How quantum annealing solves problems

AQC 2022

Nicholas Chancellor

June, 24 2022

Relevant UK projects for this audience

Collaborative computational project on quantum computing (CCP-QC)

- Work with other CCPs (academic projects) to find uses fro quantum computing within scientific research
- Idea is to use quantum computing to solve hard problems which come up in academic research rather than industry
- **https://ccp-qc.ac.uk/**

Quantum Enhanced and Verified Exascale Computing (QEVEC)

- Work on how quantum coprocessors can (eventually) support exascale computing
- Multiple projects looking at a variety of applications
- **https://excalibur.ac.uk/projects/qevec/**

Contact Viv Kendon at **viv.kendon@strath.ac.uk** if you are interested in potential collaborations

What this talk is about (+ collaborator acknowledgments)

Work I have done toward understanding how quantum annealing solves problems in increasingly realistic settings

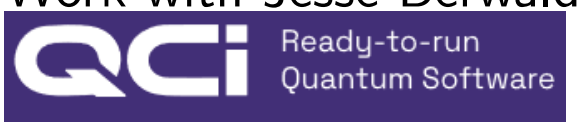
1. How do we understand anneals far from the adiabatic limit? (and even discontinuous)
   - Work with Adam Callison, Max Festenstein, Jie Chen, Laurentiu Nita, and Viv Kendon

2. How does noise effect search range in dissipation-driven annealing?
   - Work with Viv Kendon

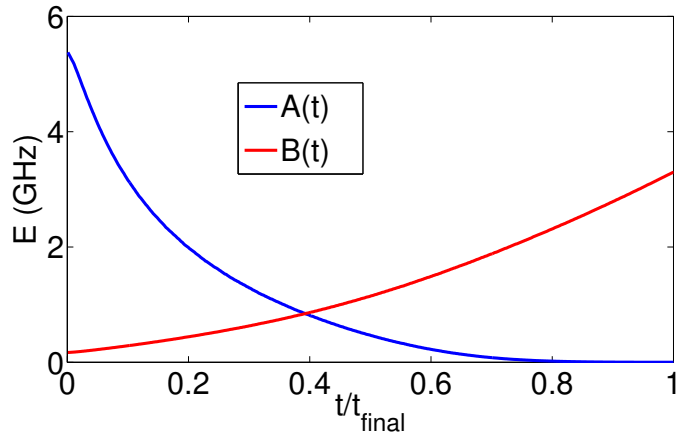
3. How does the encoding of optimisation problems affect dynamics?
   - Work with Jesse Berwald and Raouf Dridi

Adiabatic quantum computing

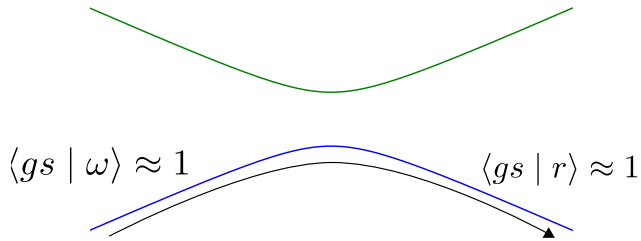
Traditional picture:

- Map an NP-hard optimization problem to a Hamiltonian, unknown ground state is solution
- *Slowly* change from a (driver) Hamiltonian with an easily prepared ground state to problem Hamiltonian
- Adiabatic theorem of quantum mechanics $\rightarrow$ success probability arbitrarily close to 100 % by running long enough



$$H(t) = A(t)H_{\text{driver}} + B(t)H_{\text{problem}}$$

Advantages and disadvantages of this picture



Theoretically satisfying

- Algorithm is effectively deterministic $\rightarrow$ "always" succeeds
- Intuitive picture involving only ground and first excited state

Let's assume P$\neq$NP

- Algorithm succeeds roughly 100% of the time
- Total runtime needs to be exponential in size of problem $\rightarrow$ system needs to remain coherent for exponentially long time*

*There are ways to apply more sophisticated adiabatic theorem to faster quenches in some cases (see: Crosson and Lidar, Nature Reviews Physics volume 3, pages 466-489 (2021)), but that isn't the topic of this talk

What can be done?

Restore coherence somehow

- Error correction, difficult to do in continuous time, but progress being made
- Low temperature dissipation can restore coherence $\rightarrow$ would have to be very low temperature

- Have to mitigate *all* errors for a *very* long time
- Not the subject of this talk

Succeed with low probability

- *Total* runtime is still exponential in problem size
- Each run is short $\rightarrow$ exponentially many needed to hit right answer

- Exponentially low success each run is *conceptually* unsatisfying...
- ... but much less demanding for coherence

Lottery

image public domain from wikimedia commons

Rapid quenches?

Energy conservation argument extended to any monotonic (closed system) quench

$$H(t) = A(t)\,H_{\text{drive}} + B(t)\,H_{\text{problem}} \quad \frac{A(t)}{B(t)} \geq \frac{A(t+\delta t)}{B(t+\delta t)} \forall_t$$

Sketch of proof:

1. Trotterize time evolution: $A(t) \to A(t+\delta t)$ and $B(t) \to B(t+\delta t)$ and apply $|\psi(t+\delta t)\rangle = \exp(-iH(t)\delta t)|\psi(t)\rangle$ in separate steps
2. Rescale time so that Hamiltonian always resembles (energy conserving) quantum walk $H_{\mathit{eff}}(\Gamma(t)) = \Gamma(t)\,H_{\text{drive}} + H_{\text{problem}}$
3. In rescaled version $\Gamma(t) \geq \Gamma(t+\delta t)$ (lowest $\langle H_{\text{drive}}\rangle$ is $-n$) $\therefore$ $\langle H_{\mathit{eff}}(\Gamma(t))\rangle_{\psi(t)} - \Gamma(t)\,n \geq \langle H_{\mathit{eff}}(\Gamma(t+\delta t))\rangle_{\psi(t)} - \Gamma(t+\delta t)\,n$
4. Because $\langle H_{\mathit{eff}}(\Gamma(t))\rangle_{\psi(t)} \geq -\Gamma(t)\,n \ \forall_t$, $\langle H_{\text{problem}}\rangle_{\psi(t)} \leq 0 \ \forall_t$

Details can be found in Callison et. al. PRX Quantum 2, 010338

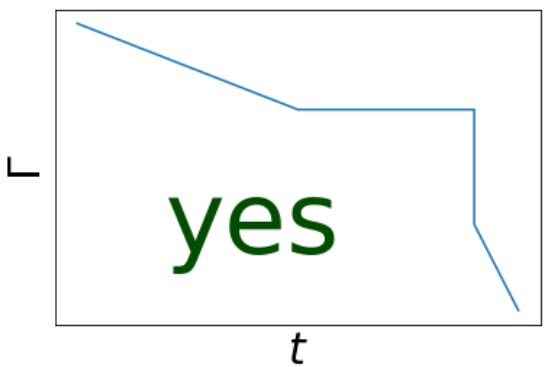
A very general result!

