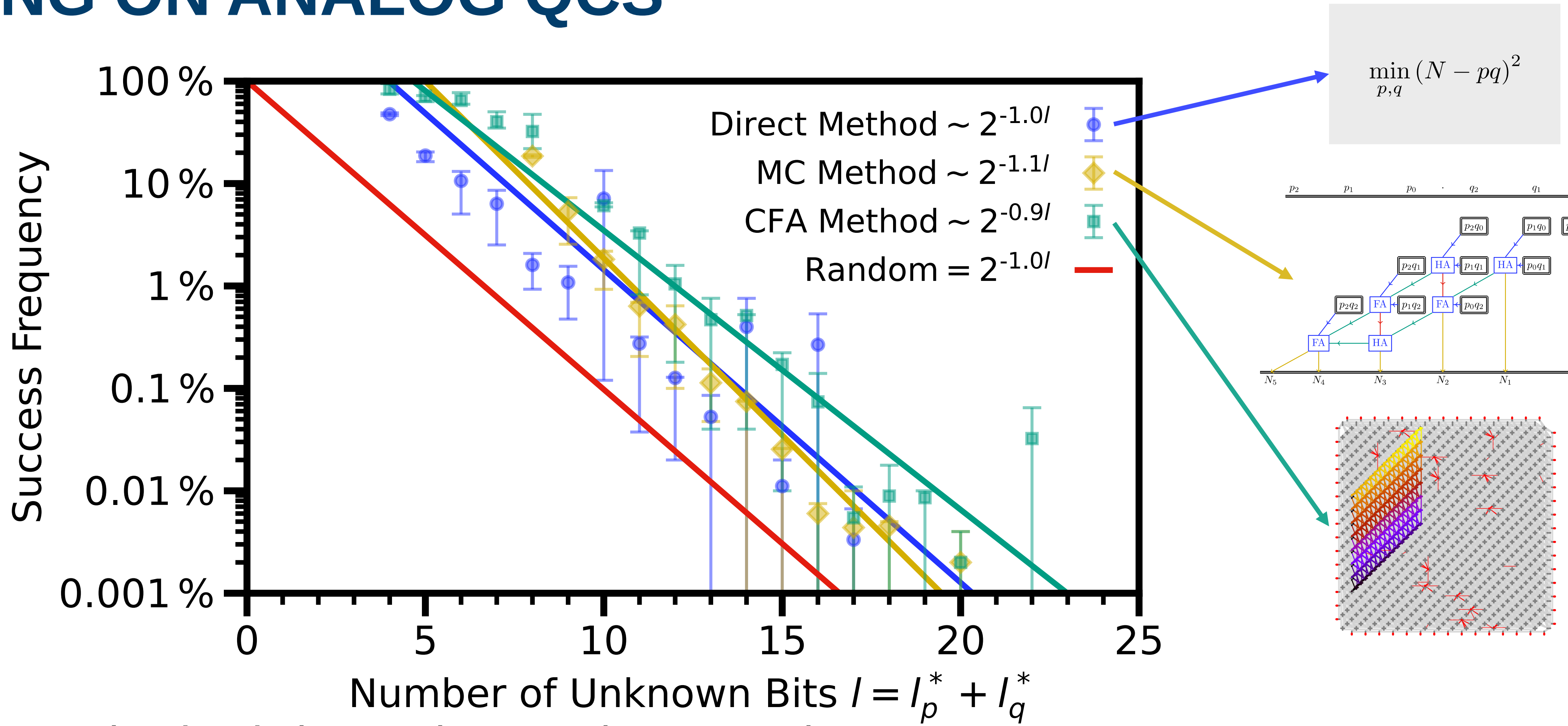
Ding et al., Sci. Rep. 14, 1 (2024)



JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

FACTORING ON ANALOG QCS

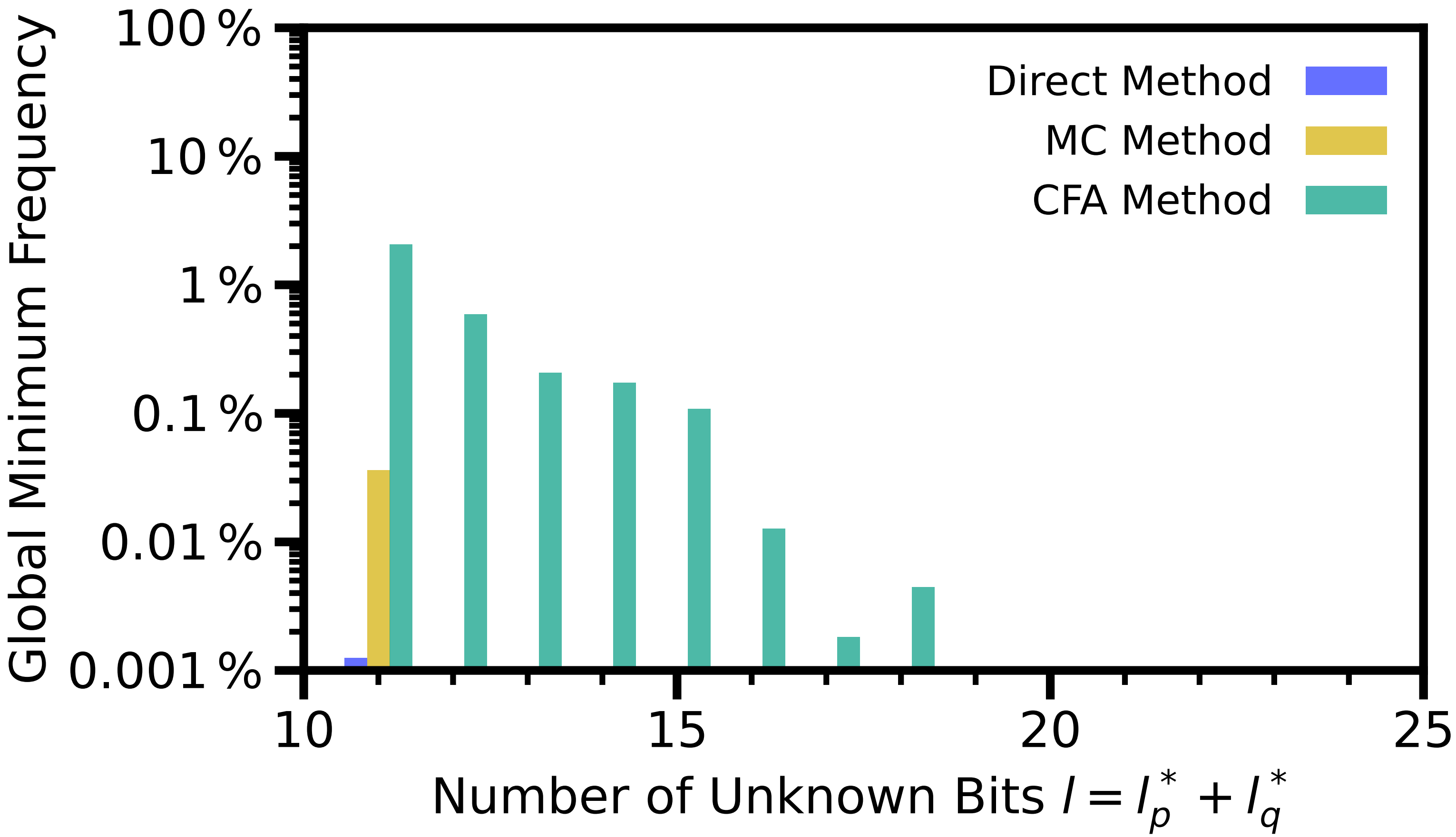
Results



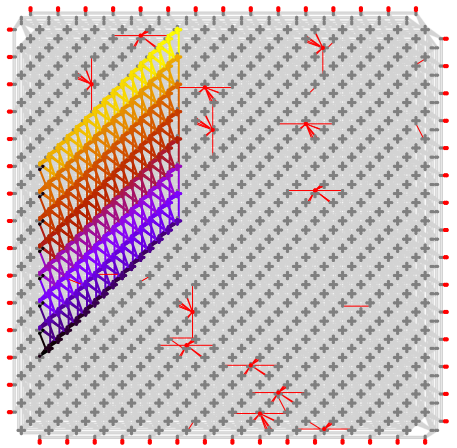
- All methods are absolutely better than random guessing
- CFA method also scales better
- **BUT:** All methods scale exponentially!

