APPLICATIONS ON QUANTUM ANNEALERS AT FZJ

JUNIQ: Jülich Unified INfrastructure for Quantum Computing

INQA CONFERENCE 2022 I DR. DENNIS WILLSCH

CONTENTS

1. Airline Scheduling
2. Traveling Salesman
3. Garden Optimization
4. 2-Satisfiability
5. Quantum Support Vector Machines
6. Quantum Boltzmann Machines

Prof. Dr. Kristel Michielsen

Prof. Dr. Hans De Raedt

Dr. Dennis Willsch

Dr. Madita Willsch

Vrinda Mehta

Carlos Gonzalez Calaza

Manpreet Jattana

Tamás Nemes

Cameron Perot

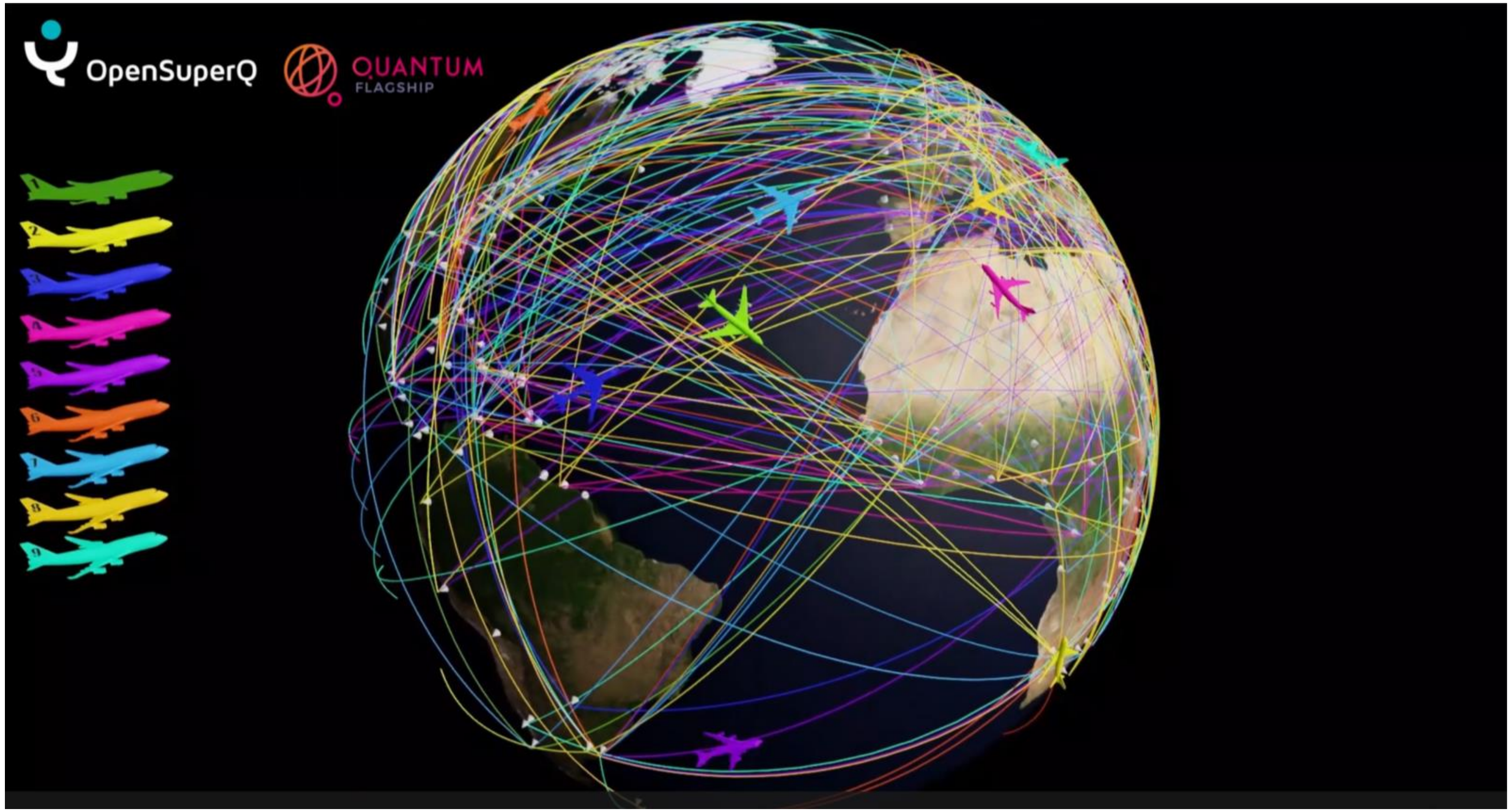
Dr. Fengping Jin

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

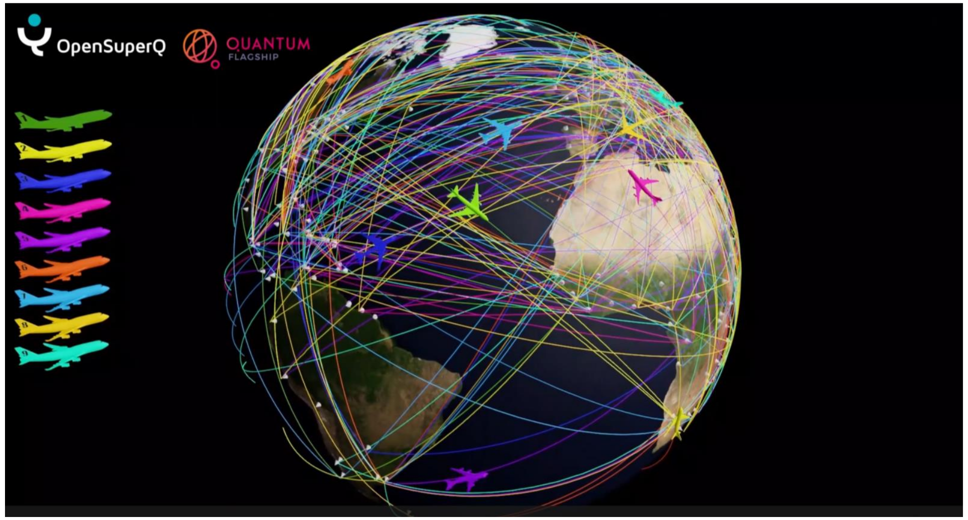
BOEING
OpenSuperQ
QUANTUM FLAGSHIP

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

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

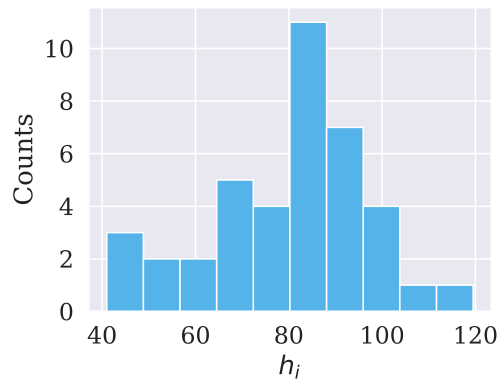
encodes cost of assigning airplanes to routes

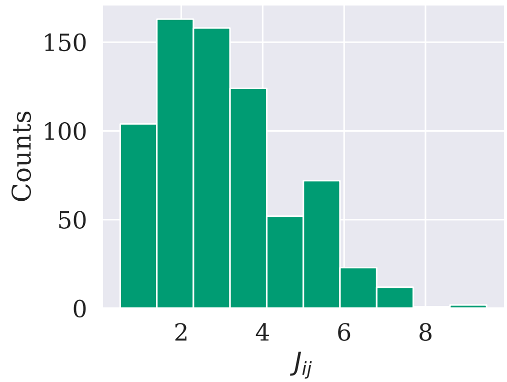
exact cover problem: each flight covered once

- Reformulation as Ising problem

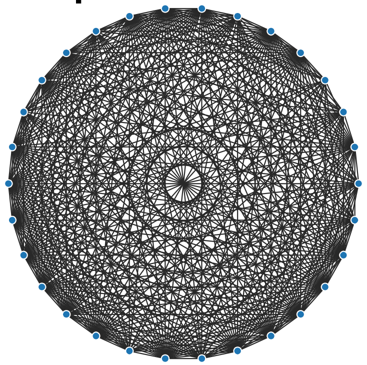
$$\text{QUBO} \leftrightarrow \text{Ising}: \quad x_i = (1+s_i)/2$$

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

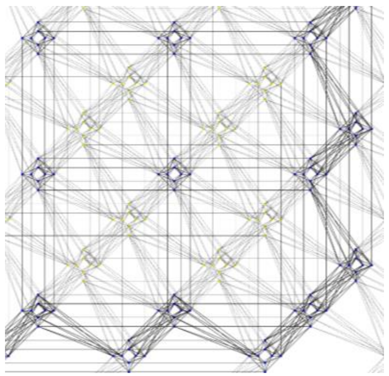




- 25 - 40 qubit almost fully-connected (“clique”) Ising problems



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)

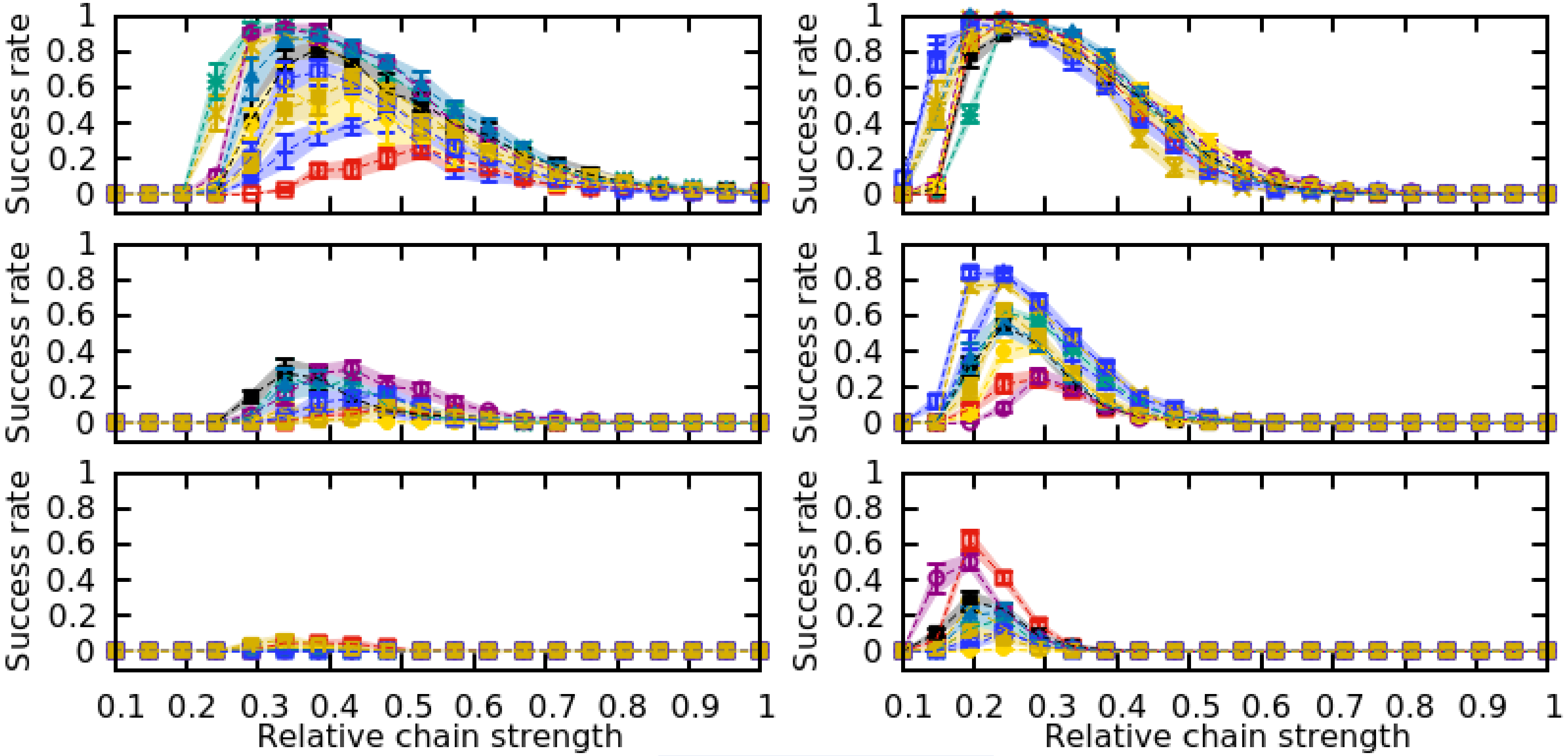
Scan of 10 different embeddings and 20 relative chain strengths:

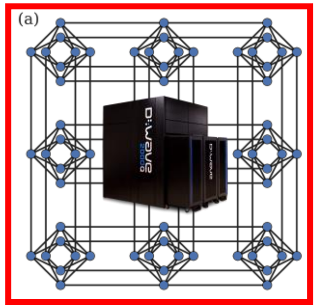
DW2000Q **Advantage**

30 qubits

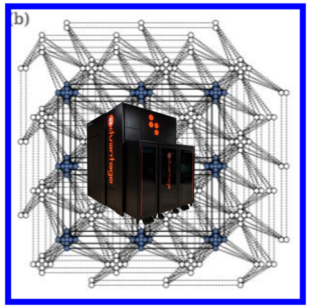
36 qubits

40 qubits





vs.