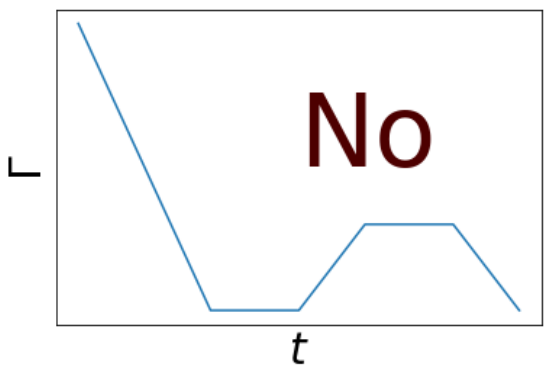
For result to hold (to be better than random guessing on average):

1. Monotonic $\Gamma(t) \geq \Gamma(t + \delta t)$ where $\Gamma(t) = \frac{A(t)}{B(t)}$
2. Start in ground state of H_{drive}
3. Driver not gapless $\rightarrow$ not a concern for real problems

What is allowed:

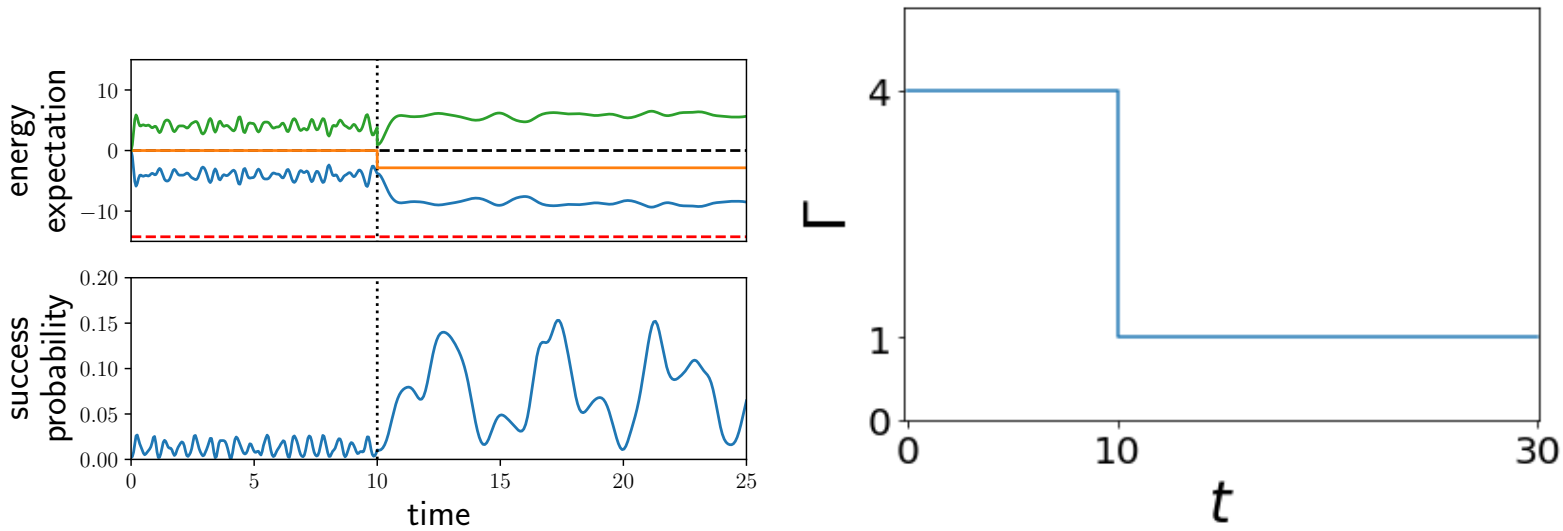
1. No limit on how fast algorithm runs
2. Discontinuities in $\Gamma(t)$ are ok
3. H_{drive} does not need to be diagonal in an orthogonal basis to H_{problem} $\rightarrow$ starting state can be biased





Intuitive example: two stage quantum walk

Perform a quantum walk at γ_1, and than use result as an input state for a second walk at $\gamma_2 < \gamma_1$



- Energy expectations: Green= $\gamma_{1,2}\langle H_{\text{drive}}\rangle$; Blue= $\langle H_{\text{problem}}\rangle$; Gold= $\gamma_{1,2}\langle H_d\rangle + \langle H_{\text{problem}}\rangle$
- Total energy conserved except for at dashed line where γ decreases
- Non-instantaneous quench effectively infinite stage quantum walk

Why is the rapid quench result important?

General, but rather weak:

Any monotonic quench at least as good as measuring the initial state

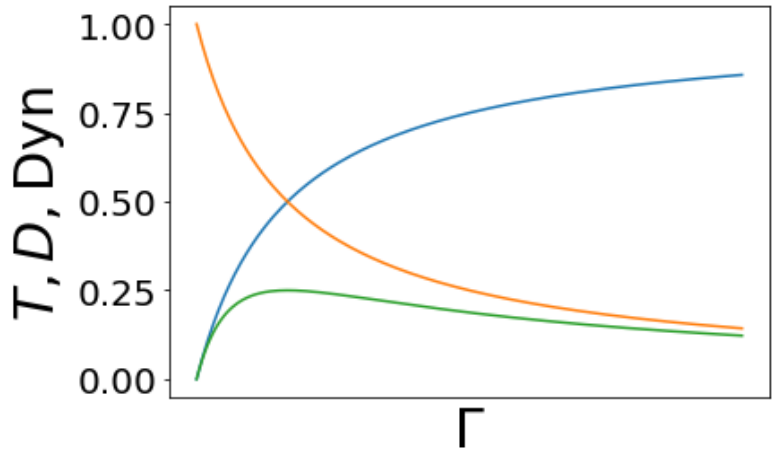
1. Design protocols to maximize dynamics $\rightarrow$ don't need to worry about dynamics being counter-productive
2. A **biased** search can already start from a very good guess
   more discussion on this later
3. Mechanism to understand dynamics very far from adiabatic limit

Quantifying dynamics in a two state subspace

Transfer coefficent, transfer between computational basis states: $T^{(jk)} = \frac{2\Gamma(t)|\langle k|H_{\text{drive}}|j\rangle|}{2\Gamma(t)|\langle k|H_{\text{drive}}|j\rangle|+|\Delta_{jk}|}$ (where Δ_{jk} is the total difference in diagonal matrix elements) Blue in figure

Disequilibrium coefficient, amount which H_{problem} breaks the initial equilibrium:
$D^{(jk)}$, defined the same as $T^{(jk)}$, but in the diagonal basis of H_{drive} rather than the computational basis Gold in figure