FACTORING ON ANALOG QCS

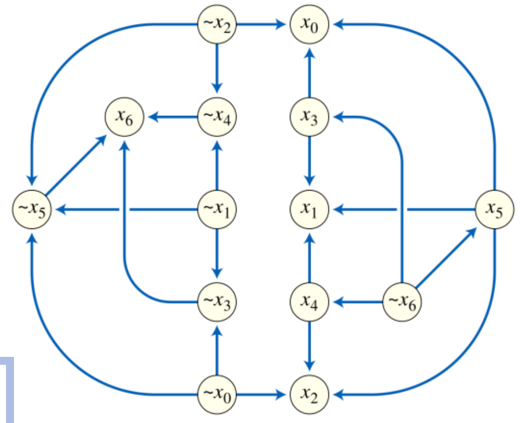
Results



- ➢ CFA method finds true ground state much more often!
- ➢ **Reason:** Custom embedding on the hardware



2-SATISFIABILITY



Mehta et al., PRA **105**, 062406 (2022)
Mehta et al., PRA **104**, 032421 (2021)

Overview

- Mathematical formulation

$$F = (L_{1,1} \vee L_{1,2}) \wedge (L_{2,1} \vee L_{2,2}) \wedge .... \wedge (L_{M,1} \vee L_{M,2})$$

where $L_{j,k} = x_i$ or $\overline{x_i}$ with $x_i = 0, 1$

→ find assignment to x_i that makes F true

Ising: $\min\limits_{s_i=\pm 1} \left(\sum\limits_i h_i s_i + \sum\limits_{i<j} J_{ij} s_i s_j \right)$

- Reformulation as Ising Problem

$$H_{2SAT} = \sum_{\alpha=1}^{M} h_{2SAT}(\epsilon_{\alpha,1} s_{i[\alpha,1]}, \epsilon_{\alpha,2} s_{i[\alpha,2]})$$

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

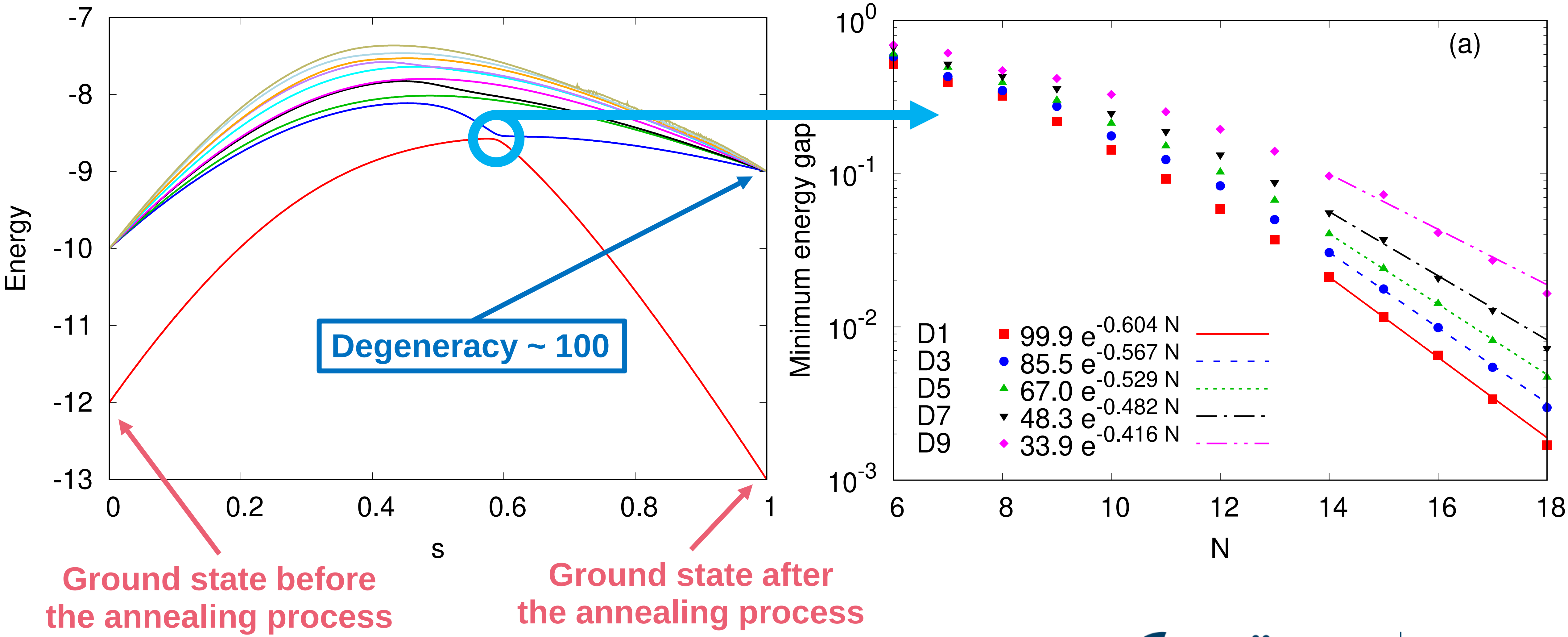
$\epsilon_\alpha = 1$ for x_i
$\epsilon_\alpha = -1$ for $\overline{x_i}$

  - Example for clause $x_1 \vee x_2$: $h_{2SAT} = s_1 s_2 - (s_1 + s_2) + 1$

- “Hard 2-SAT problems”: The purpose is benchmarking
  - Chosen to be “hard” for quantum annealers (not for digital computers)

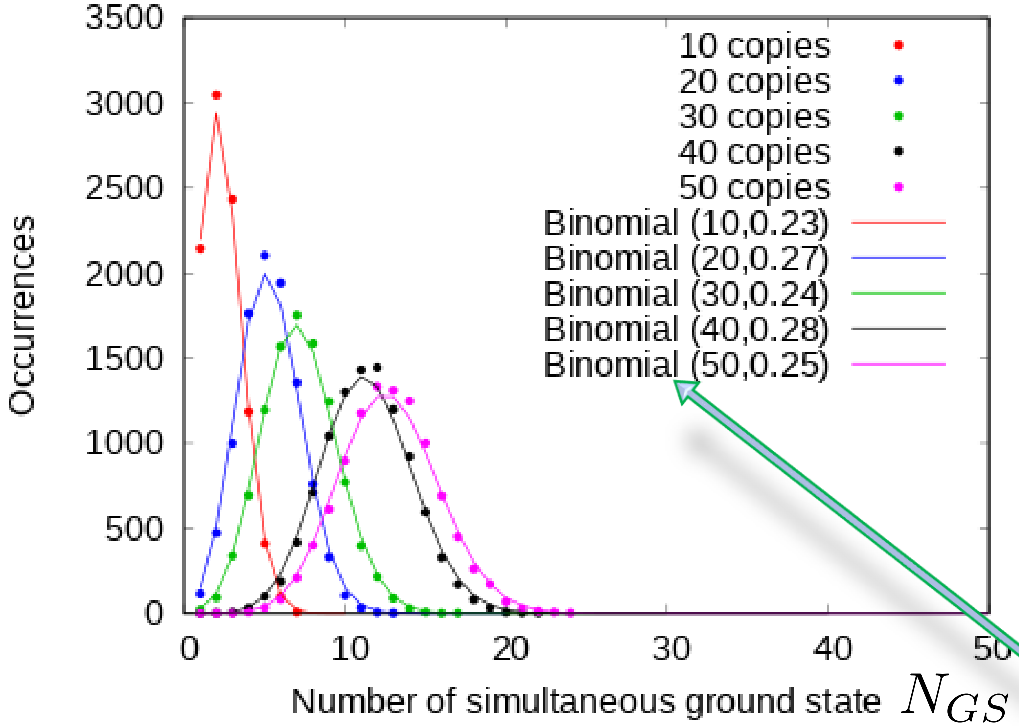
2-SATISFIABILITY

Properties: unique ground state, highly degenerate first excited level, small gaps