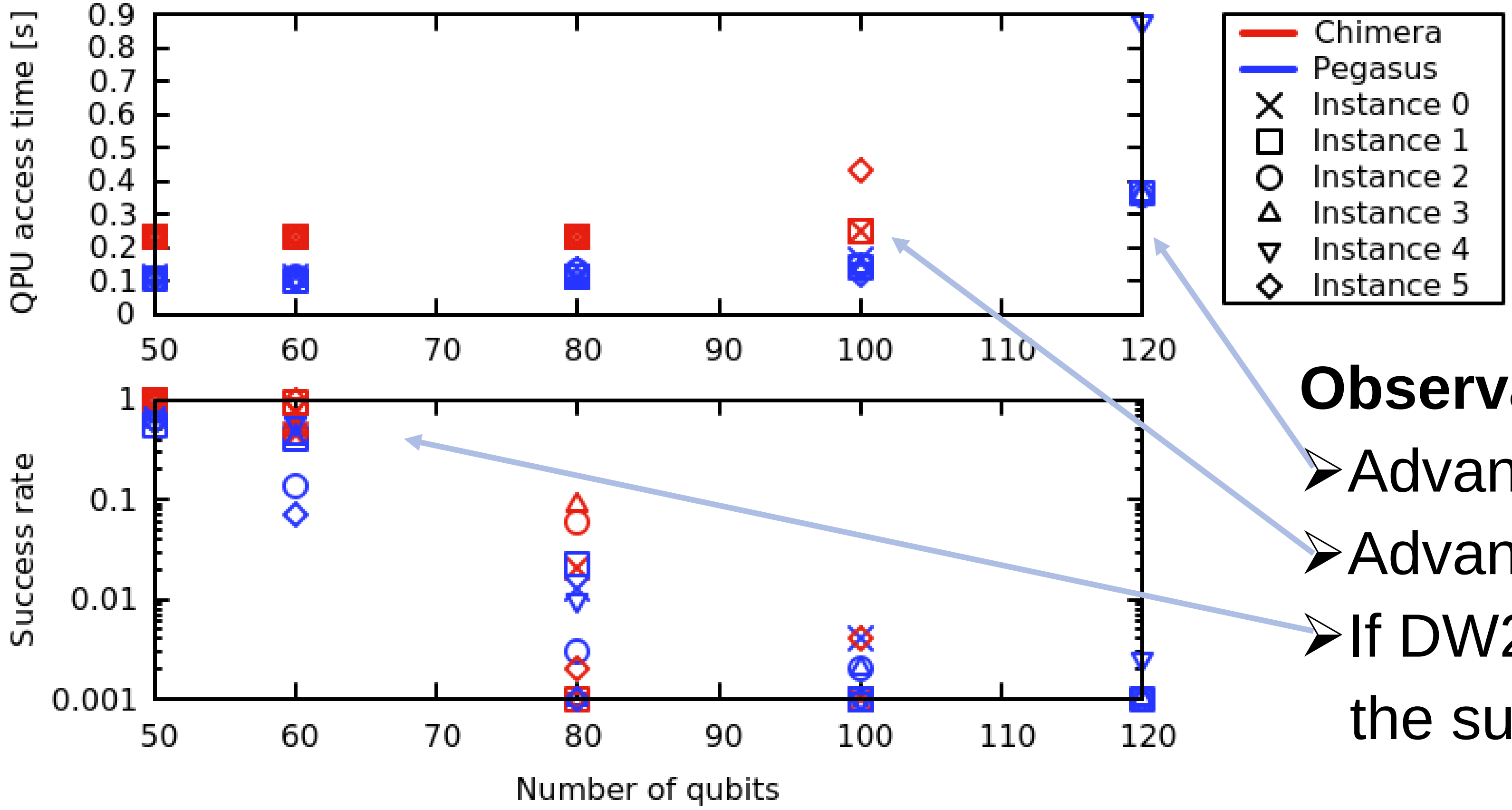
Relative chain strength
= CS / max(|hi|, |Jij|)

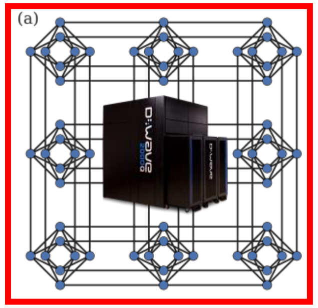
Willsch et al. (2022)
QINP **21**, 141
arXiv:2105.02208

TAIL ASSIGNMENT: RESULTS

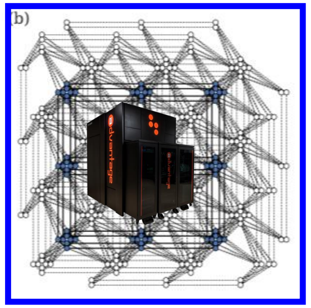
50-120 qubits: larger but sparser problems (20% nonzero couplers)

The fastest successful runs that reproducibly gave a solution:





vs.



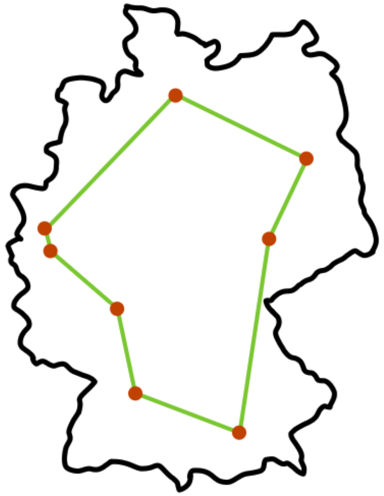
Observations:

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

Nemes et al. (2022)
https://tsp-quantum.netlify.app/

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

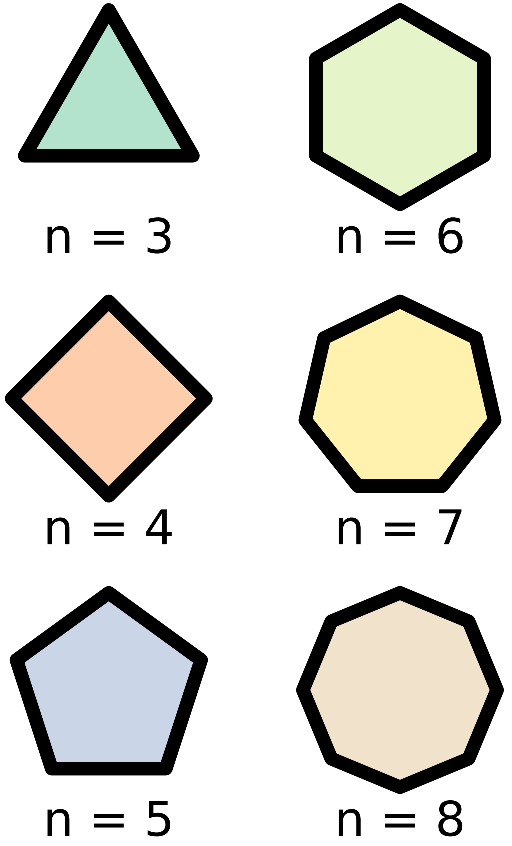
	0	1	2	3
A	1	0	0	0
B	0	0	0	1
C	0	1	0	0
D	0	0	1	0

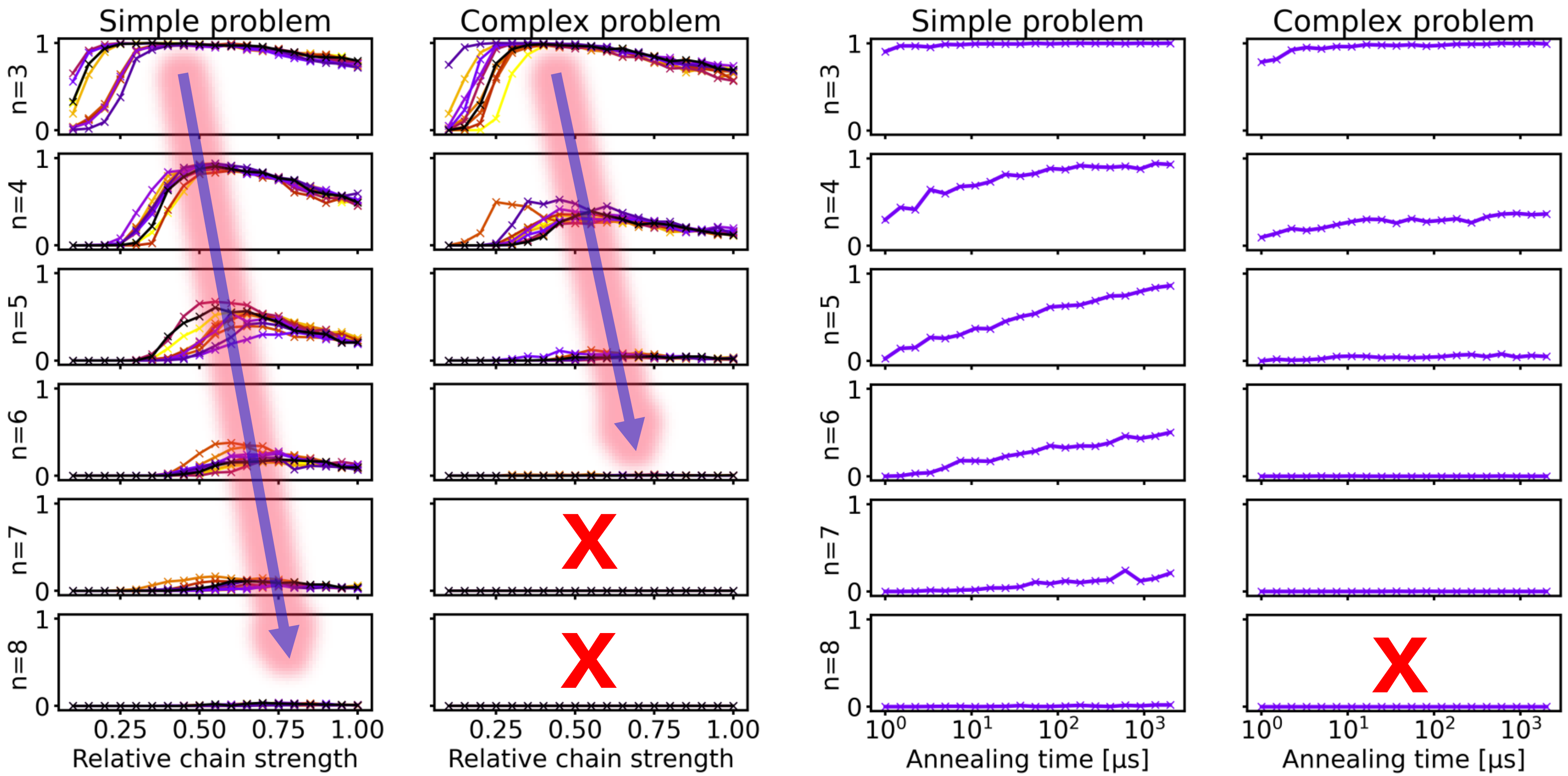
- Assign cities & times to qubits:
  - Qubit $x_{it} = 1$ means the traveler is at city i at time t
  - Count the cost c_{ij} if the traveler goes from i to j at some time step
  - Traveler must pass city i only once
  - Traveler can only be at one city at each time step
- Not the DFJ formulation: linear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