$\chi^{(jk)} = T^{(jk)}\, D^{(jk)}$ quantifies total dynamics Green in figure

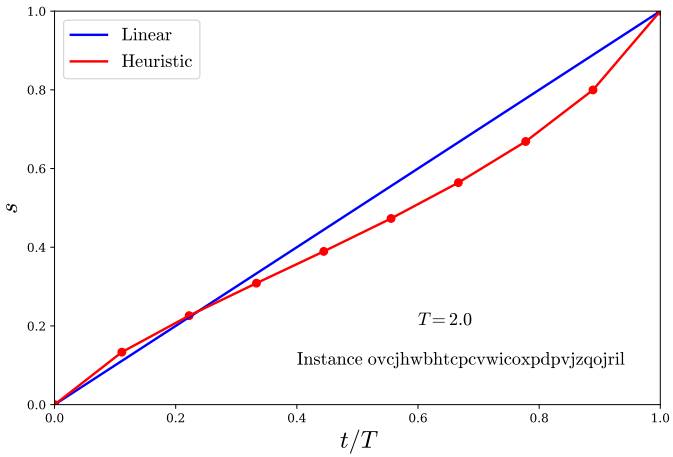


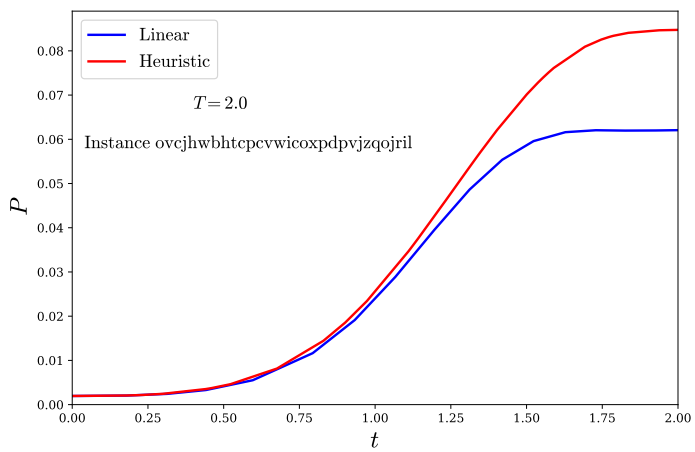
Unlike spectral gap, these quantities can be efficiently calculated for large problems!

Finding optimal annealing schedules

$$H(t) = A(t)\, H_{\text{drive}} + B(t)\, H_{\text{problem}}$$

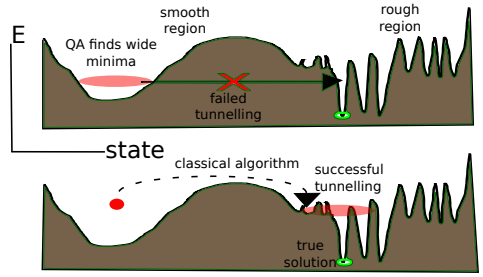
- Define $A(t) = (1 - s(t))$ and $B(t) = s(t)$
- Set $\frac{\partial s}{\partial t} \propto \frac{1}{\chi}$
- Compare to linear schedule $s \propto t$ for single SK instance





Heuristic performs better than linear schedule!

Hybrid protocols using this mechanism?



Known techniques:

Dissipative reverse annealing NC 2017 New J. Phys. 19 023024 as implemented on D-Wave devices

Relies on dissipation, not suitable for coherent algorithms

Coherent reverse annealing Perdomo-Ortiz et. al. Quantum Inf Process (2011) 10: 33. doi:10.1007/s11128-010-0168-z

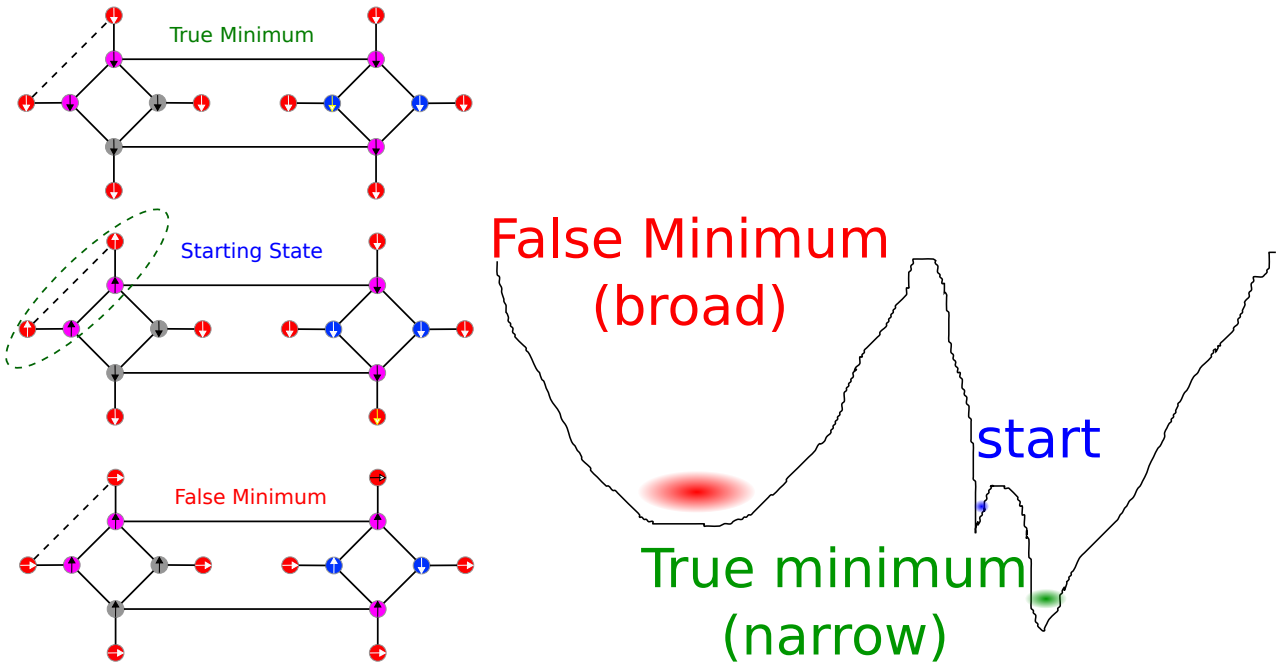
Involves three separate Hamiltonians, not compatible with rapid sweep proof in Callison et. al. PRX Quantum 2, 010338

Biased driver Hamiltonian Chinese Physics Letters, 30 **1** 010302 and Tobias Graß Phys. Rev. Lett. 123, 120501 (2019)

Compatible with proof in Callison et. al. PRX Quantum 2, 010338, can apply the mechanisms discussed here

Solving problems in a dissipative setting

Ising model gives us rich control to design energy landscapes, compare behaviour of D-Wave quantum annealers with different noise levels

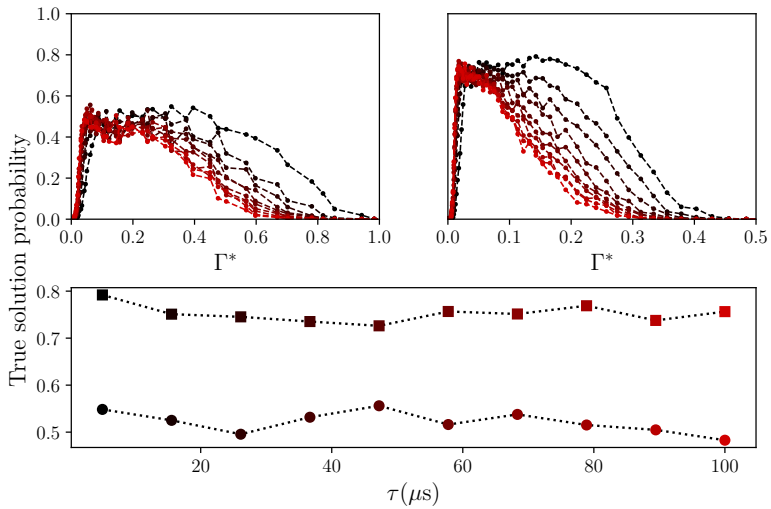


Core idea: starting state near a true minima, and further away from a false minima which would 'trick' forward annealing
dashed coupling J_t controls barrier between start and true minimum

Details in Chancellor and Kendon PRA 104, 012604 (2021)