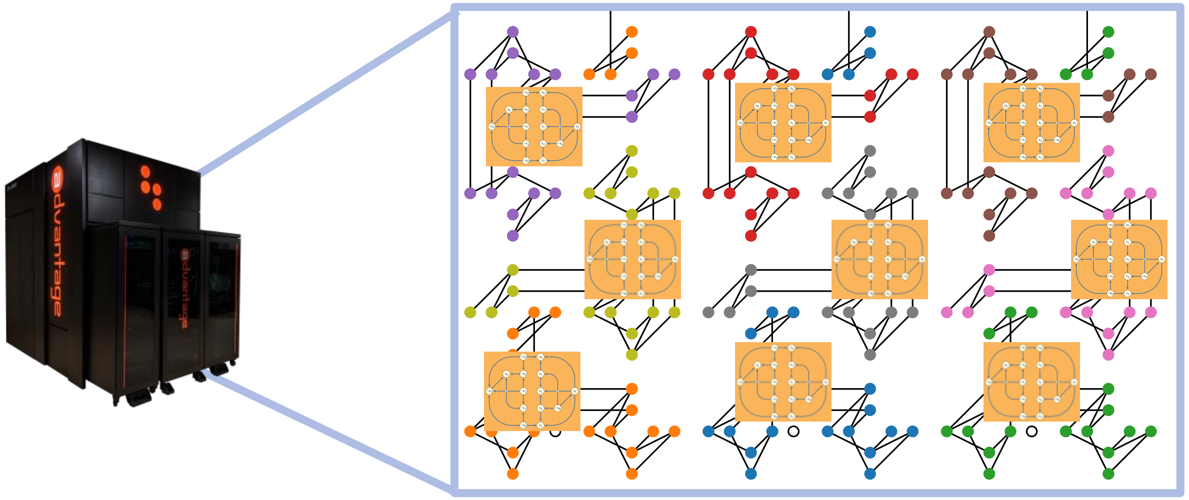
2-SATISFIABILITY: RESULTS

Embed the same problem multiple times



$$\binom{N_{\text{copies}}}{N_{GS}} p^{N_{GS}} (1-p)^{N_{\text{copies}}-N_{GS}}$$

Success rate: probability to find the unique solution (GS)



- The occurrence of optimal solutions follows a binomial distribution
  - In no run could all ground states be found simultaneously
  - Different parts of the QPU work almost independently

Willsch et al., CPC 248, 107006 (2020)
Cavallaro et al., IGARSS 2020, 1973 (2020)
Delilbasic et al., IGARSS 2021, 2608 (2021)
Pasetto et al., LGRS 19, 1505705 (2022)
Delilbasic et al., JSTARS 17, 1434 (2023)
Pasetto et al., JSTARS 17, 3263 (2024)
Zardini et al., TQE 5, 3103512 (2024)

QUANTUM SUPPORT VECTOR MACHINES

From classical SVM to quantum SVM

- An SVM is a supervised machine-learning method for binary classification
- Given a training set

Feature vector — Label

$$D = \{(\mathbf{d}_n, t_n) : n = 0, \ldots, N-1\}$$

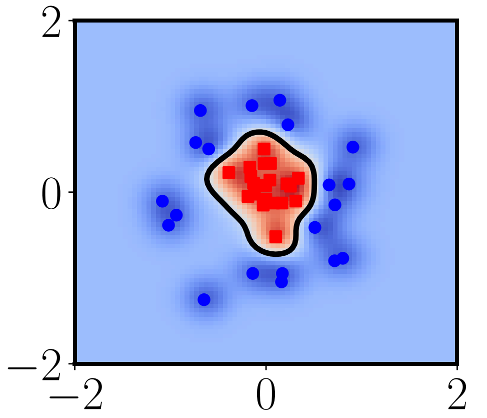
an SVM is trained by solving the Quadratic Programming problem

$$\text{minimize} \qquad E(\{\alpha_n\}) = \frac{1}{2}\sum_{nm}\alpha_n\alpha_m t_n t_m k(\mathbf{d}_n, \mathbf{d}_m) - \sum_n \alpha_n$$

$$\text{subject to} \qquad 0 \le \alpha_n \le C \quad \text{and} \quad \sum_n \alpha_n t_n = 0$$

Continuous problem variables

Kernel function (nonlinear SVM)



- QUBO problems are also quadratic, but for binary variables

Number of qubits per problem variable

→ Use encoding: $\alpha_n = \sum_{k=0}^{K-1} B^{k-P} x_{[n,k]}$ → $E(\{\alpha_n\})$ → QUBO formulation

Base & Exponent

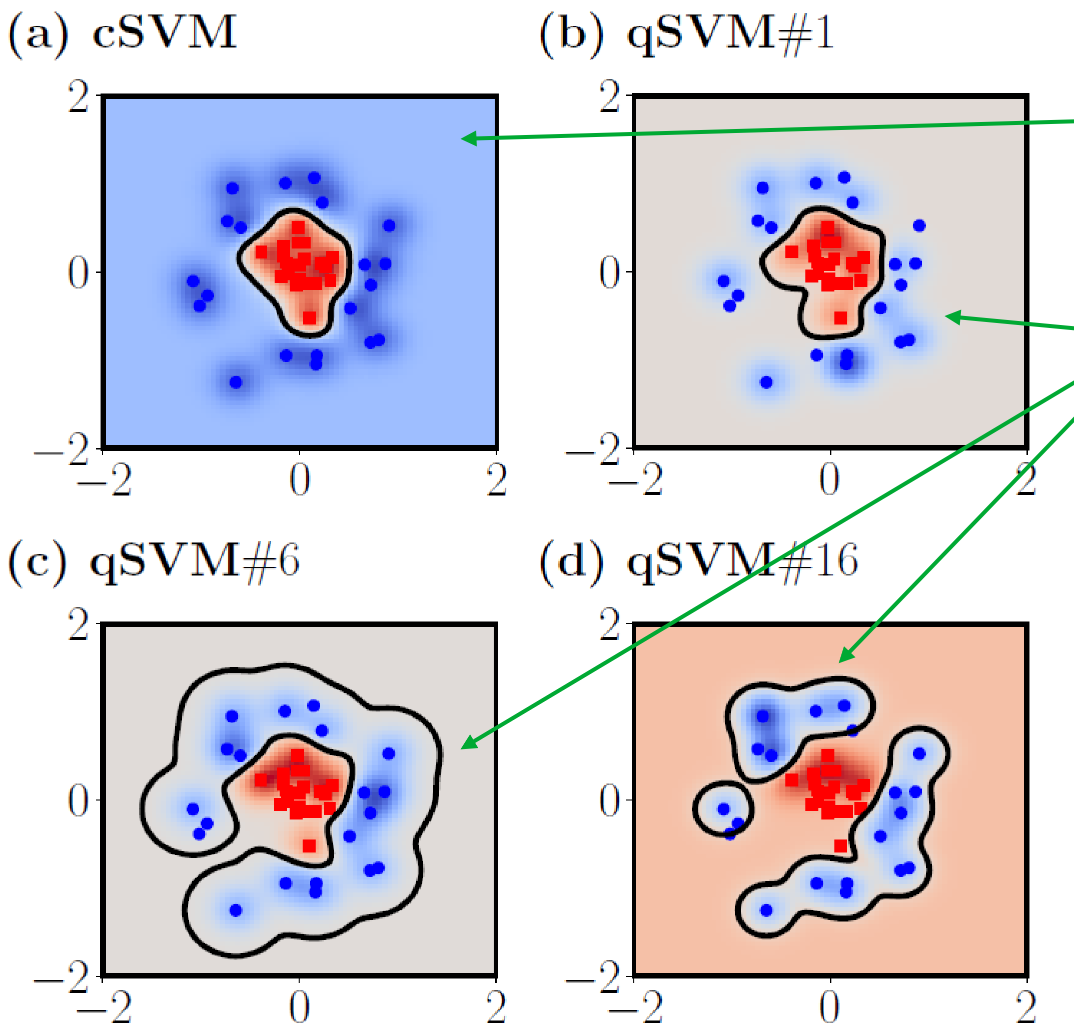
$$\min_{x_i=0,1}\left(\sum_{i,j} x_i Q_{ij} x_j\right)$$

JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

Willsch et al., CPC 248, 107006 (2020)
Cavallaro et al., IGARSS 2020, 1973 (2020)
Delilbasic et al., IGARSS 2021, 2608 (2021)
Pasetto et al., LGRS 19, 1505705 (2022)
Delilbasic et al., JSTARS 17, 1434 (2023)
Pasetto et al., JSTARS 17, 3263 (2024)
Zardini et al., TQE 5, 3103512 (2024)

QUANTUM SUPPORT VECTOR MACHINES

Results



- Classical SVM (**cSVM**) yields the global minimum, but only for the training data
- Quantum SVM (**qSVM**) yields additional low-energy classifiers from ensemble of solutions

Combination of multiple low-energy qSVM classifiers generalizes better to unseen data