Nemes et al. (2022)
https://tsp-quantum.netlify.app/

TRAVELING SALESMAN PROBLEM

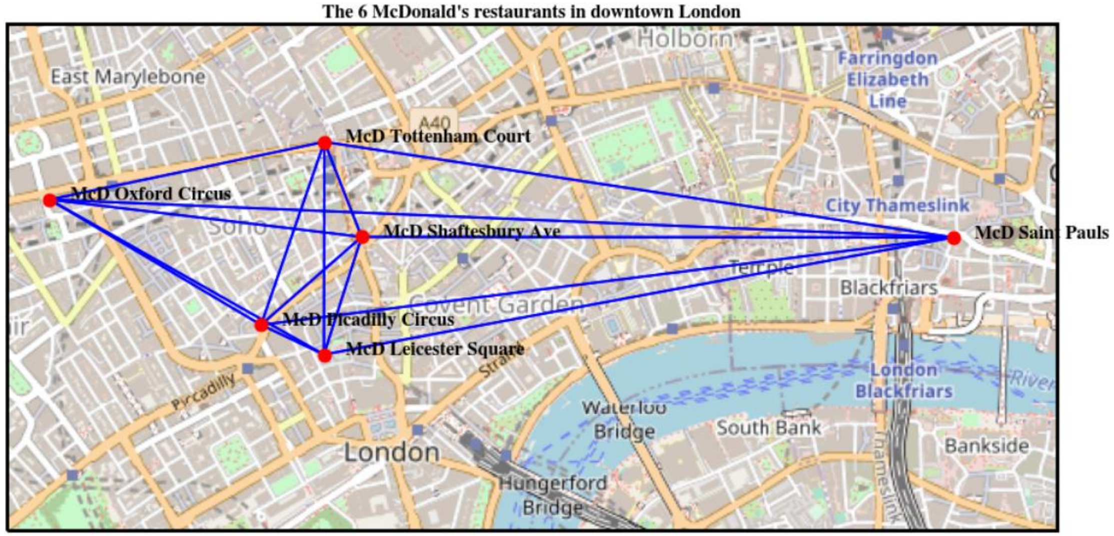
Scaling of chain strength and annealing time

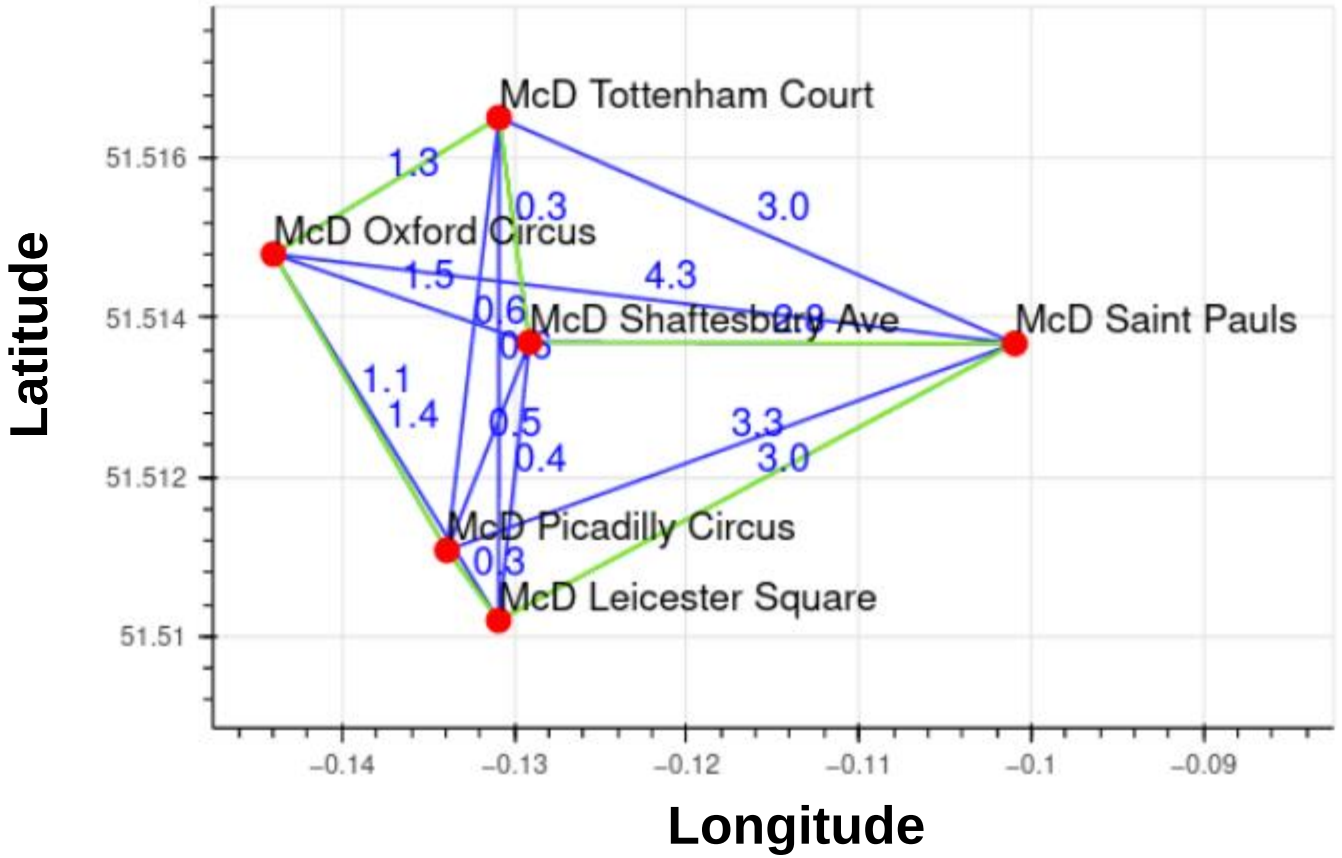




TRAVELING SALESMAN PROBLEM

The 6 McDonald's restaurants in downtown London



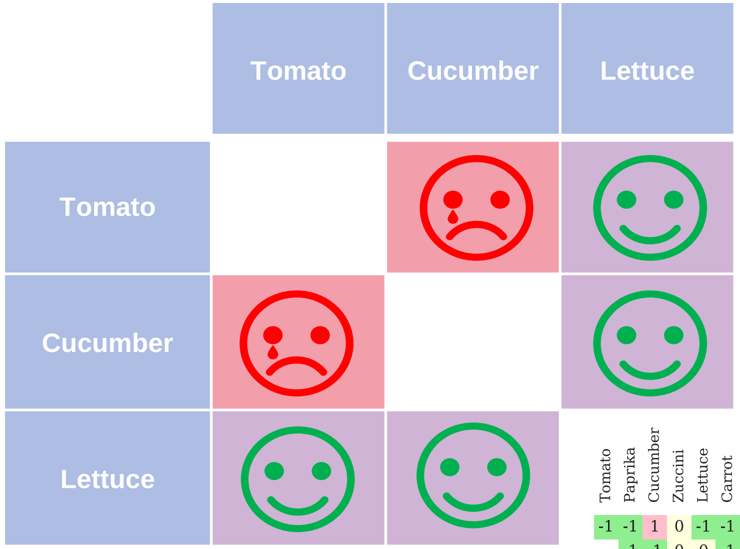


Gonzalez Calaza et al. (2021)
QINP **20**, 305
arXiv:2101.10827
https://jugit.fz-juelich.de/qip/garden-optimization-problem

VEGETABLE GARDEN OPTIMIZATION

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

➢ 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

$$\min_{x_{ij}=0,1} \Bigg\{ \sum_{i,i'=0}^{n-1} J_{ii'} \left(1 + \sum_{j,j'=0}^{t-1} x_{ij} C_{jj'} x_{i'j'}\right)$$
$$+\lambda_1 \sum_{i=0}^{n-1} \left(1 - \sum_{j=0}^{t-1} x_{ij}\right)^2$$
$$+\lambda_2 \sum_{j=0}^{t-1} \left(c_j - \sum_{i=0}^{n-1} x_{ij}\right)^2$$
$$+\lambda_3 \sum_{i=0}^{n-1} \sum_{j=0}^{t-1} \left(i\%2 - s_j\right)^2 x_{ij} \Bigg\}$$

- 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

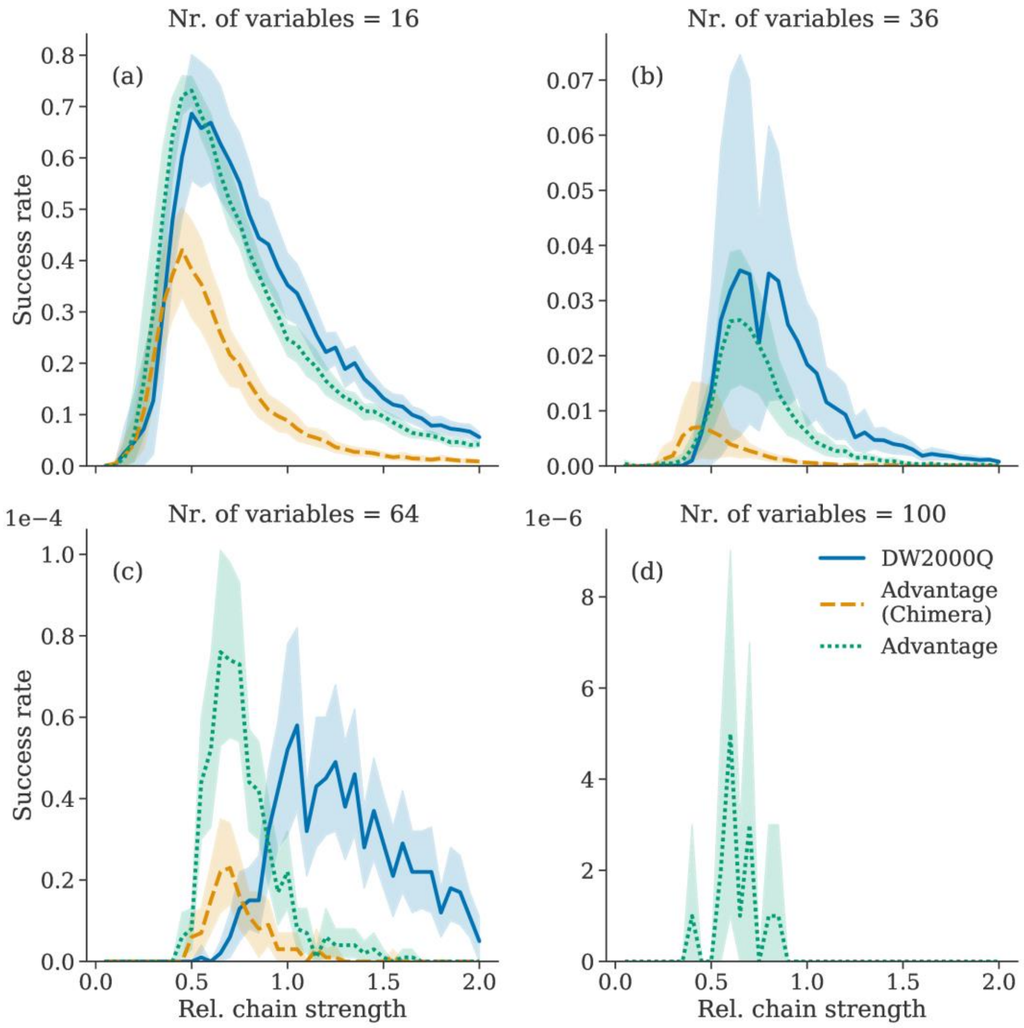
n: nr. of pots in the garden
t: nr. of plant species to place
J: connectivity between pots
C: relationship between species
c_j: nr. of plants of species j
s_j: size of species j