Need more fluctuations on less noisy QPU

- Dissipation mediates reverse annealing local search, less coupling to bath $\rightarrow$ energy dissipates more slowly
- For the same runtime more fluctuations (higher Γ) is needed
- Suggests that spin bath polarisation is not the dominant effect here, otherwise noisier version would need higher Γ

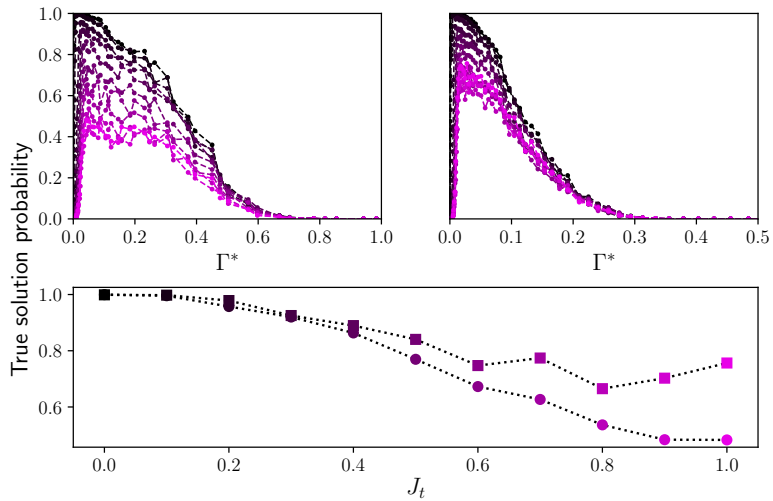


Increasing hold time τ allows more tunnelling out of the true minimum at higher Γ (left: lower noise, right: higher)
doesn't affect the height of the peak (important for modelling)

More exciting difference: higher peak with noisier QPU

Why is this exciting?

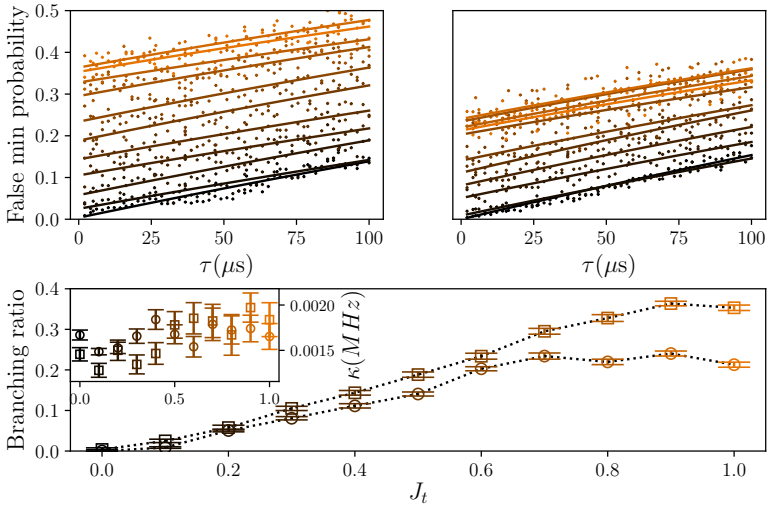
False minimum is further away, more tunnelling from start state means longer range search with lower noise



- Top left: lower noise true min probability, Right: higher, bottom, peak values, bottom, peak value versus J_t
- As it gets harder to tunnel to the true minimum the difference between the higher and lower noise QPUs grows

A simple model: initial branching then tunnelling

$$P_{\text{false}}(\tau) = 1 - [1 - R_{\text{false}}] \exp(-\kappa\tau)$$



- Branching ratio R_{false} indeed higher on lower noise QPU(left fit, squares on bottom)
- Don't have time for full details, see Chancellor and Kendon PRA 104, 012604 (2021)
- Confirms model, lowering noise causes more branching to further local minima $\rightarrow$ longer range search

Effect of problem structure and encoding*

Ready-to-run Quantum Software

Consider higher-than-binary discrete problems; appear often in real world optimisation, for example:

- A truck can go down any of three roads...
- A tasks can be scheduled at any of five times...
- A component can be placed any of seven places on a chip...

- Define two index objects:

$$x_{i,\alpha} = \begin{cases} 1 & \text{variable } i \text{ takes value } \alpha \\ 0 & \text{otherwise} \end{cases}$$

- Discrete Quadratic models, (DQM), made from pairwise interactions of x terms:

$$H_{\mathrm{DQM}} = \sum_{i,j} \sum_{\alpha,\beta} D_{(i,j,\alpha,\beta)} x_{i,\alpha} x_{j,\beta}$$

*Details in arχiv:2108.12004

Discrete variables into binary, three ways

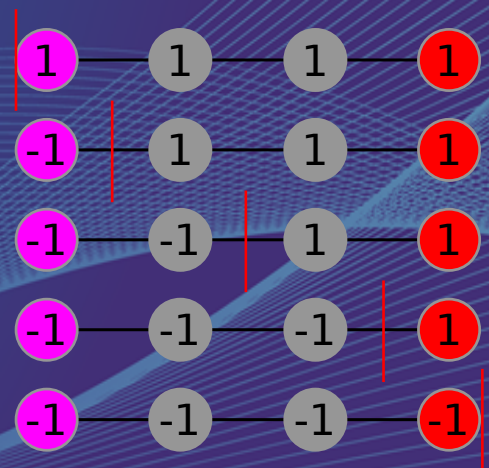
Ready-to-run Quantum Software

Variable size=m

performance metric	binary	one-hot	domain wall*
# binary variables	$\lceil \log_2(m) \rceil$	m	$m-1$
# couplers for encoding	0 if $m = 2^n$ $n \in \mathbb{Z}$ complicated otherwise	$m(m-1)$	$m-2$
intra-variable connectivity	N/A or complicated	complete	linear
maximum order needed for two variable interactions	$2\lceil \log_2(m) \rceil$	2	2

Binary= assign bitstrings to configurations
One hot= constrain variables so exactly one can be 1
Domain wall= new encoding w/ better performance

encoded value	qubit configuration
0	1111
1	-1111
2	-1-111
3	-1-1-11
4	-1-1-1-1



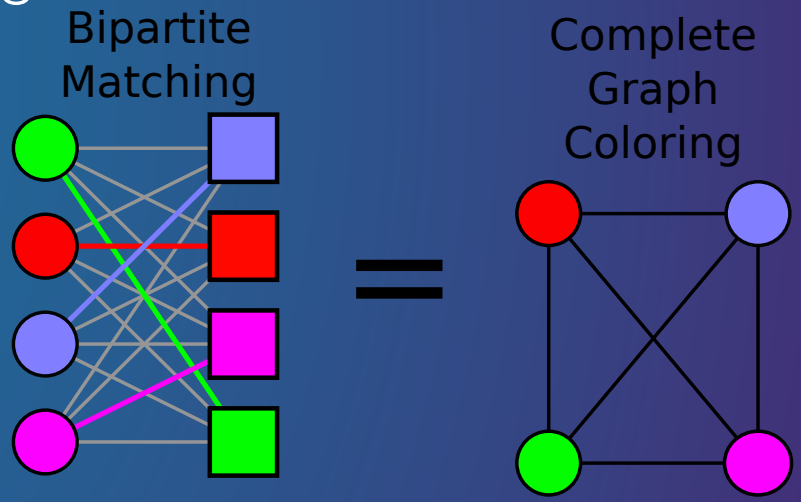
*For details see: Chancellor, Quantum Sci. Technol. 4 045004