QUANTUM SUPPORT VECTOR MACHINES

Willsch et al., CPC 248, 107006 (2020)
Cavallaro et al., IGARSS 2020, 1973 (2020)
Delilbasic et al., IGARSS 2021, 2608 (2021)
Pasetto et al., GRSL 1505705 (2023)
Delilbasic et al., JSTARS 17, 1434 (2023)
Pasetto et al., JSTARS 17, 3263 (2024)

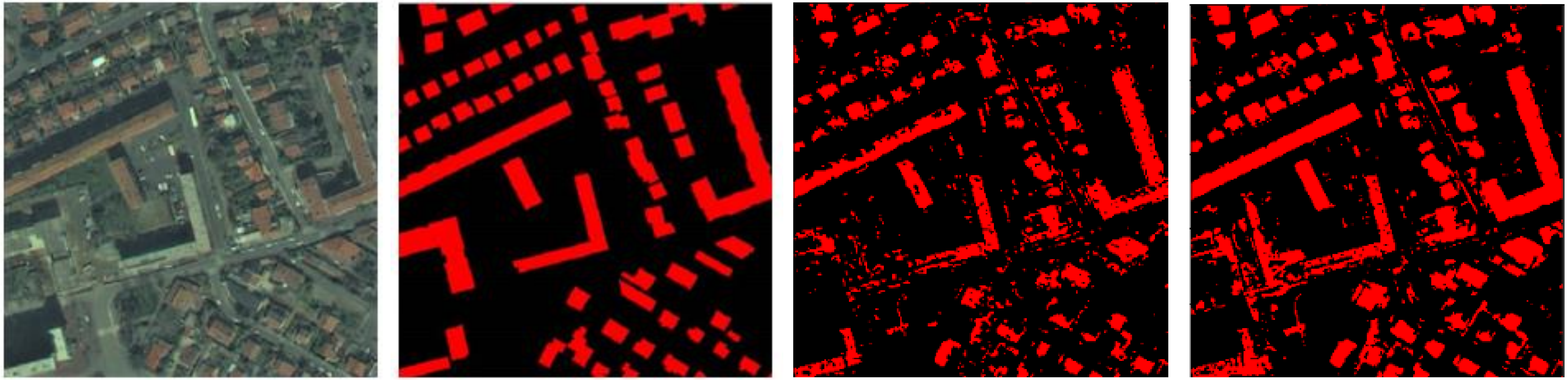
Application to remote sensing

QUANTUM SUPPORT VECTOR MACHINES

Willsch et al., CPC 248, 107006 (2020)
Cavallaro et al., IGARSS 2020, 1973 (2020)
Delilbasic et al., IGARSS 2021, 2608 (2021)
Pasetto et al., LGRS 19, 1505705 (2022)
Delilbasic et al., JSTARS 17, 1434 (2023)
Pasetto et al., JSTARS 17, 3263 (2024)
Zardini et al., TQE 5, 3103512 (2024)

Application to remote sensing: Results

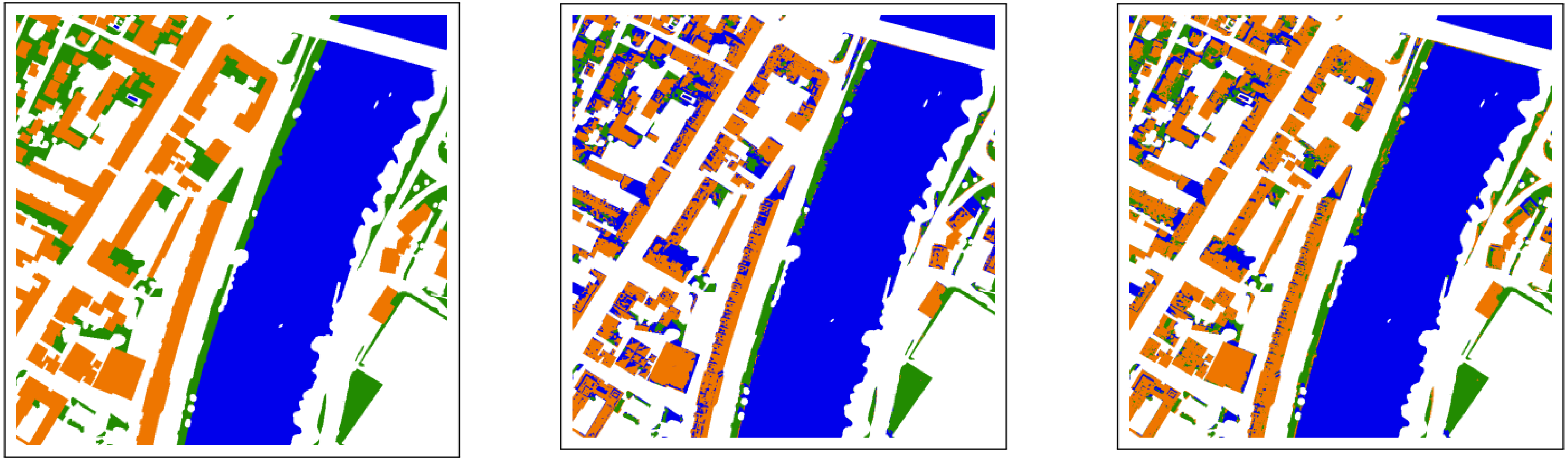
Original | Ground truth | Classical SVM | Quantum SVM



QUANTUM SUPPORT VECTOR MACHINES

Willsch et al., CPC 248, 107006 (2020)
Cavallaro et al., IGARSS 2020, 1973 (2020)
Delilbasic et al., IGARSS 2021, 2608 (2021)
Pasetto et al., LGRS 19, 1505705 (2022)
Delilbasic et al., JSTARS 17, 1434 (2023)
Pasetto et al., JSTARS 17, 3263 (2024)
Zardini et al., TQE 5, 3103512 (2024)

Extension to multiple classes



Ground truth　　Classical SVM　　Quantum SVM

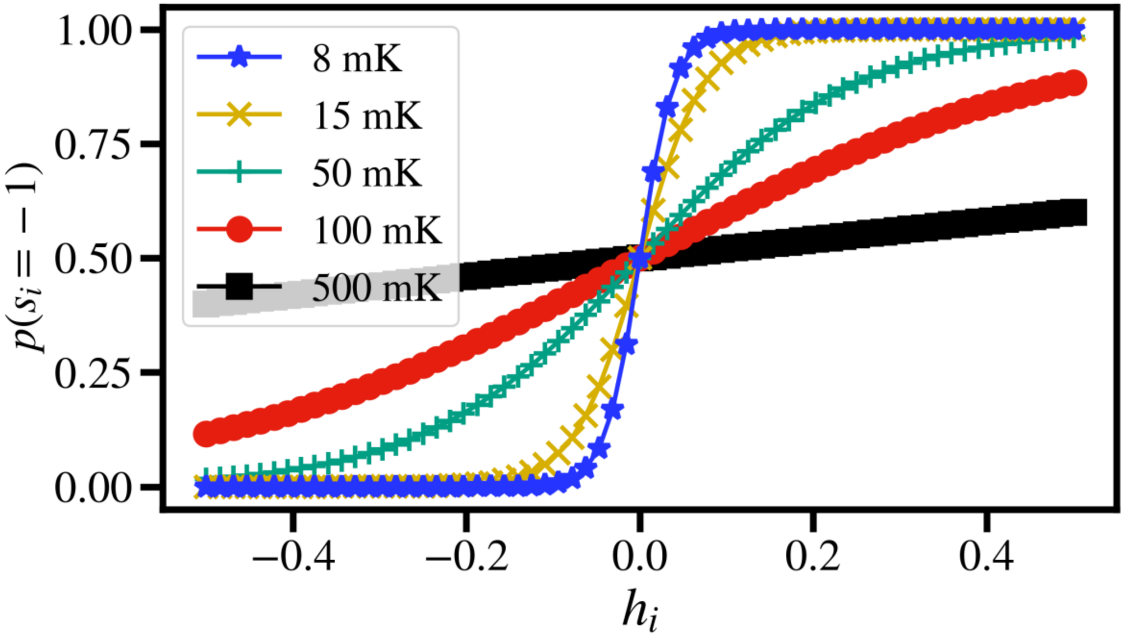
Perot et al. (2022)
https://jugit.fz-juelich.de/qip/qbm-quant-finance