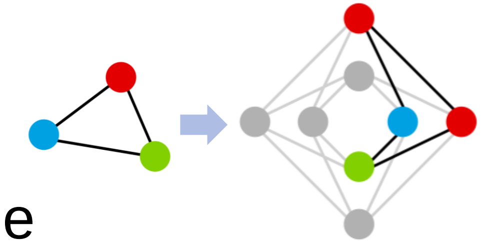
JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

VEGETABLE GARDEN OPTIMIZATION

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

Finding the optimal chain strength: Results





Experiment:

- 4 problems of increasing size
- Success = constraints weren't violated
- 10 different embeddings for each system
- Scanned 40 different values for:

  Rel. chain strength = CS / max(|ai|,|bij|)

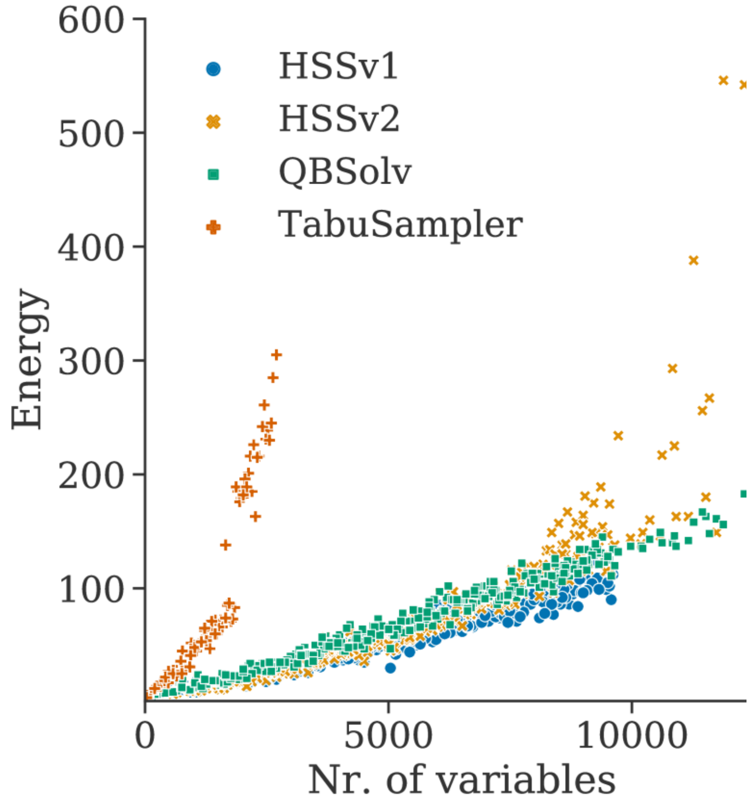
Conclusions:

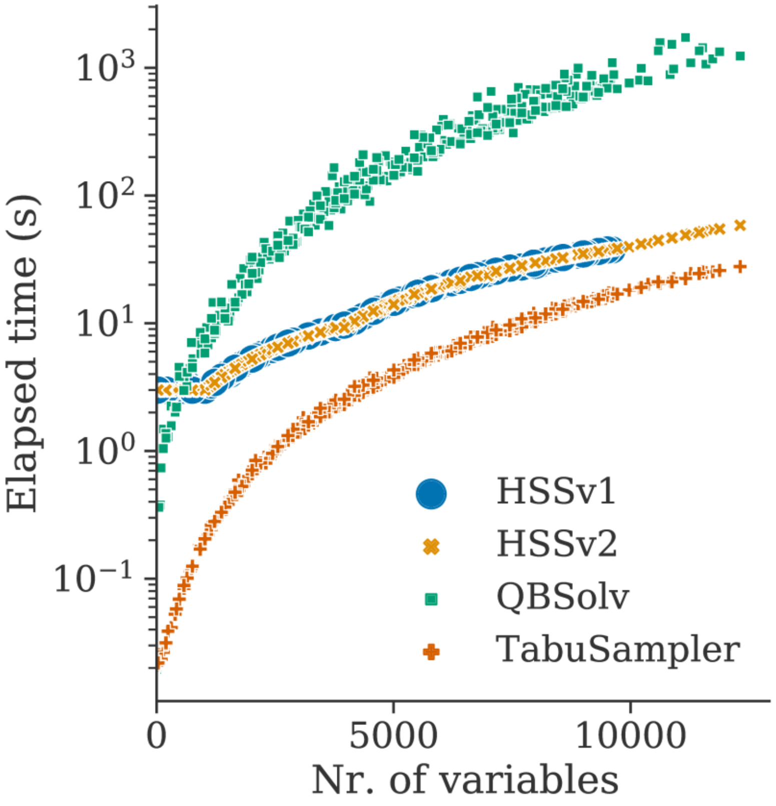
- Trying several embeddings is very important
- Chain strength heavily affects success rate
- Longer chains require higher chain strengths

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



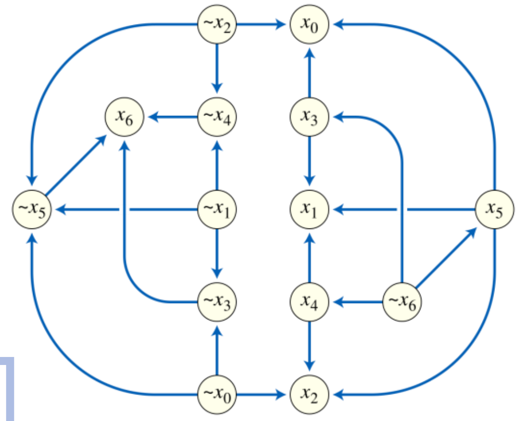


Results:
- HSSv1: best but only up to 10k vars.
- HSSv2: same as v1 up to 2k vars., worse later.
- QBSolv: good but very slow.
- TabuSampler: returns unusable results extremely fast.

Conclusion:
- Hybrid solvers outperform classical solvers.

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: $$\min_{s_i=\pm 1} \left(\sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j \right)$$

- 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 $\overline{x_i}$

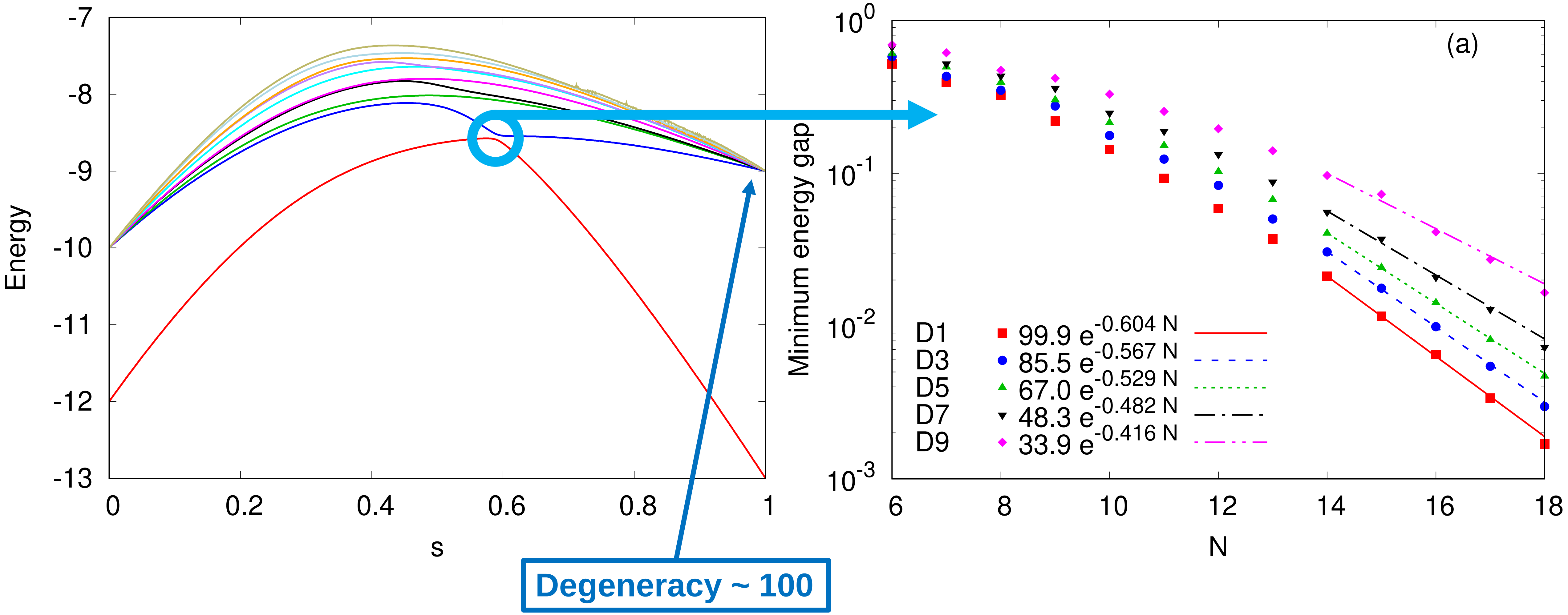
  - Example for clause $x_1 \vee x_2$: $h_{2SAT} = s_1 s_2 - (s_1 + s_2) + 1$

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

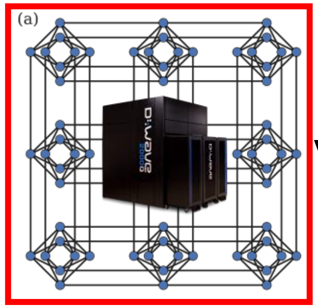
Mehta et al., PRA **105**, 062406 (2022)
Mehta et al., PRA **104**, 032421 (2021)

2-SATISFIABILITY

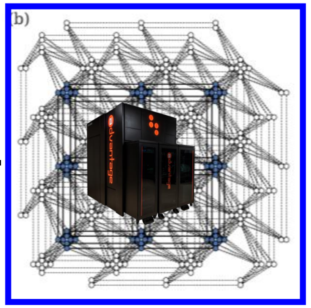
Properties: unique ground state, highly degenerate first excited level, small gaps



2-SATISFIABILITY: RESULTS



vs.

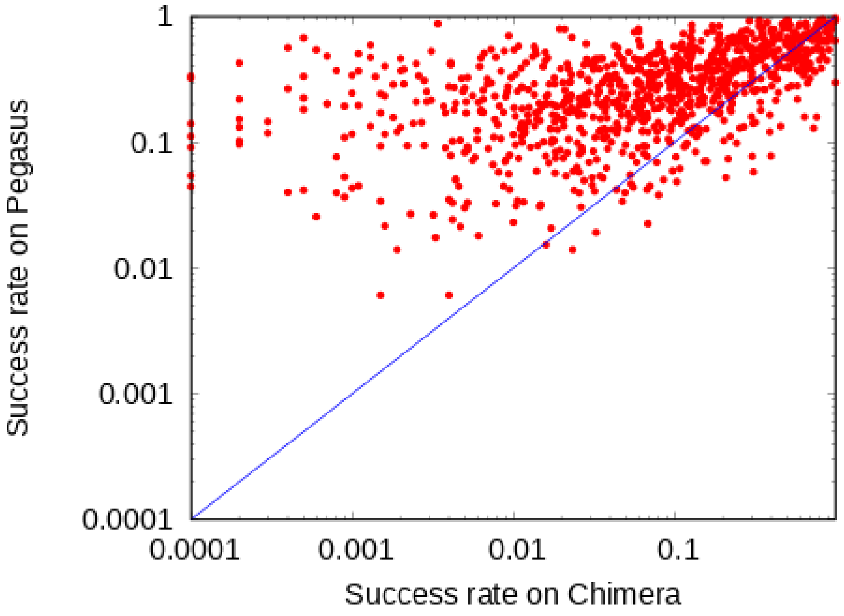


Mehta et al., PRA **105**, 062406 (2022)
Mehta et al., PRA **104**, 032421 (2021)