*Chen et. al. IEEE Transactions on Quantum Engineering 3102714 (2021)

Quadratic Assignment Problem (QAP)

Ready-to-run Quantum Software

Assign m facilities to m locations such that a single facility is only assigned to one location and vice-versa



- General (hard) version $\rightarrow$ pairs of assignments are weighted, we use unweighted $\rightarrow$ **not** hard, but symmetry and large degeneracy useful for analysis
- Can be thought of as a colouring problem on an m-node fully connected graph
- $m!$-fold degenerate ground state

Aside, improved performance on max three colouring*

Green=statistically significant result (95% confidence)

	Adv. dw/oh		2000Q dw/oh		dw Adv./2000Q		oh Adv./2000Q		(dw, Adv.)/(oh, 2000Q)		(dw, 2000Q)/(oh, Adv.)	
5 node (b,w)	0	0	0	0	0	0	0	0	0	0	0	0
5 node p												
10 node (b,w)	42	0	37	0	2	0	19	21	39	0	40	0
10 node p	2.27×10^{-13}		7.28×10^{-12}		2.50×10^{-1}		6.82×10^{-1}		1.82×10^{-12}		9.09×10^{-13}	
15 node (b,w)	85	2	95	3	32	34	70	22	94	1	91	2
15 node p	2.47×10^{-23}		4.95×10^{-25}		6.44×10^{-1}		2.67×10^{-7}		2.42×10^{-27}		4.41×10^{-25}	
20 node (b,w)	99	0	100	0	43	41	94	3	100	0	93	2
20 node p	1.58×10^{-30}		7.89×10^{-31}		4.57×10^{-1}		9.60×10^{-25}		7.89×10^{-31}		1.15×10^{-25}	
25 node (b,w)	100	0		FAIL	66	20		FAIL		FAIL	98	2
25 node p	7.89×10^{-31}				3.33×10^{-7}						3.98×10^{-27}	
30 node (b,w)	100	0		FAIL	72	20		FAIL		FAIL	97	2
30 node p	7.89×10^{-31}				2.30×10^{-8}						7.81×10^{-27}	
35 node (b,w)	100	0	FAIL	FAIL		FAIL		FAIL		FAIL	FAIL	
35 node p	7.89×10^{-31}											
40 node(b,w)	100	0	FAIL	FAIL		FAIL		FAIL		FAIL	FAIL	
40 node p	7.89×10^{-31}											

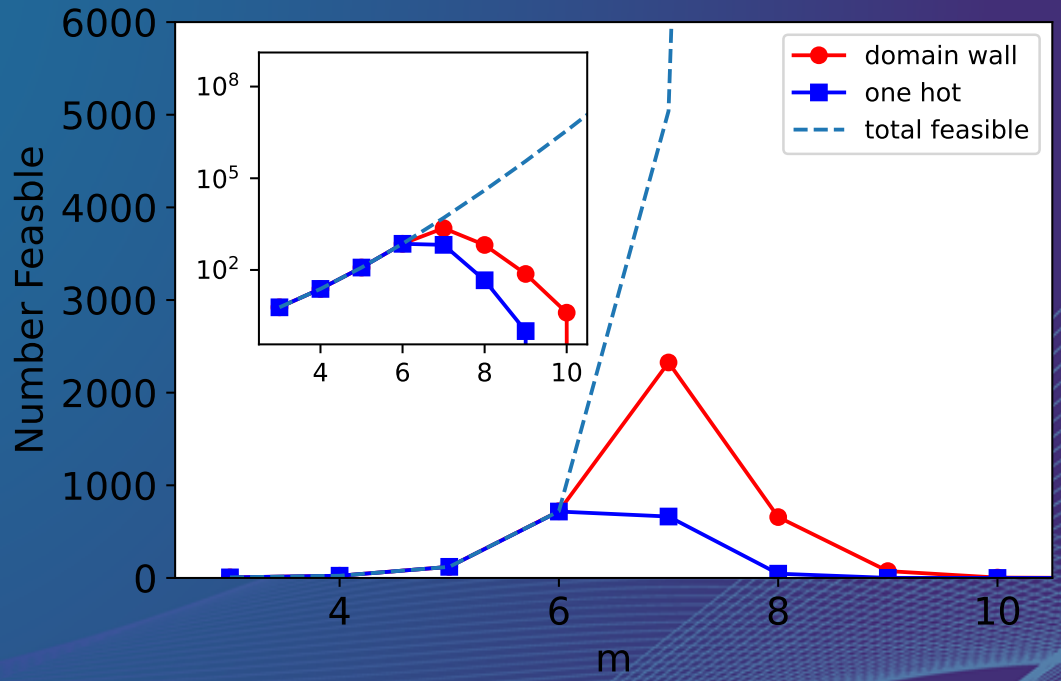
- Domain-wall on 2000Q beats one-hot on Advantage (100 total each size b=number better, w=number worse, p=statistical significance)
- Trend continue up to size where no longer possible to embed in 2000Q (FAIL), similar results for k-colouring (not shown)
- Worth trying if you have discrete problems to encode

*Chen et. al. IEEE Transactions on Quantum Engineering 3102714 (2021)

Experimental tests (unweighed assignment)

Ready-to-run Quantum Software

Run on D-Wave *Advantage* annealer 10 embeddings at each size with $10,000$ reads for total of $100,000$ reads at each size (default settings otherwise)