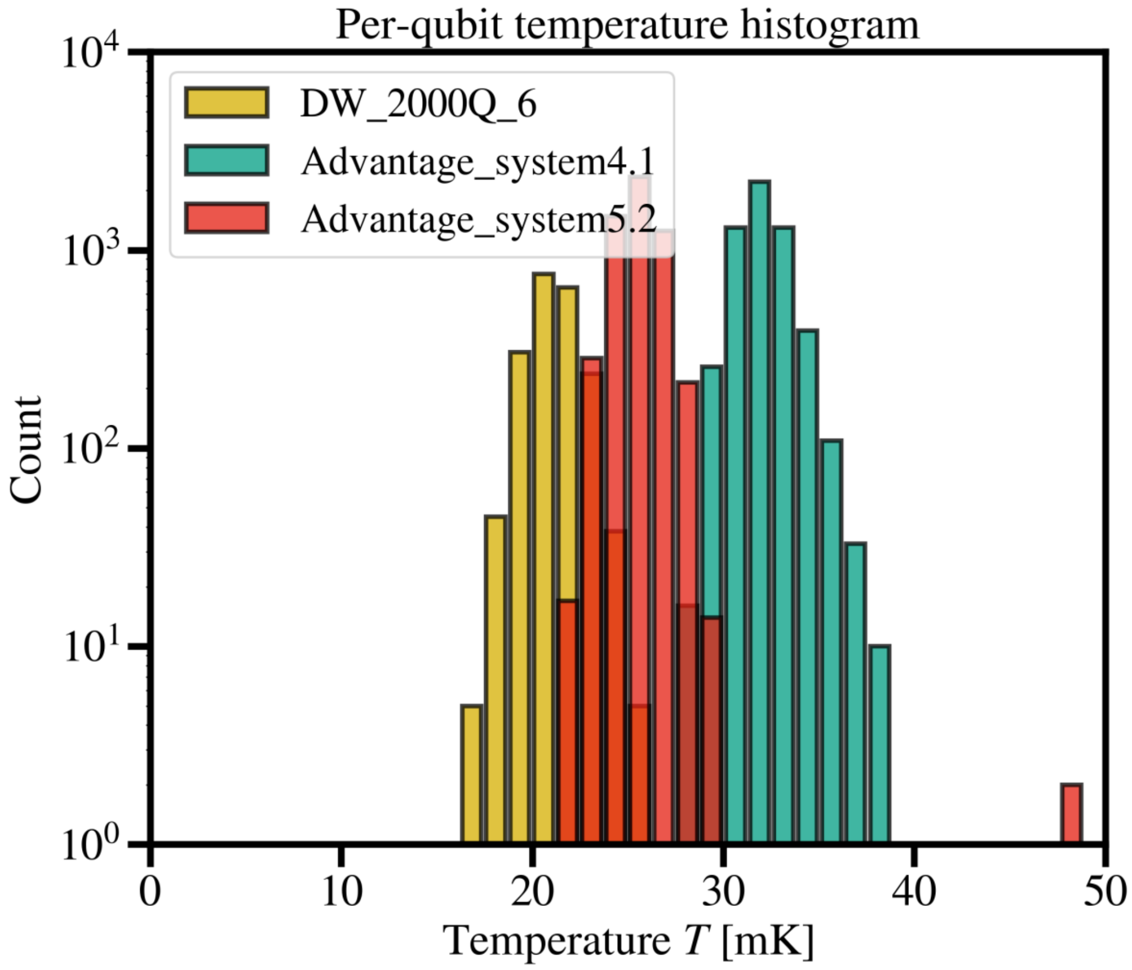
QUANTUM BOLTZMANN MACHINES

Experiment #1: Effective per-qubit temperature distribution

$$H(s) = \frac{A(s)}{2}\left(\sum_i \sigma_i^x\right) + \frac{B(s)}{2}\left(\sum_i \boxed{h_i \sigma_i^z} + \sum_{i<j} \cancel{J_{ij}} \sigma_i^z \sigma_j^z\right)$$

$$\boxed{p(s_i = \pm 1) = \langle s_i | \frac{1}{Z} e^{-\beta H(1)} | s_i \rangle = \frac{1}{2}\left(1 + \tanh\left(-\frac{B(1) h_i}{2 k_B T} s_i\right)\right)}$$