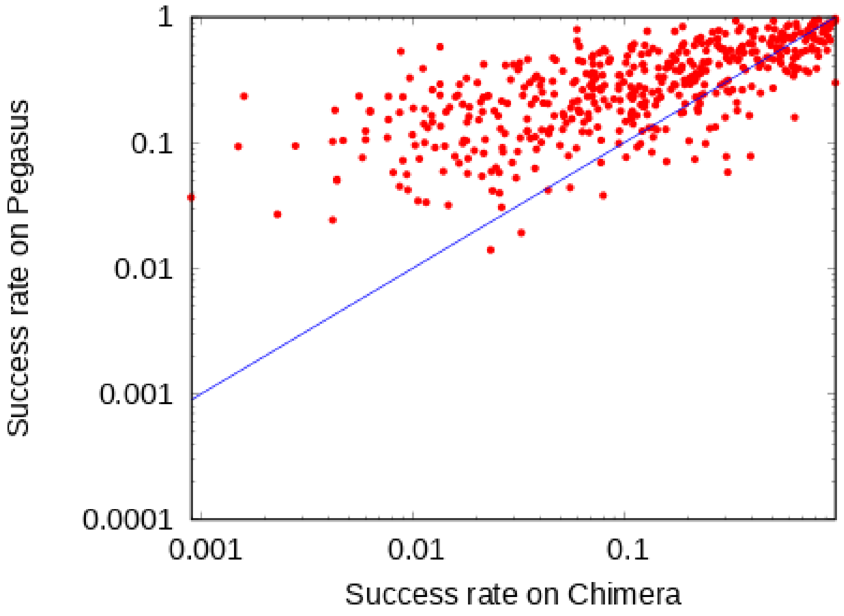
Success rate for 1000 2-SAT problems

- **Direct mapping:**
  ~50% on DW2000Q
  ~90% on Advantage
- Advantage performs better for a majority of the problems, especially the difficult problems

All problems



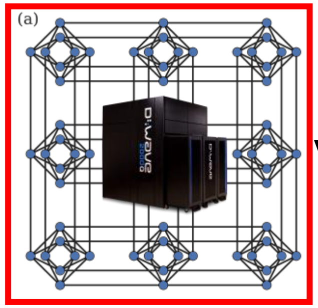
Problems with direct mapping



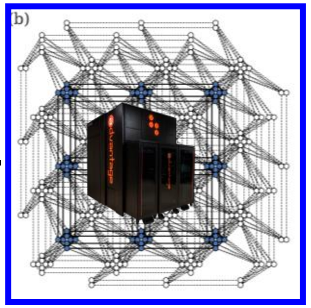
- Largest enhancement for cases that require embedding only on Chimera
  ➔ Improvement due to increased connectivity

2-SATISFIABILITY: RESULTS

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

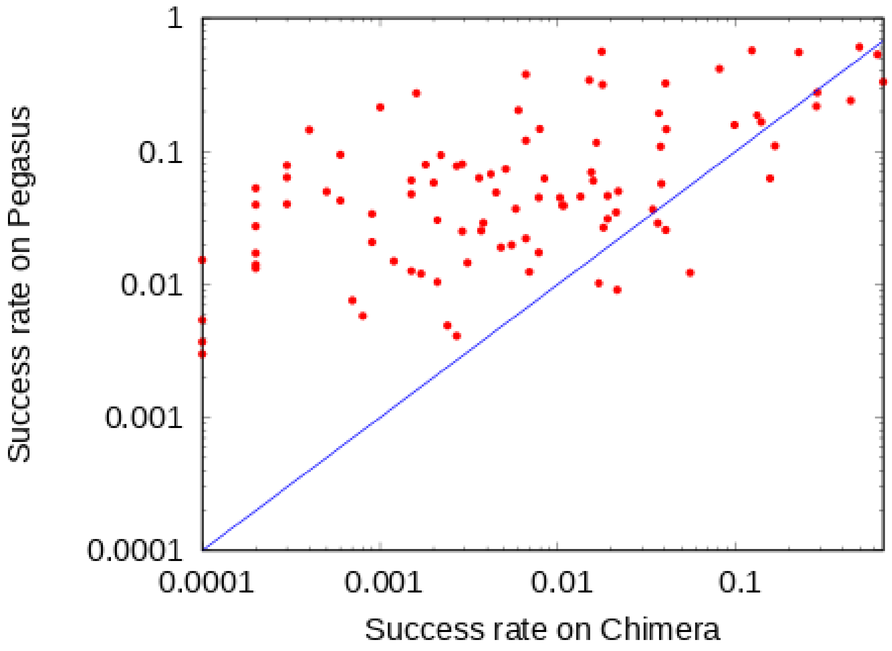


vs.

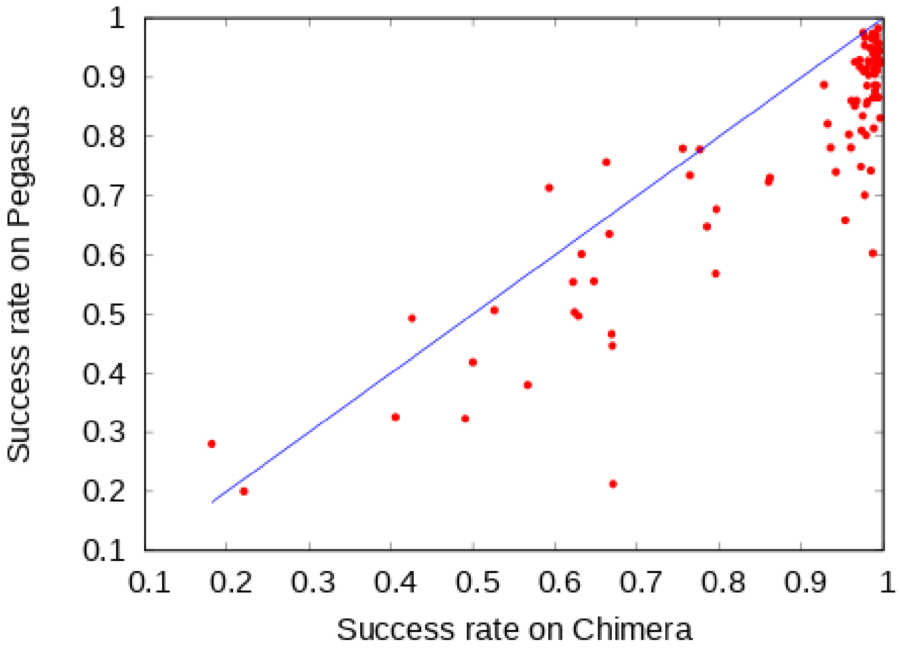


Success rate for 2 particular 2-SAT problems

Hard Problem



Easy problem

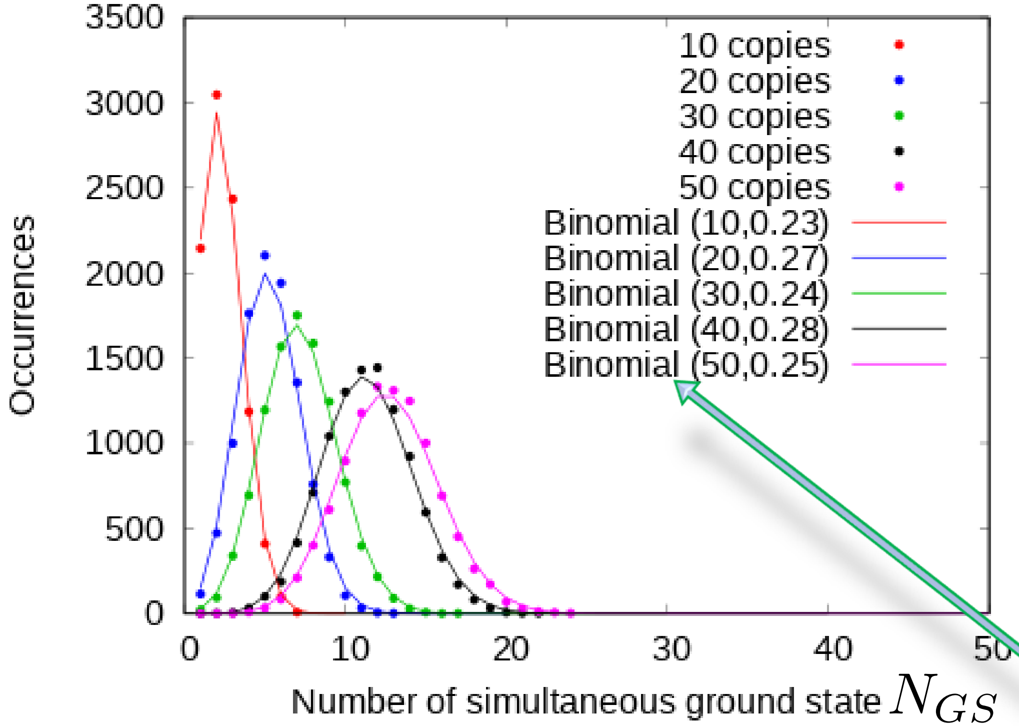


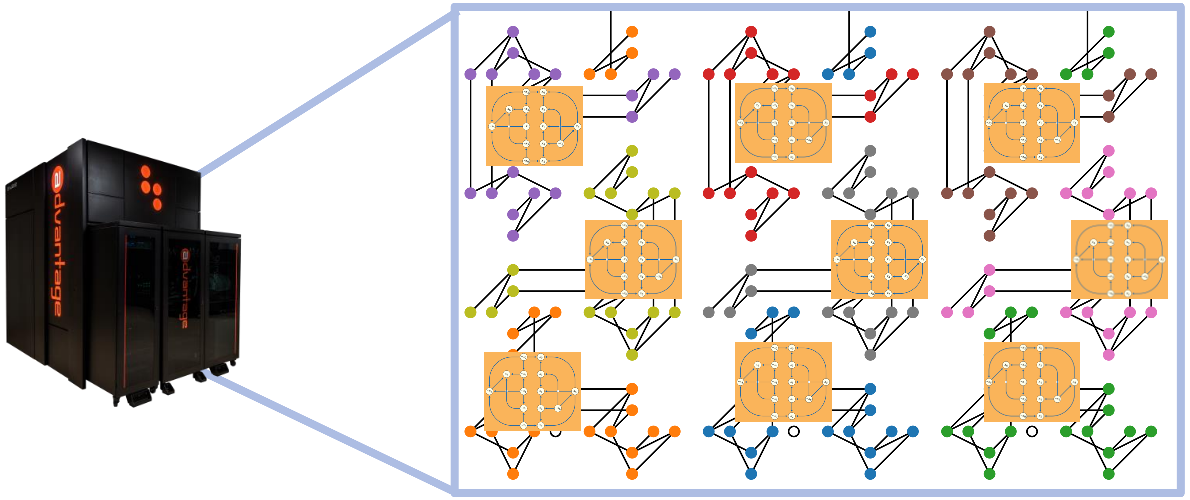
- For a hard problem, Advantage performs better
- However, for an easy problem, DW2000Q performs better

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

2-SATISFIABILITY: RESULTS

Embed the same problem multiple times





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

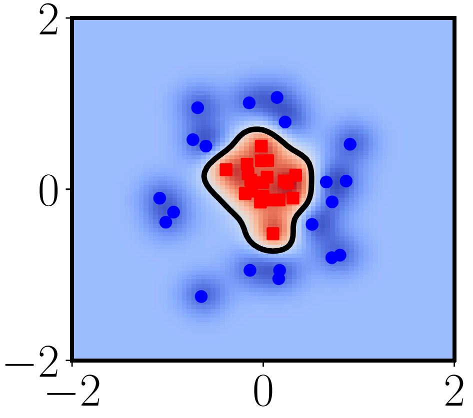
Success rate: probability to find the unique solution (GS)

- The occurrence of optimal solutions follows a binomial distribution
- Parts of the solver corresponding to each copy work almost independently
- Note that in no run could all ground states be found simultaneously

Willsch et al., CPC **248**, 107006 (2021)
Delilbasic et al., IGARSS 2021, 2608 (2021)
Cavallaro et al., IGARSS 2020, 1973 (2020)