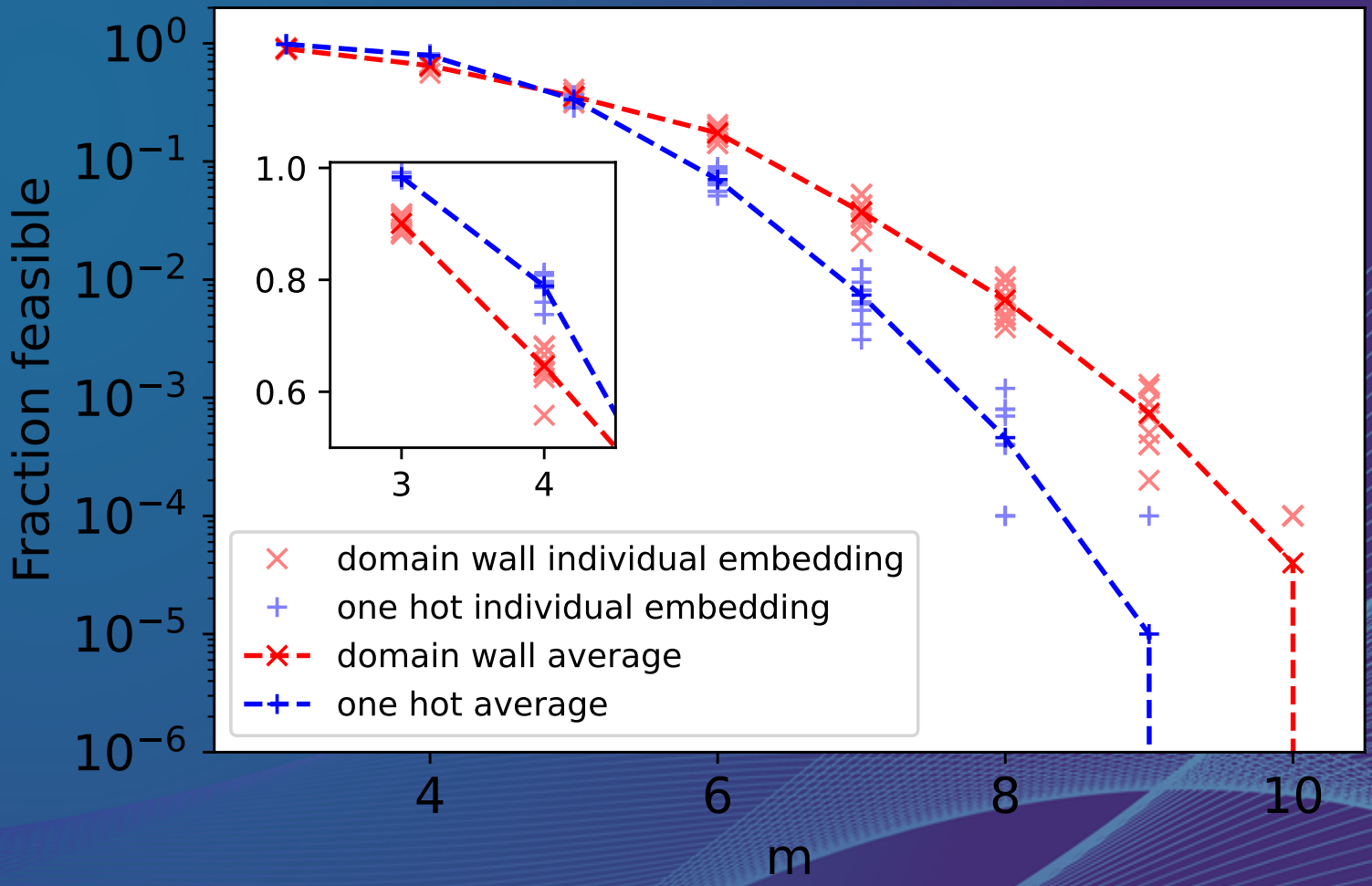


Able to find all feasibles up until about size 6, then both struggle, but domain-wall encoding performs much better.

Rate of feasible solutions

Ready-to-run Quantum Software

Stars represent fractions of returned solutions which are feasible

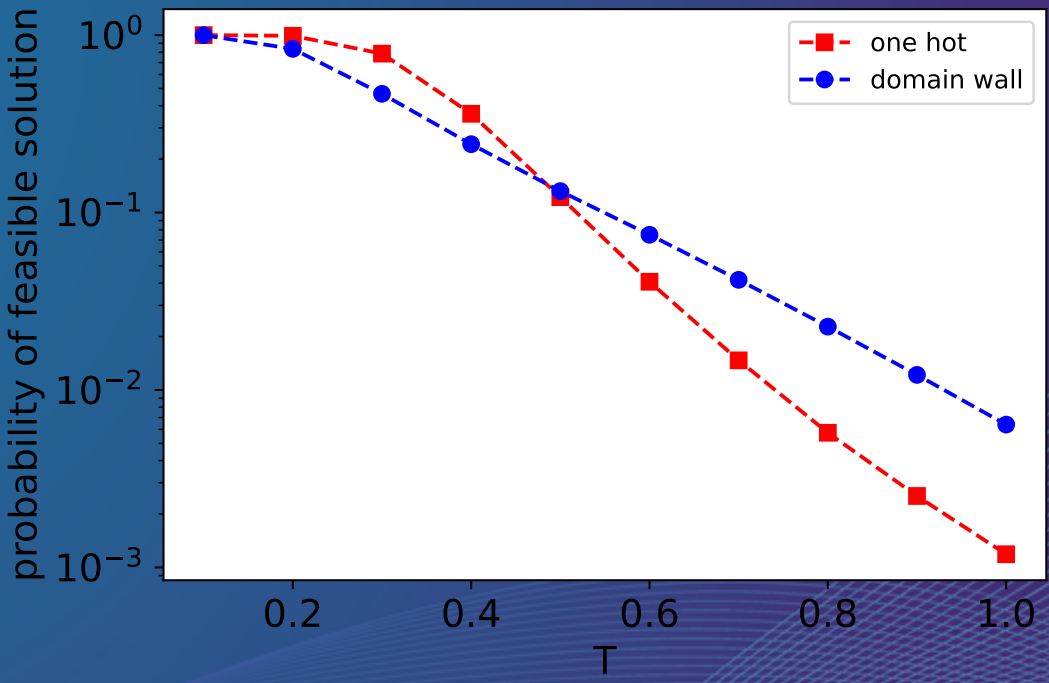


At largest size ($n = 10$) domain-wall encoding finds solutions while one-hot finds none.

One explanation: thermal excitations

Ready-to-run Quantum Software

Symmetry of problem means Metropolis algorithm converges quickly, efficient thermal sampling

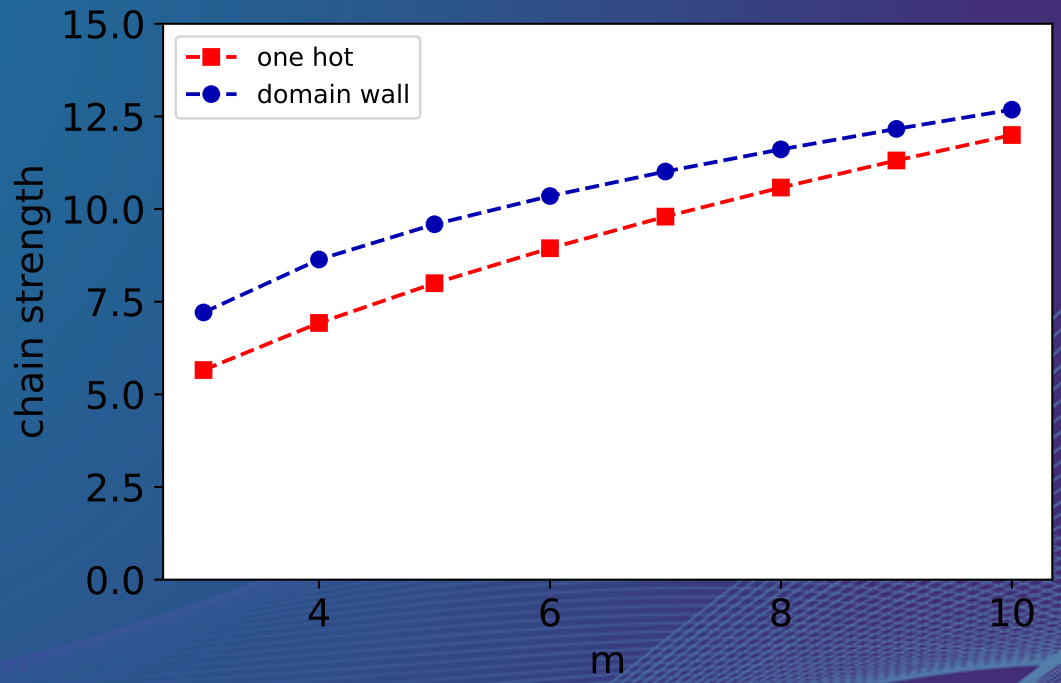


Probability of feasible solution is better at higher temperature with domain-wall encoding, makes sense one fewer qubit $\rightarrow$ smaller solution space

Dynamic range squeezing

Ready-to-run Quantum Software

Minor embedding chains need to be stronger for larger problems → less range left for problem, effectively higher temperature *