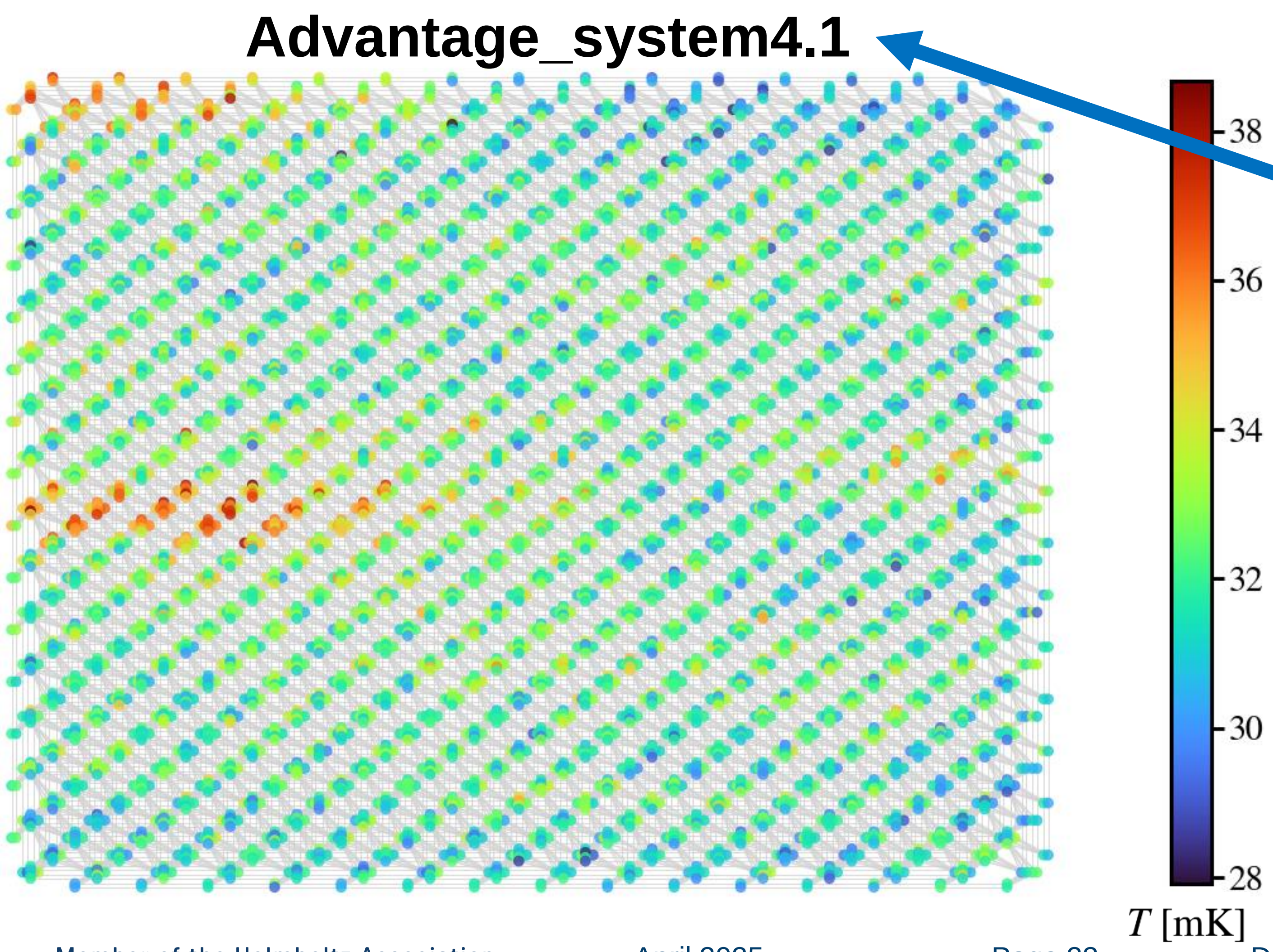


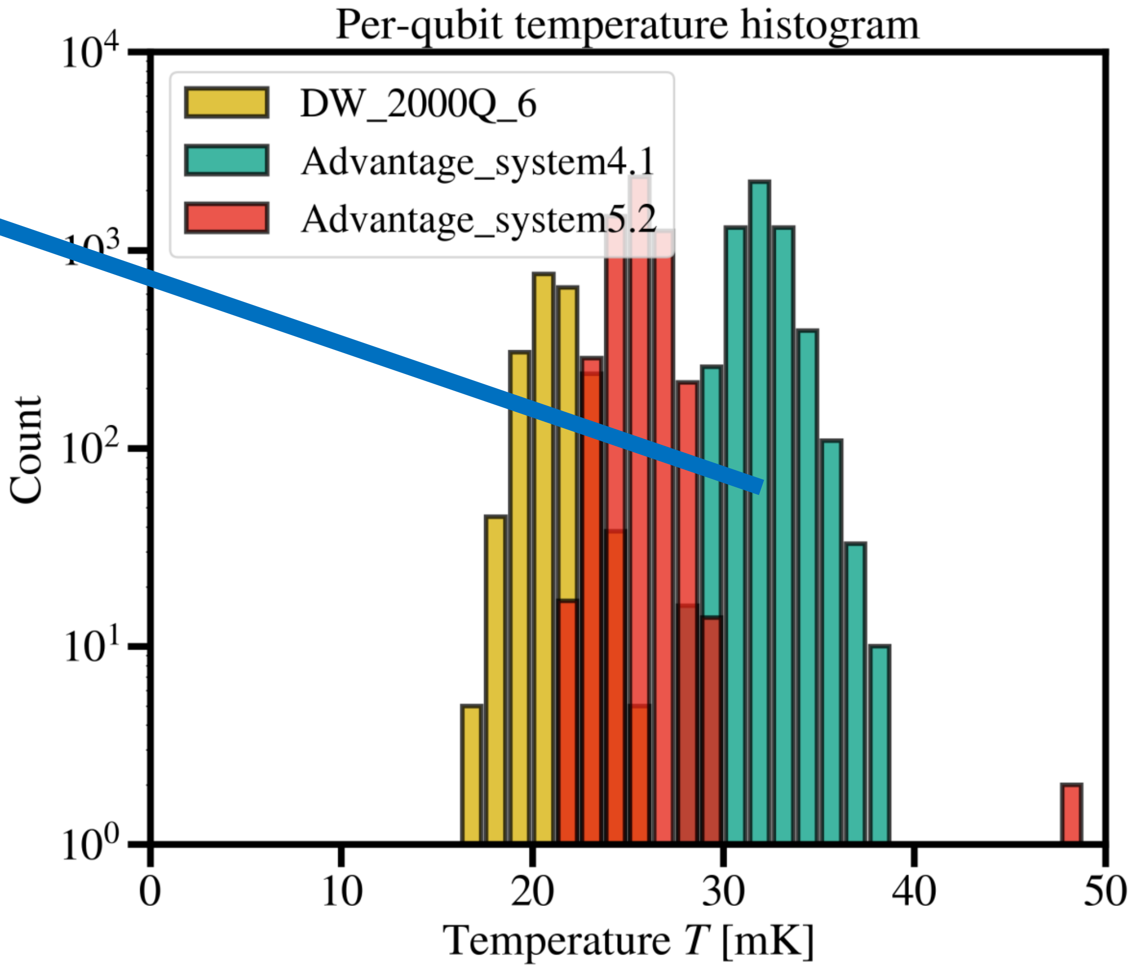
JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

Perot et al. (2022)
https://jugit.fz-juelich.de/qip/qbm-quant-finance

QUANTUM BOLTZMANN MACHINES

Experiment #1: Effective per-qubit temperature distribution



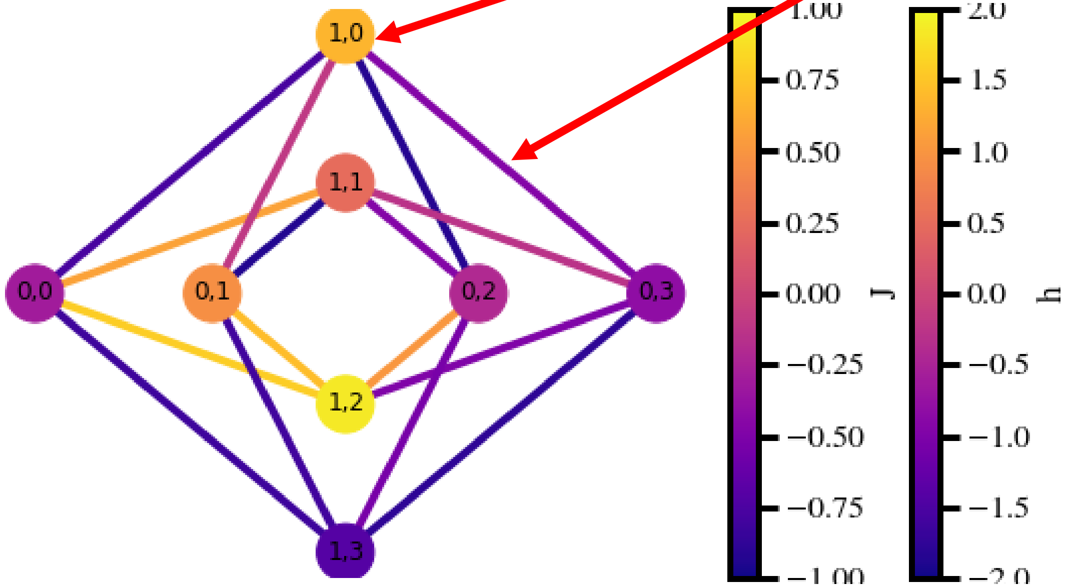


Perot et al. (2022)
https://jugit.fz-juelich.de/qip/qbm-quant-finance

QUANTUM BOLTZMANN MACHINES

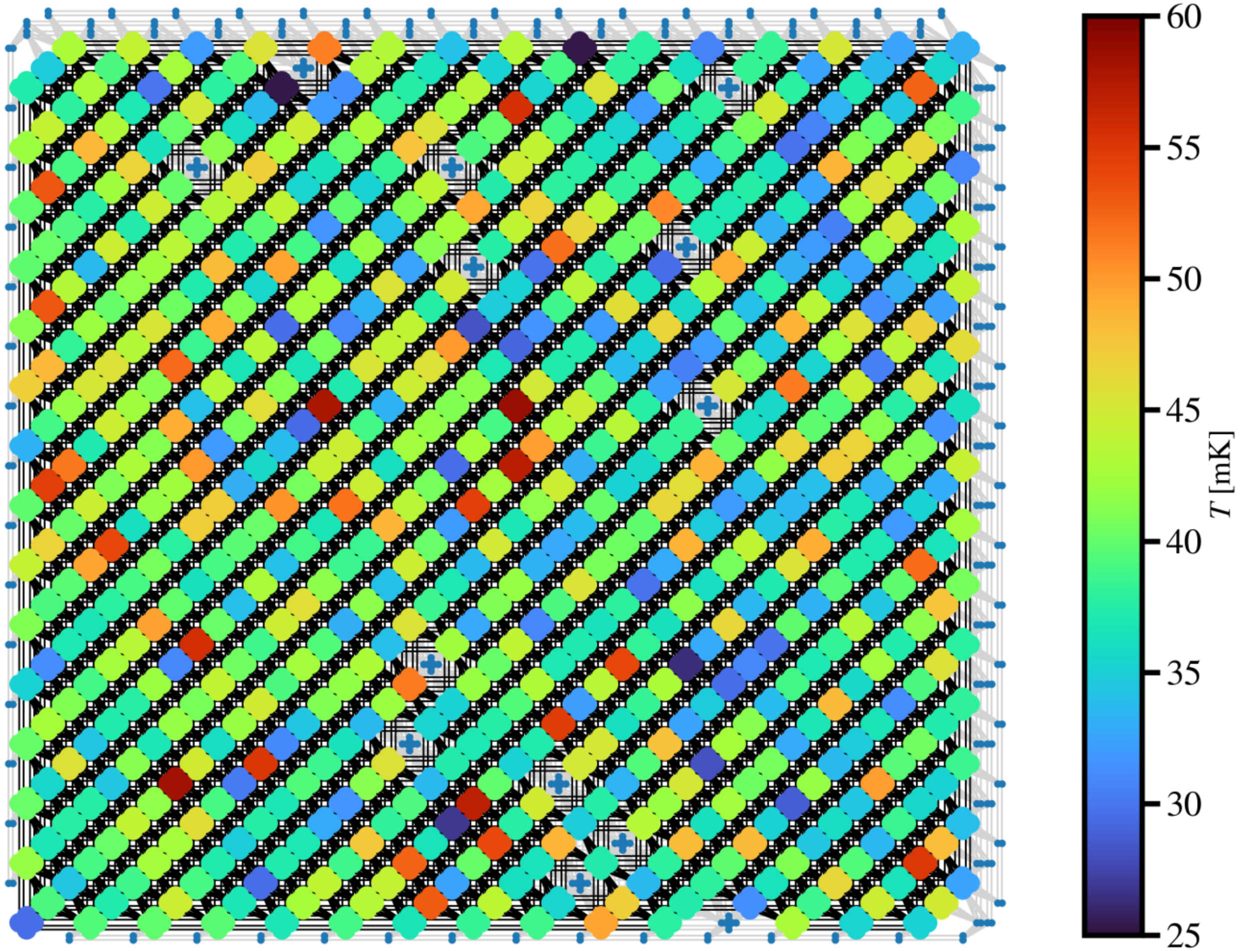
Experiment #2: 8-qubit problem on each Chimera cell of the Pegasus graph