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

Base & Exponent

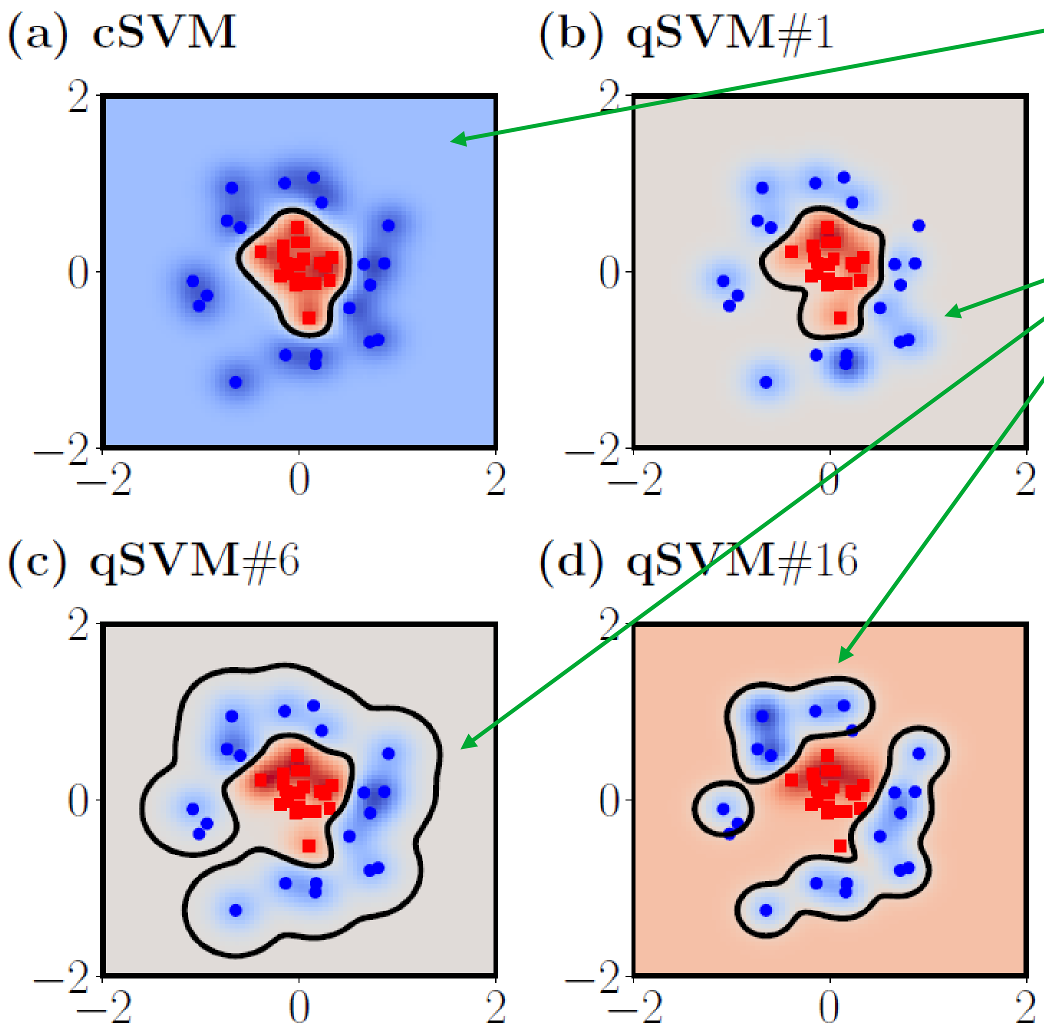
$$\min_{x_i = 0,1} \left(\sum_{i,j} x_i Q_{ij} x_j \right)$$

JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

QUANTUM SUPPORT VECTOR MACHINES

Willsch et al., CPC **248**, 107006 (2021)
Delilbasic et al., IGARSS 2021, 2608 (2021)
Cavallaro et al., IGARSS 2020, 1973 (2020)

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

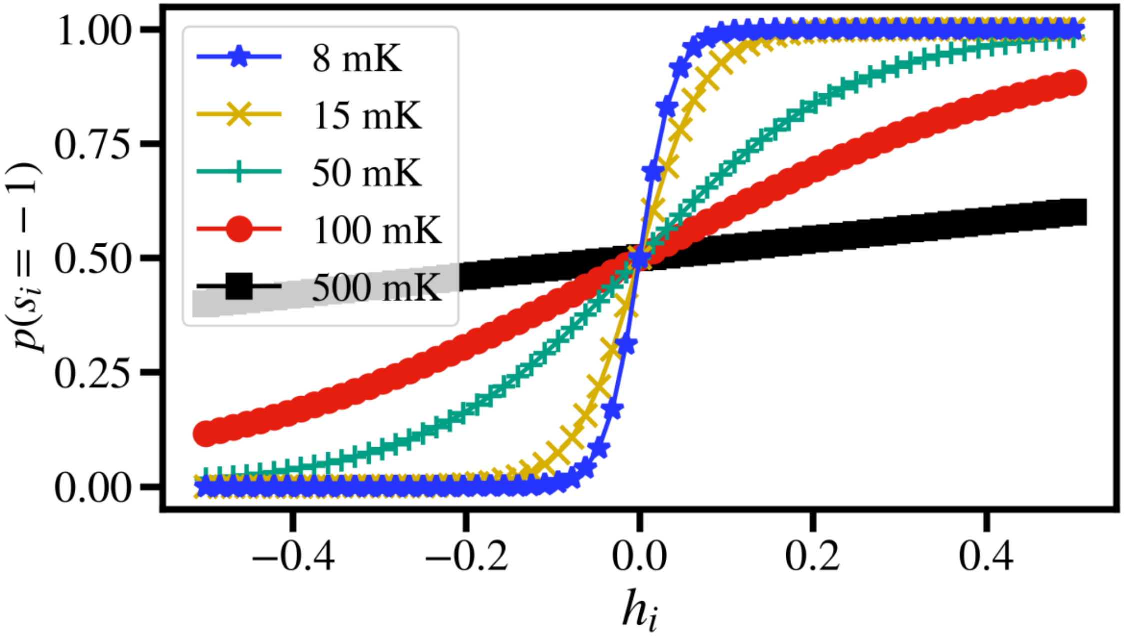
Combination of multiple low-energy qSVM classifiers generalizes better to unseen data