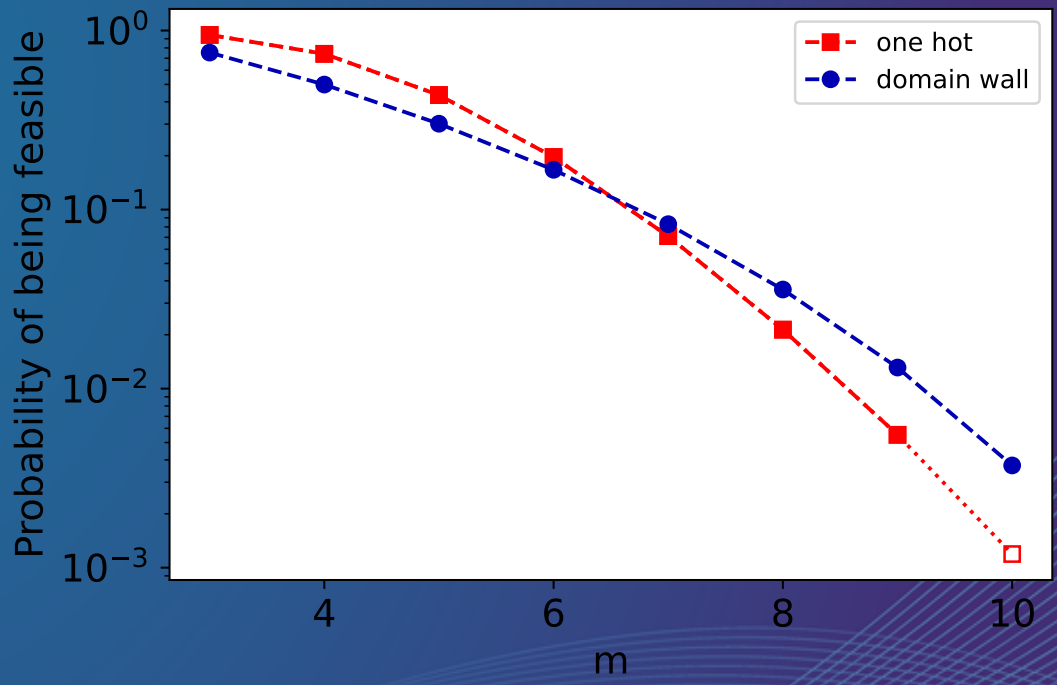


*we use default "uniform torque compensation" method

Thermal equilibrium model

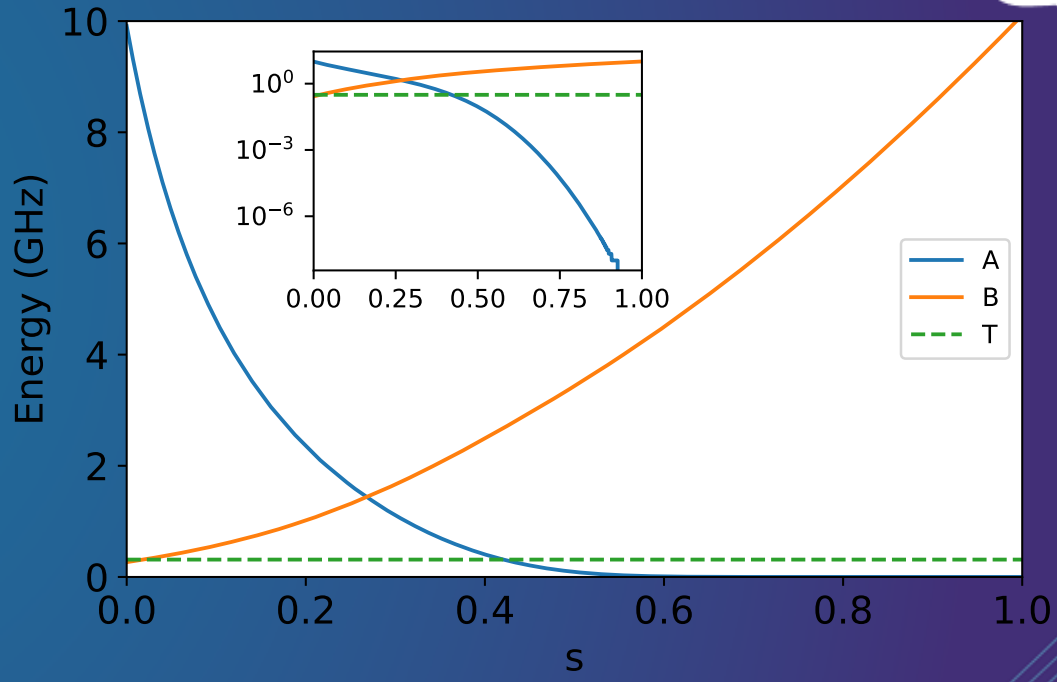
Ready-to-run Quantum Software

Assuming an energy scale of ≈ 5 GHz at the freezing point we find feasible probability for a purely thermal model



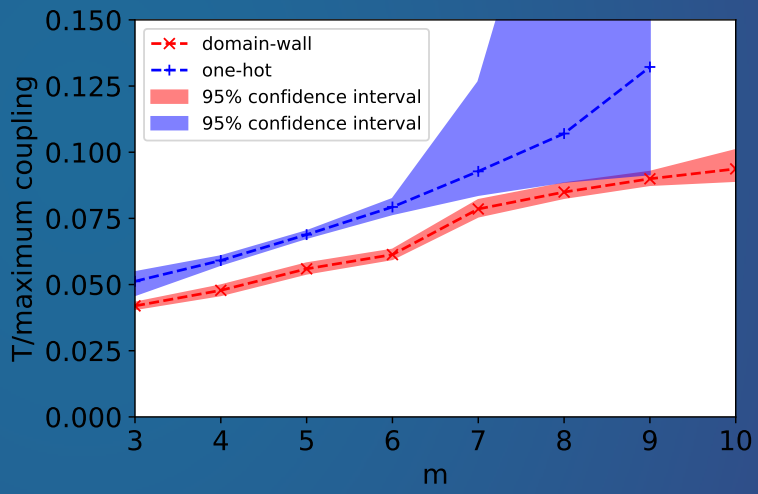
- Shows same crossover as real data
- Not in the same location, but...
  1. Estimate of energy scale is rough
  2. Not all sizes will freeze at the same time each will have different scales

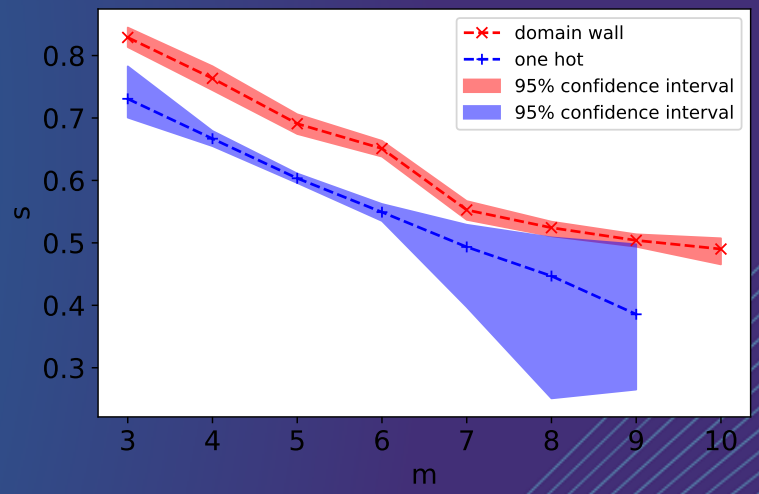
Estimate energy scale



1. Assume "frozen in" thermal distribution $\rightarrow$ Kibble-Zurek style approximation
2. Known physical temperature and experimental success probabilities
3. Back calculate energy scale (B) and therefore freeze point (s)
4. Verify that quantum fluctuations (A) can be safely ignored at freeze point