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



Fit Boltzmann distribution to observed samples at $s = 1$ to infer $\beta = 1/k_B T$:

$$p(\{s_i = \pm 1\} \mid \beta, s) = \langle \{s_i = \pm 1\} | \frac{1}{Z} e^{-\beta H(s)} | \{s_i = \pm 1\} \rangle$$



Perot et al. (2022)
https://jugit.fz-juelich.de/qip/qbm-quant-finance

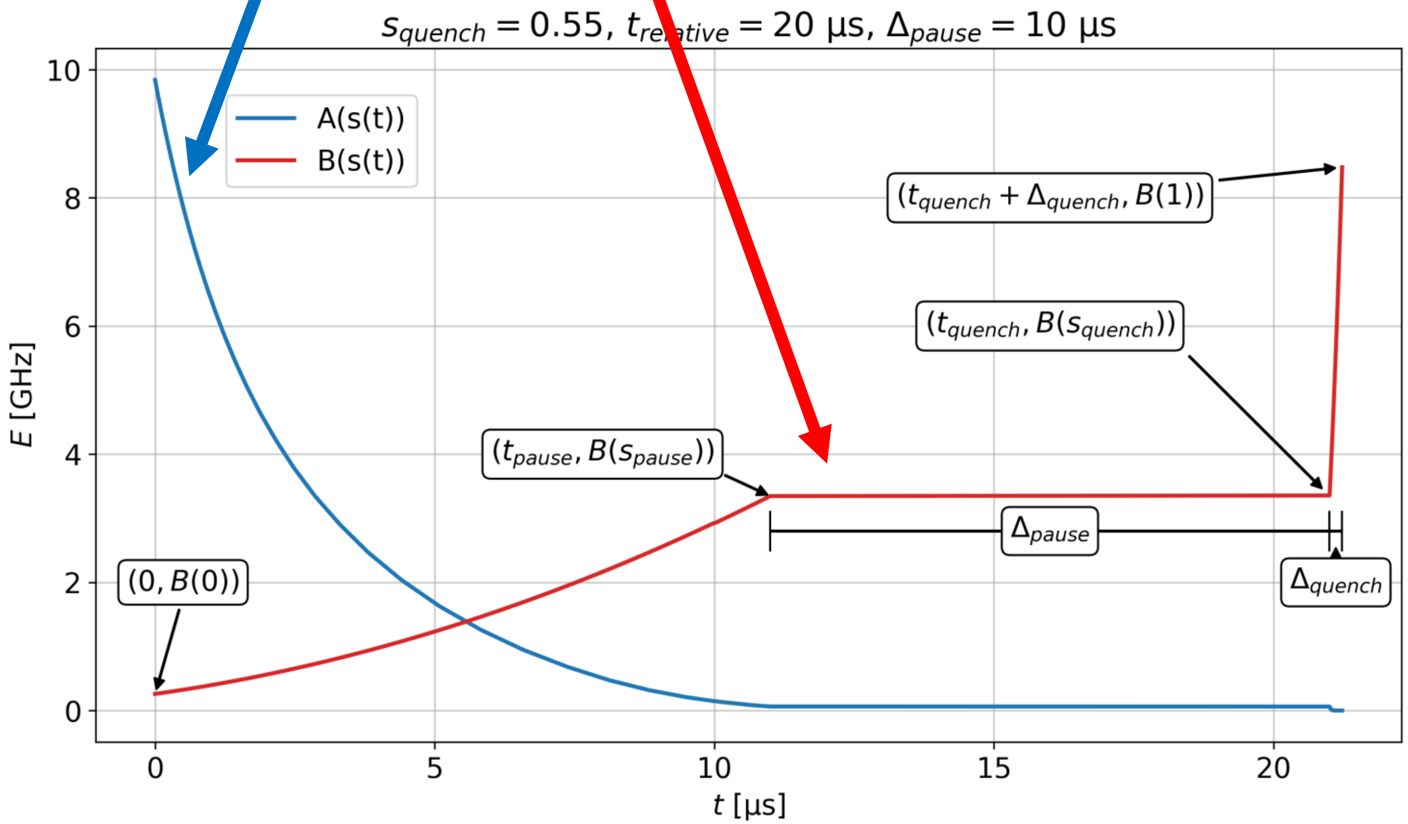
QUANTUM BOLTZMANN MACHINES

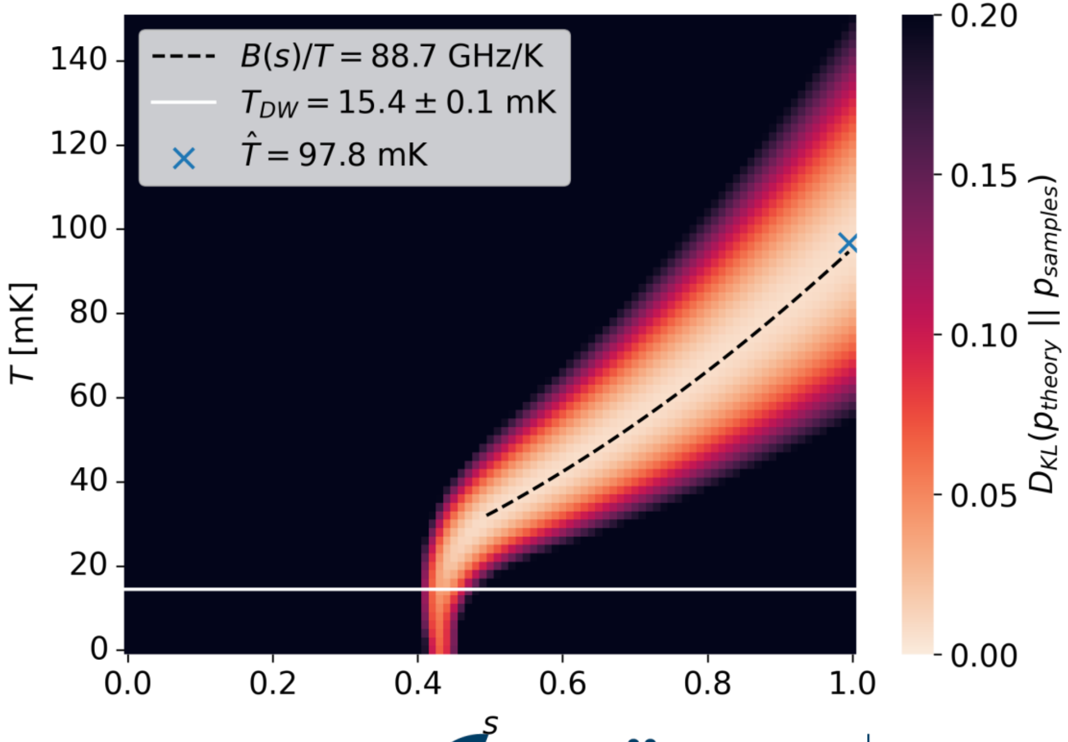
Experiment #3: 12-qubit problem (also for intermediate times s != 1)

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

Quantum Boltzmann distribution
at inverse temperature β and time s:

$$p(\{s_i = \pm 1\} \mid \beta, s) = \langle \{s_i = \pm 1\} | \frac{1}{Z} e^{-\beta H(s)} | \{s_i = \pm 1\} \rangle$$





JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

CONTENTS

1. Airline Scheduling
2. Traveling Salesman
3. Garden Optimization
4. Factorization
5. 2-Satisfiability
6. QSVM
7. QBM

optimization problem
with constraints
[increasing complexity]

constraints only

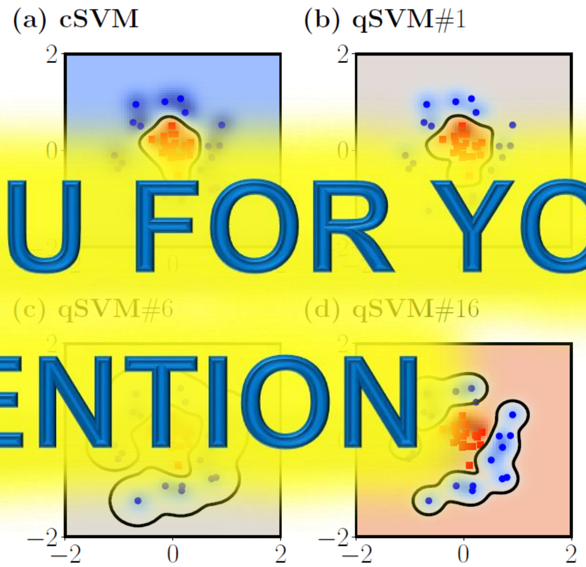
constraints only

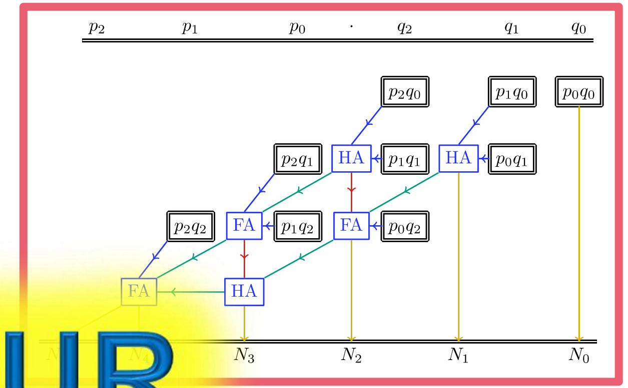
optimization only [basically]

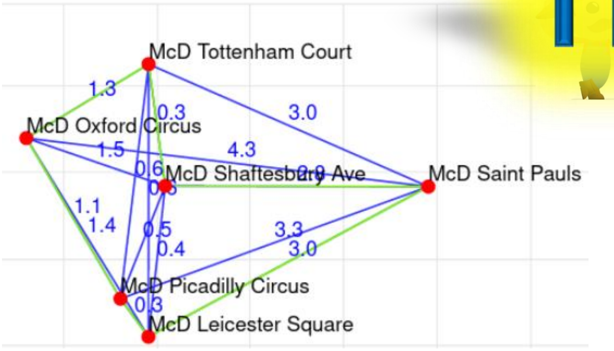
sampling problem

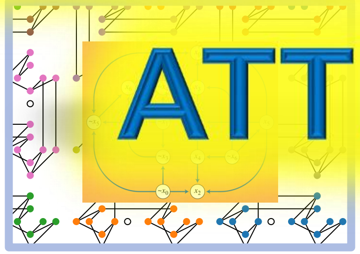
JUNIQ Rolling Call:

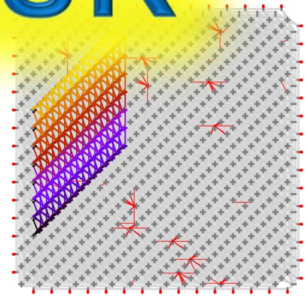
or search for
“FZJ JUNIQ ACCESS”









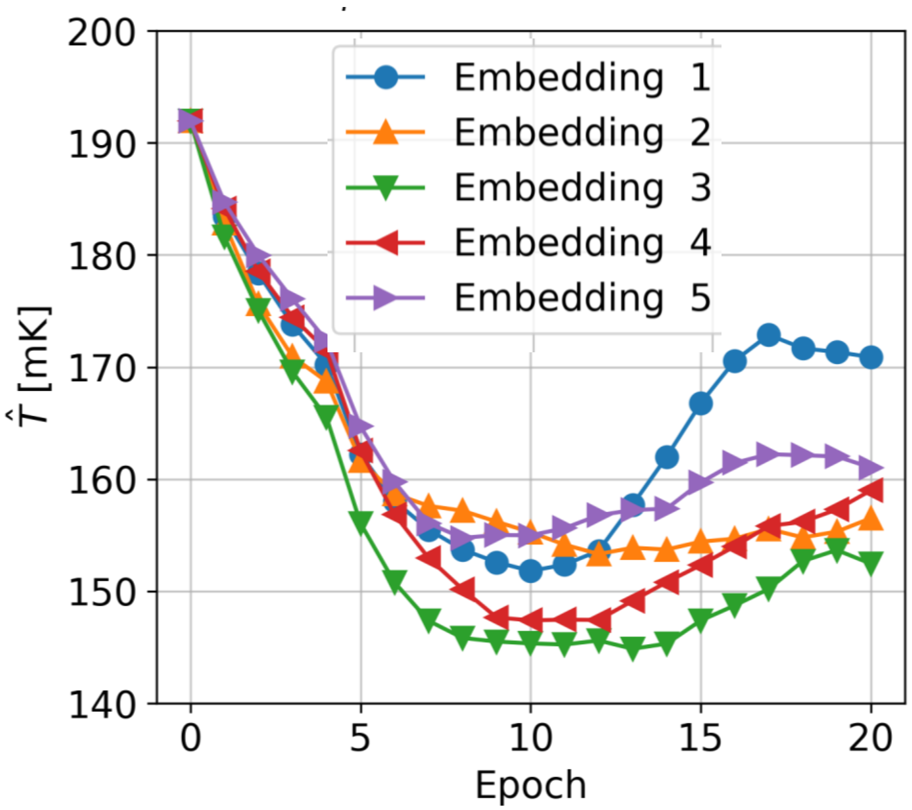


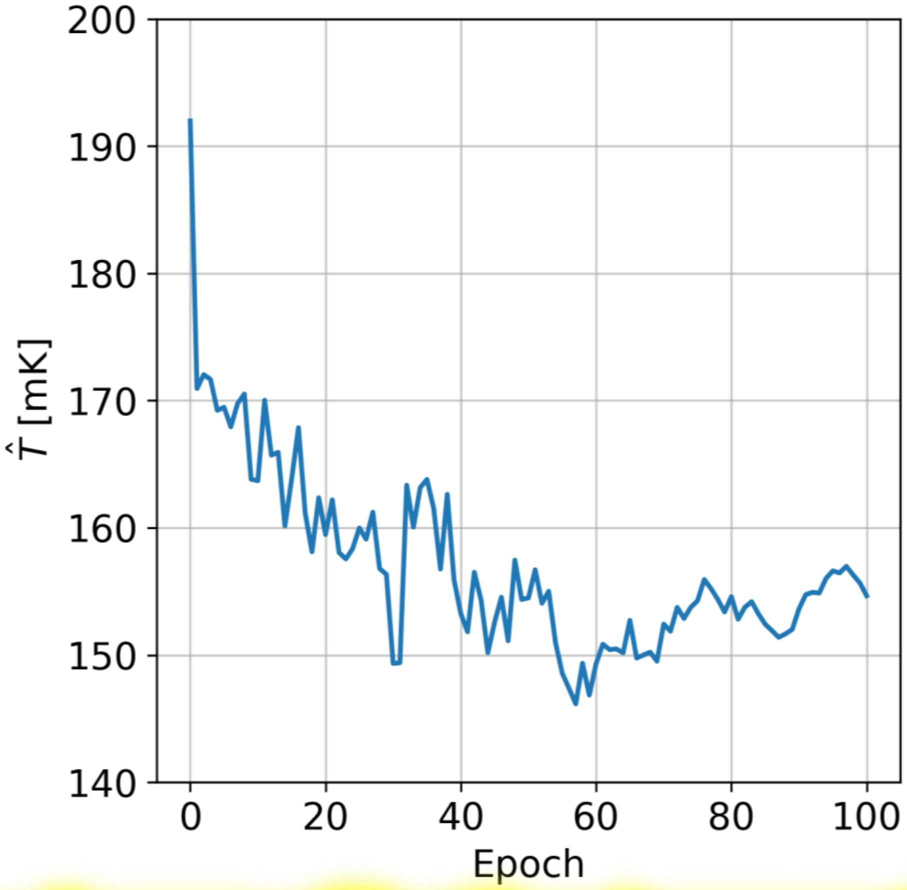
THANK YOU FOR YOUR ATTENTION

JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

QUANTUM BOLTZMANN MACHINES

Experiment #4: 96-qubit problem (~400 physical qubits, application in quantitative finance)





→ Observation: Larger problems ~ larger effective temperature