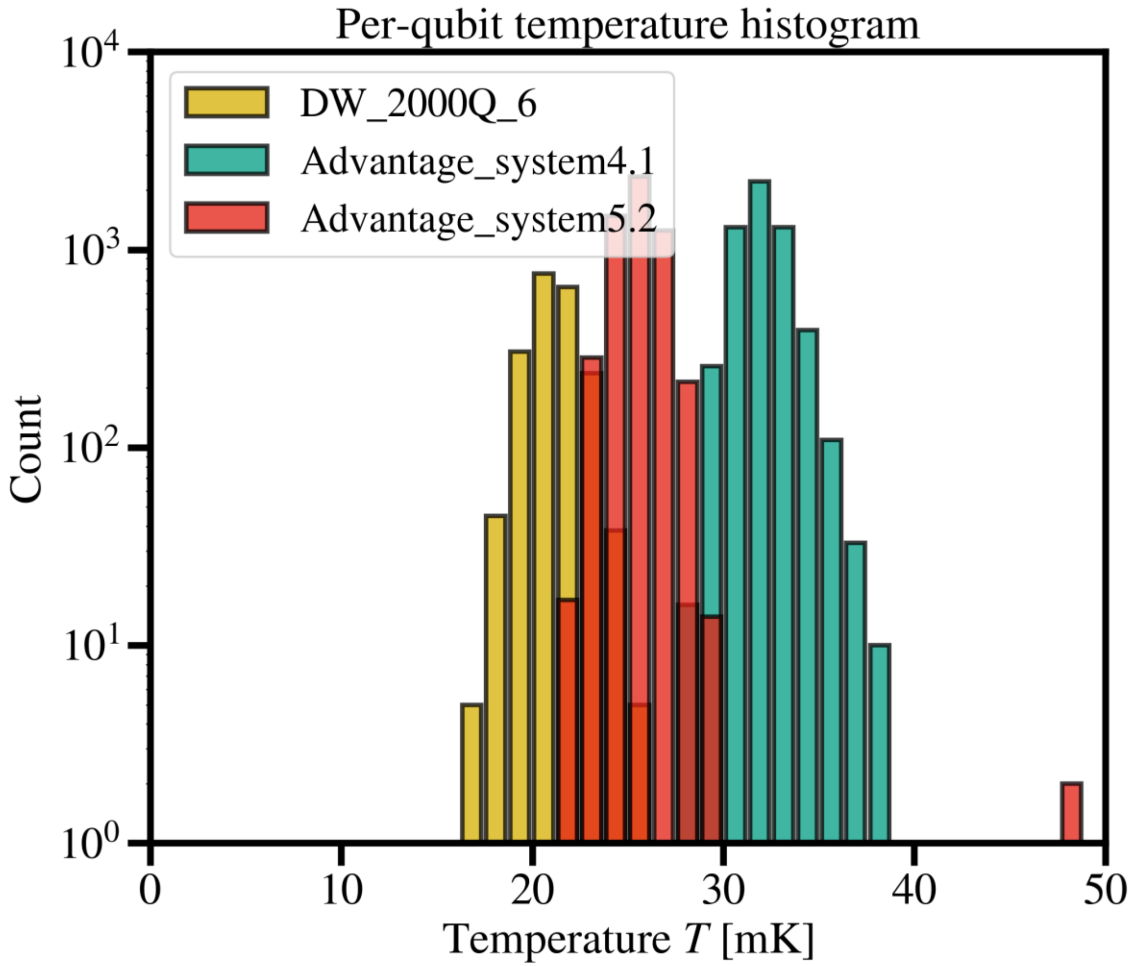
QUANTUM BOLTZMANN MACHINES

Experiment #1: Effective per-qubit temperature distribution

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

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

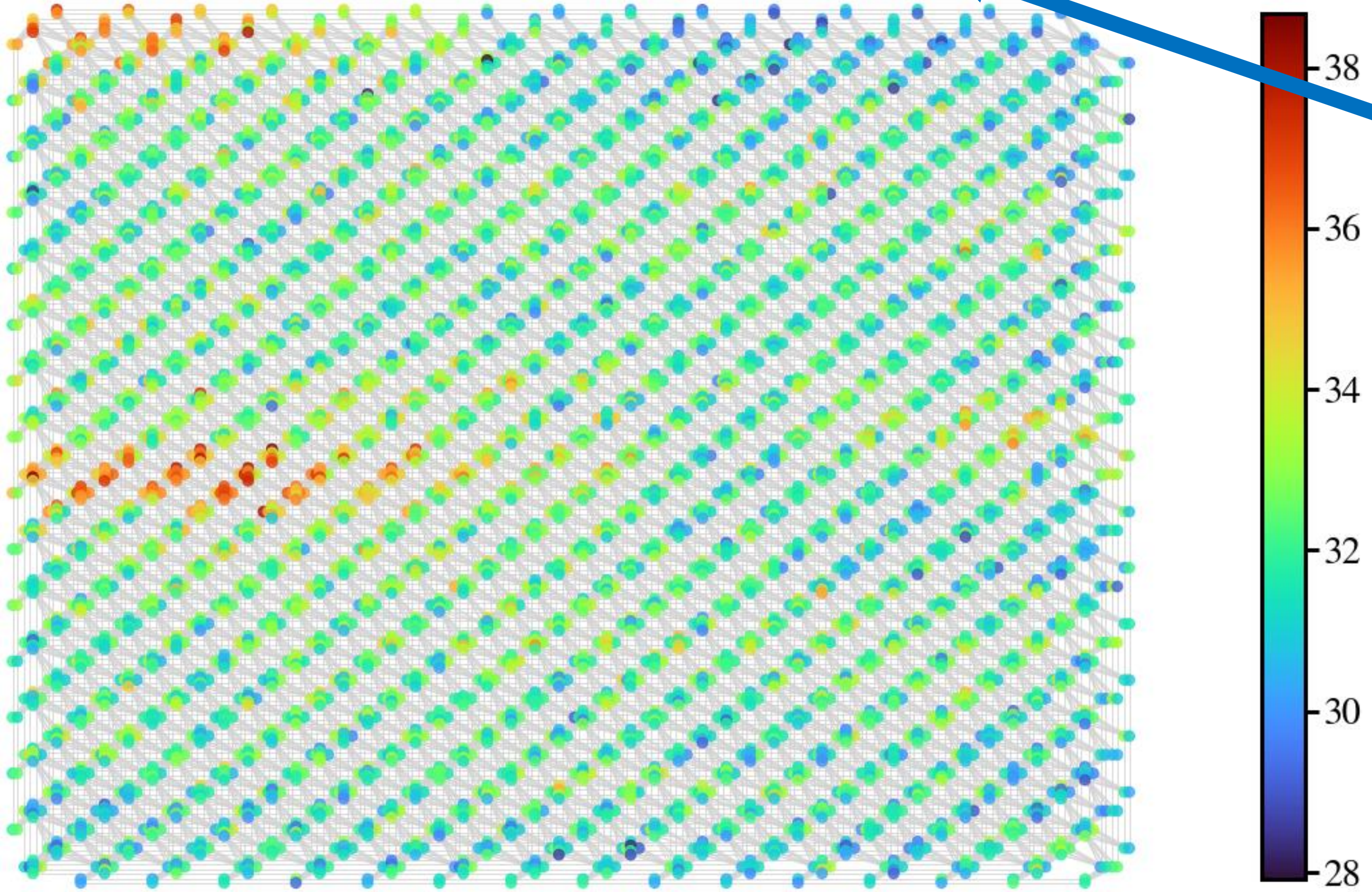


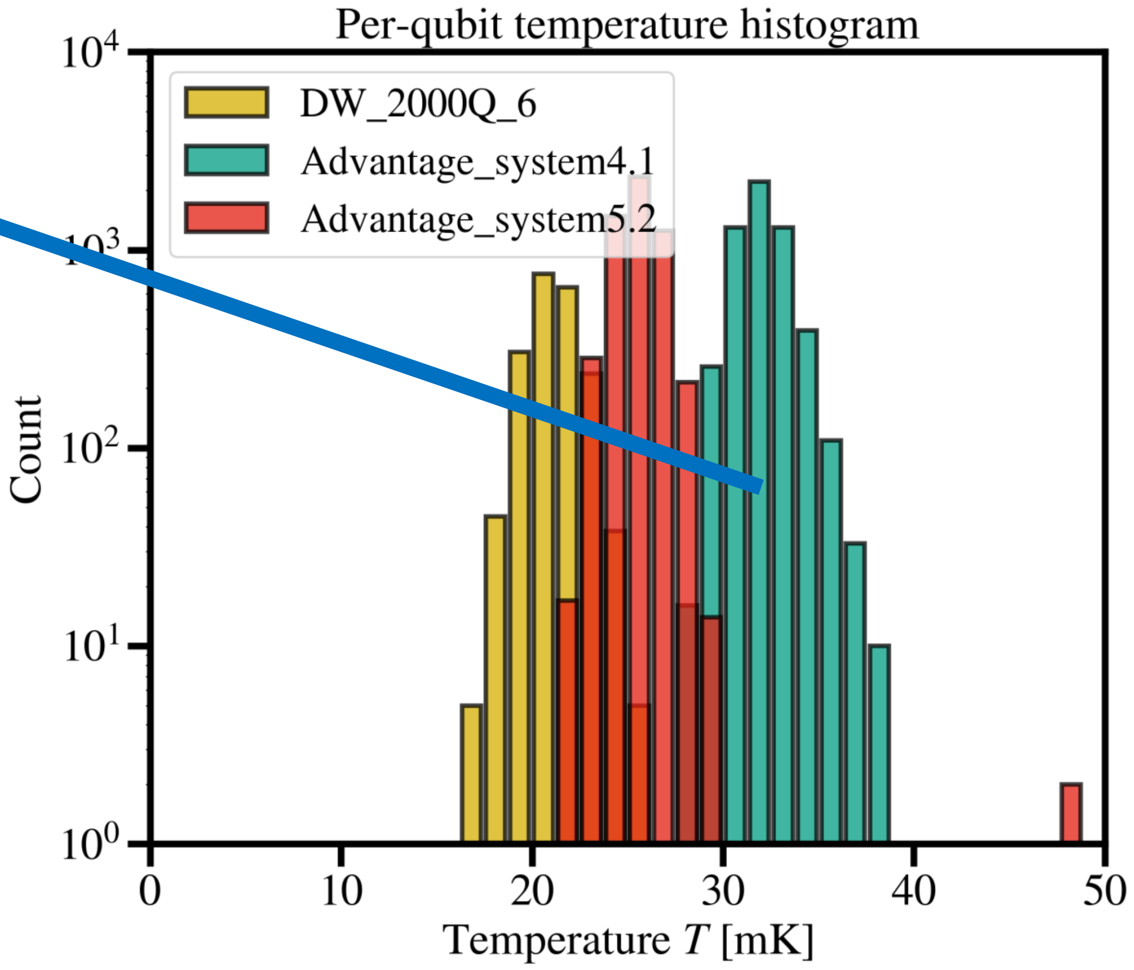


QUANTUM BOLTZMANN MACHINES

Experiment #1: Effective per-qubit temperature distribution

Advantage_system4.1



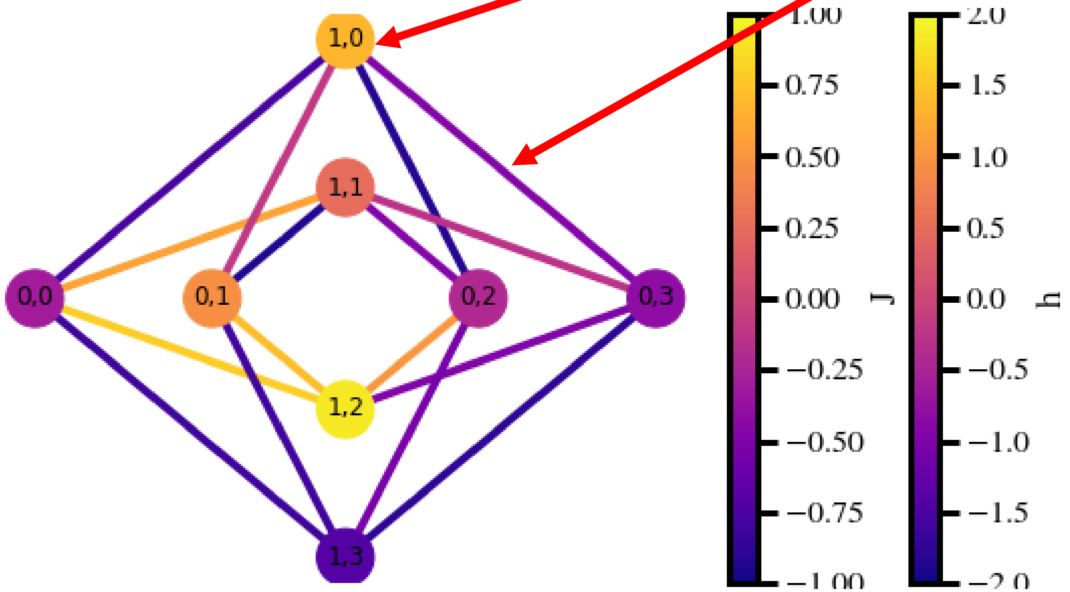


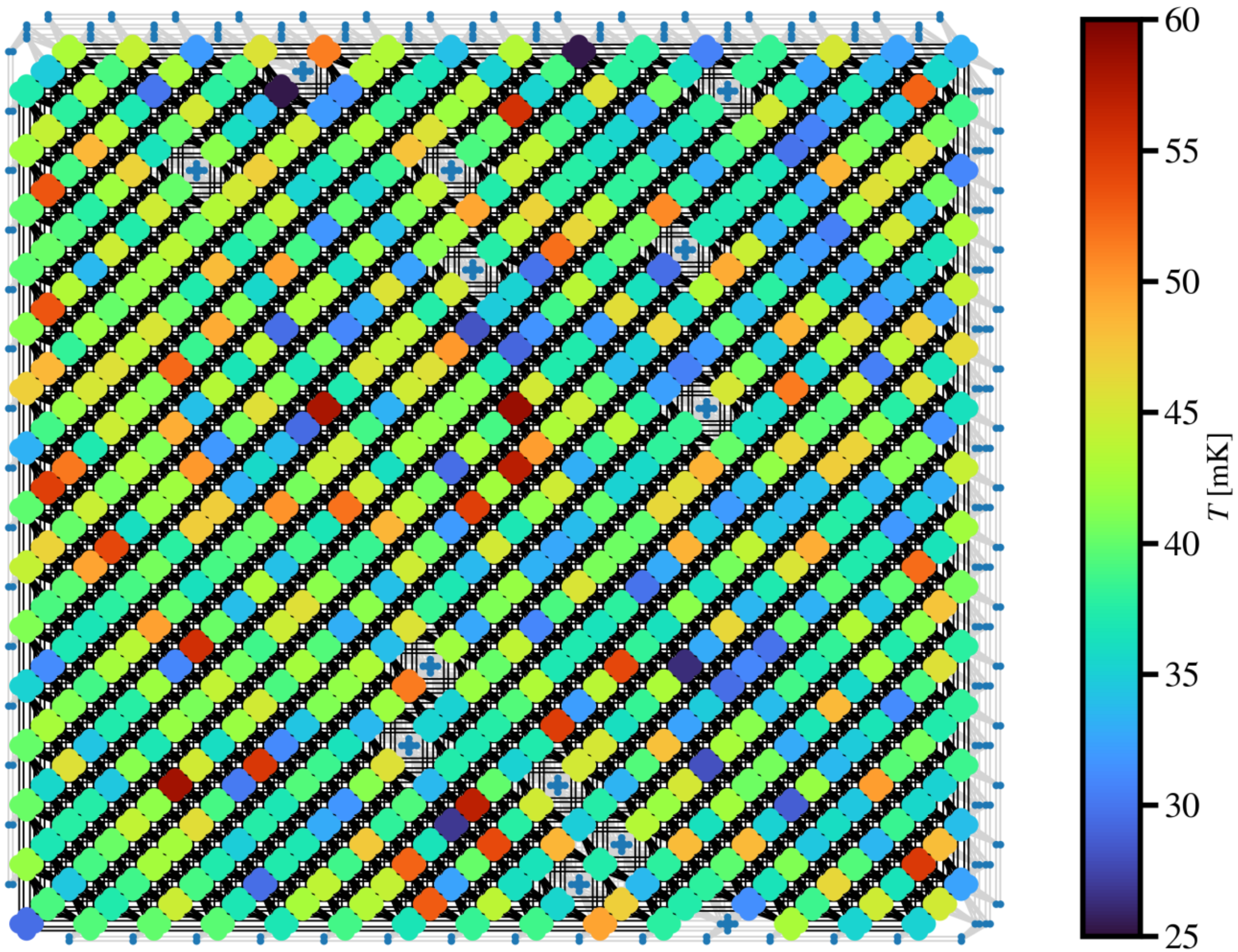
Perot et al. (2022)
https://jugit.fz-juelich.de/qip/qbm-quant-finance

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 $\beta = 1/k_B T$:

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

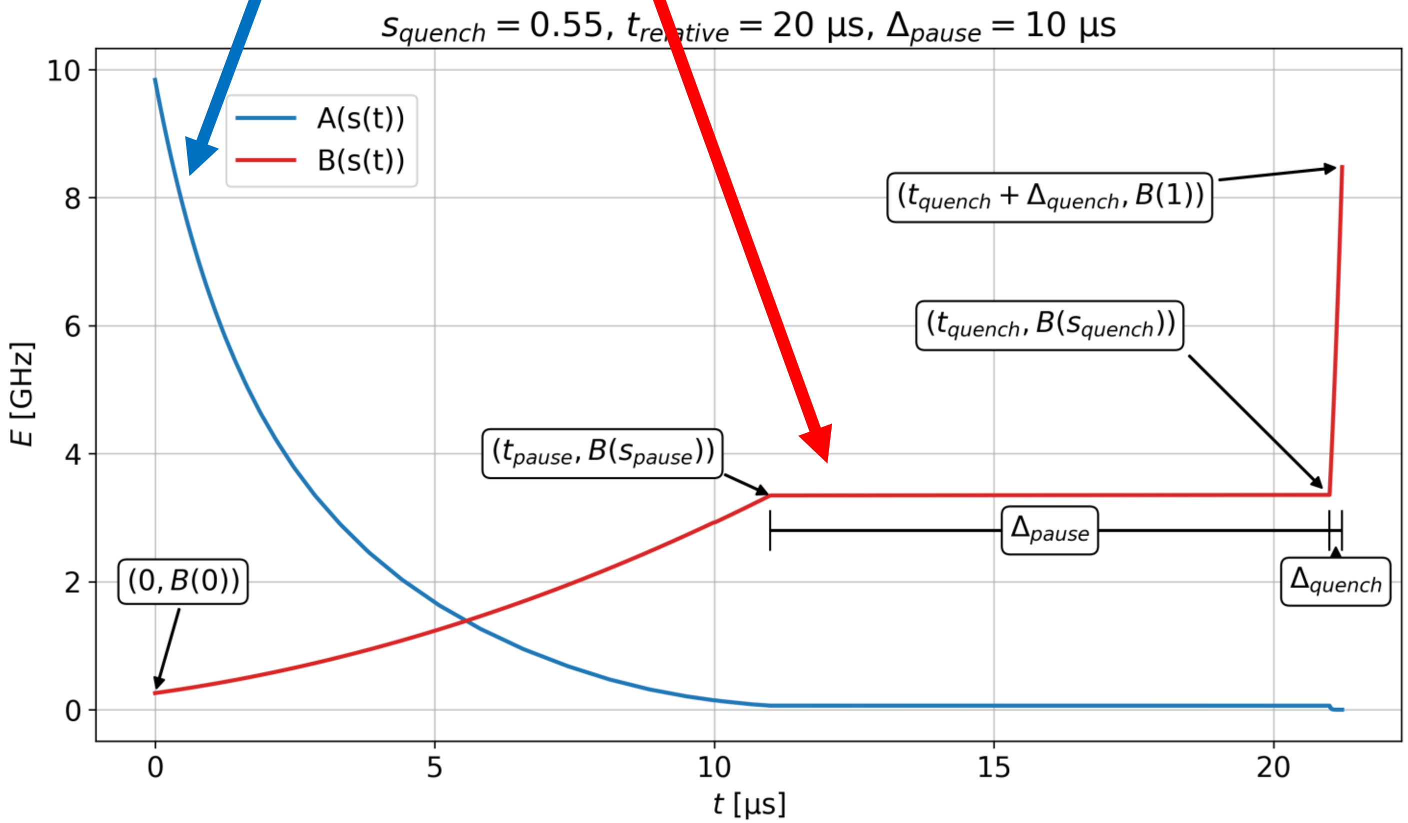
QUANTUM BOLTZMANN MACHINES

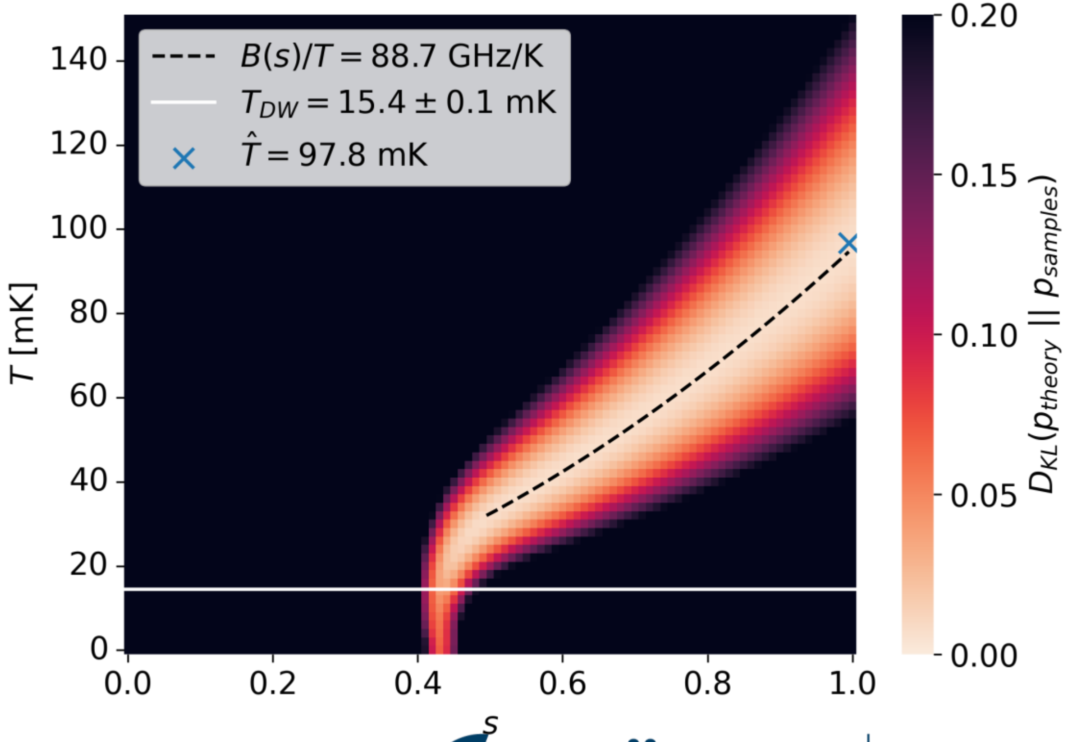
Experiment #3: 12-qubit problem (also for intermediate times s != 1)

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

Quantum Boltzmann distribution
at inverse temperature β and time s:

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



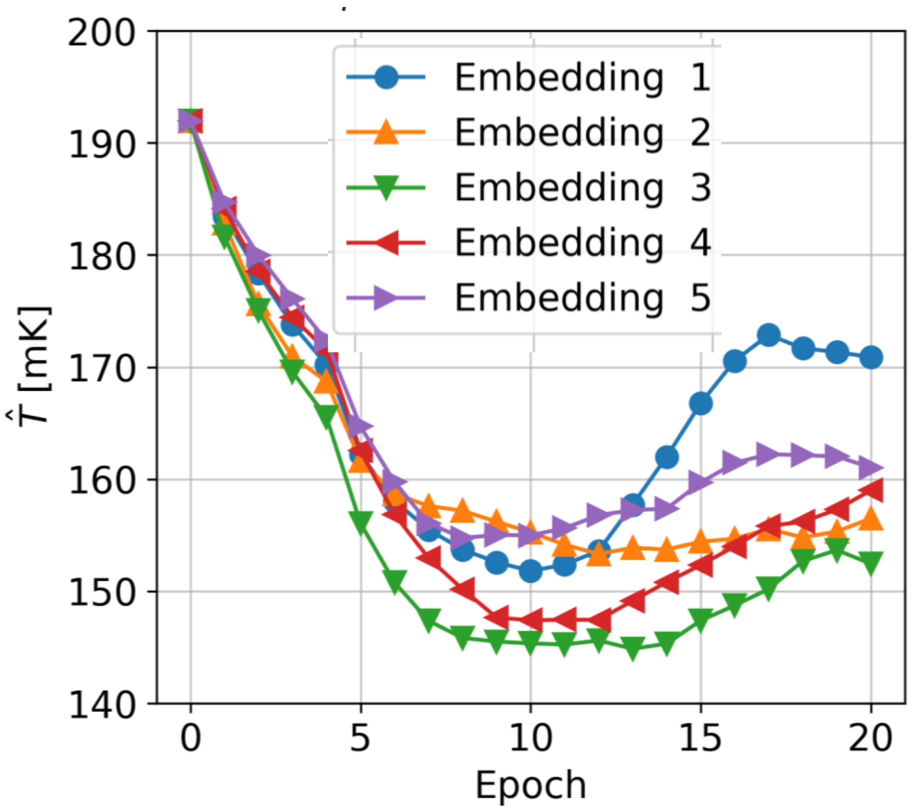


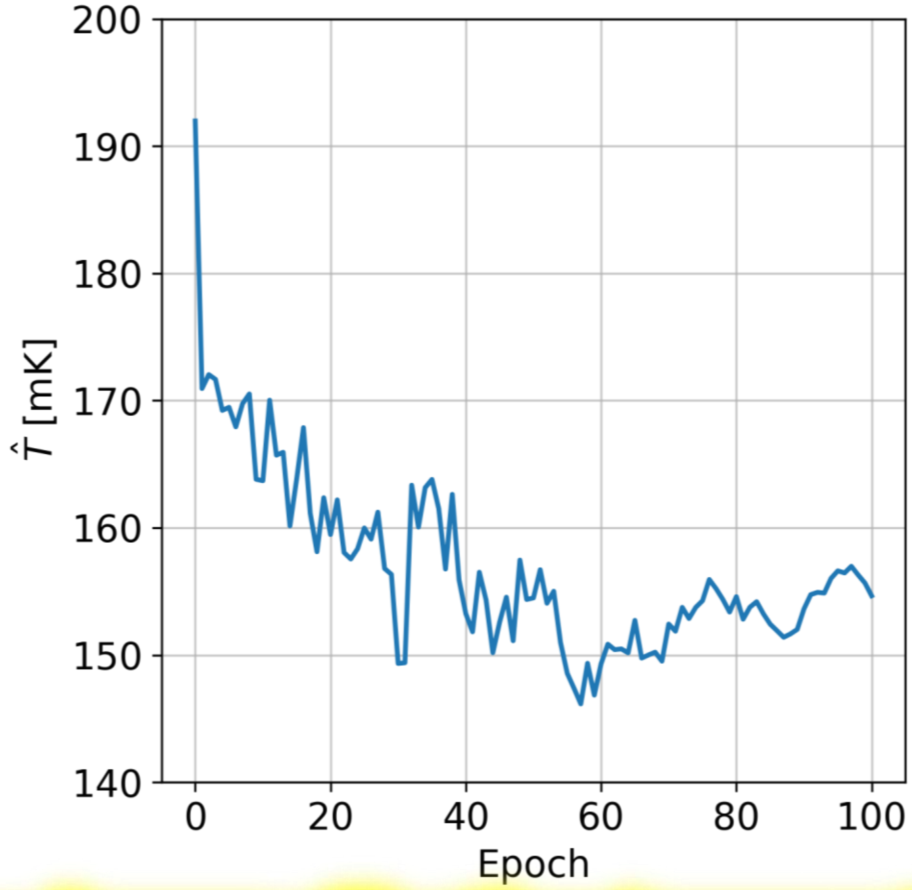
JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY

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

QUANTUM BOLTZMANN MACHINES

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





→ Observation: Larger problems ~ larger effective temperature

SUMMARY

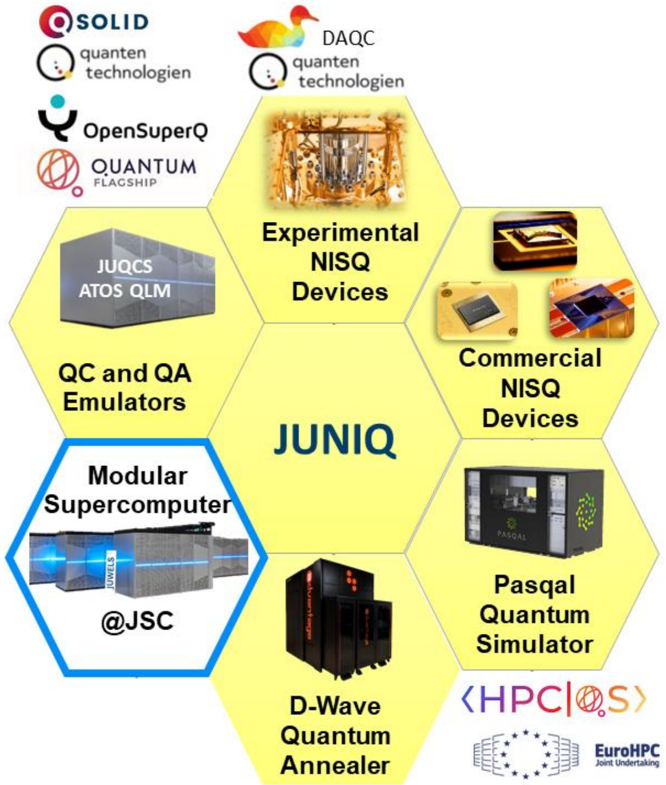
1. **Airline Scheduling**
2. **Traveling Salesman**
3. **Garden Optimization**
4. **2-Satisfiability**
5. **QSVM**
6. **QBM**

optimization problem
with constraints
[increasing complexity]

constraints only

optimization only [basically]

sampling problem



THANK YOU FOR YOUR ATTENTION

JUNIQ Rolling Call:

or search for
"FZJ JUNIQ ACCESS"

JÜLICH Forschungszentrum | JUNIQ QUANTUM USER FACILITY