Effective temperature and freeze point



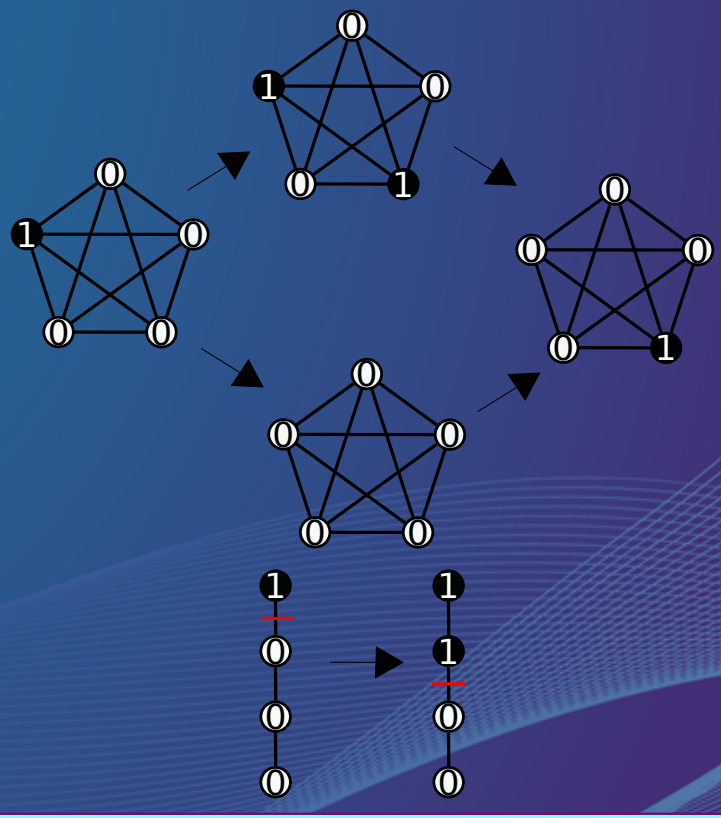


- Already taken into account embedding strength
- Domain-wall version effectively sampled at lower temperature $\leftrightarrow$ later freezing

Encoding has a strong effect on the dynamics of how the problem is solved

Why might this be true?

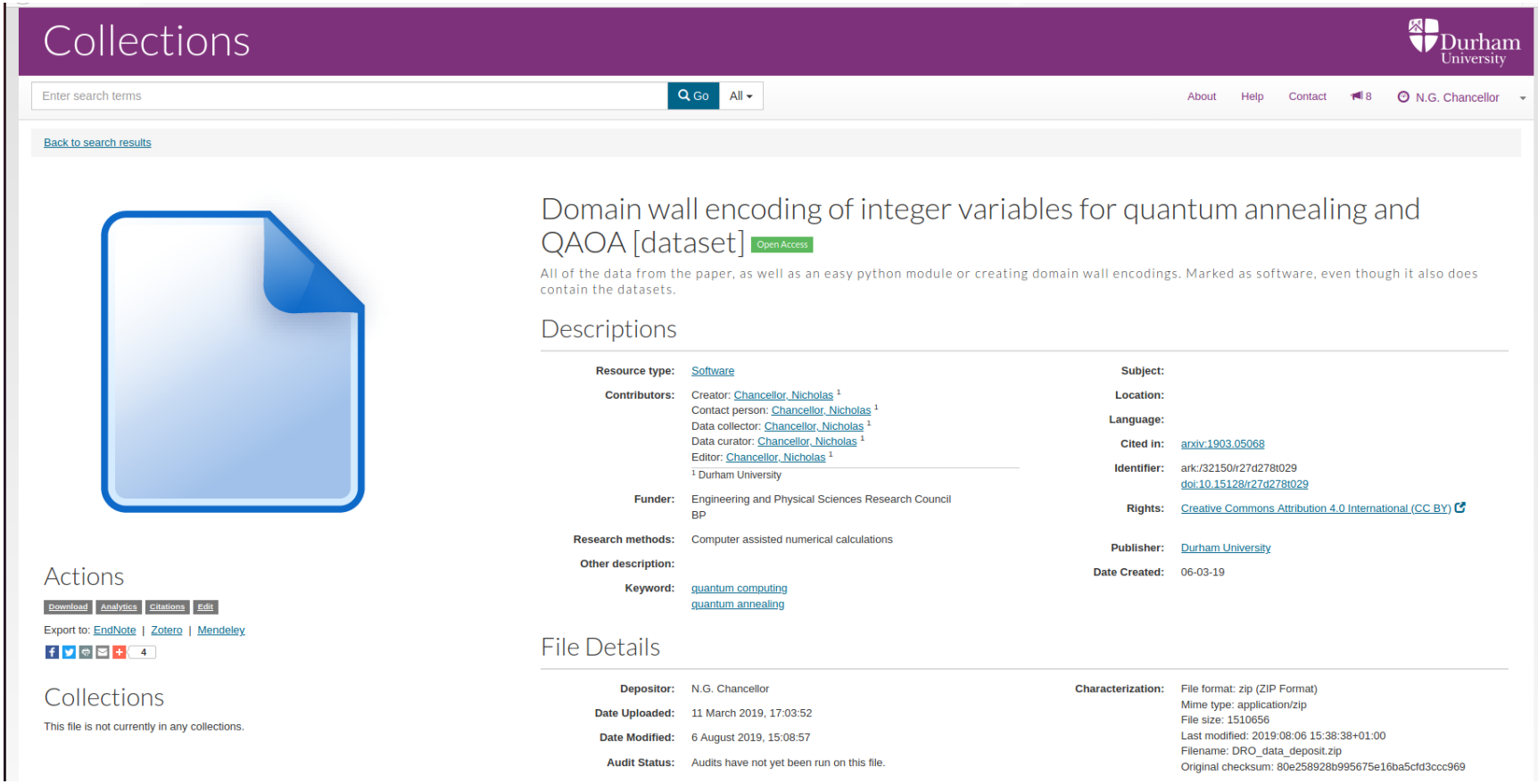
- One hot value cannot be changed by flipping a single binary variable
- Domain wall can therefore easier for transverse field to update



Need to consider underlying physics with encoding

Want to try it yourself?

Python code to create domain wall encodings available at https://collections.durham.ac.uk/: "Domain wall encoding of integer variables for quantum annealing and QAOA [dataset]"*



*https://doi.org/10.15128/r27d278t029

Key points

Non-traditional (i.e. not relying on adiabatic theorem) approaches to understanding quantum annealing in more realistic settings exist and are useful

Same structures which make quantum annealers versatile for solving problems allow for sophisticated energy landscapes for experiments

Reducing noise can and does lead to a longer range search of solution space

We need a better understanding of the interface between problem encoding and physical dynamics

Encoding can have a fundamental and dramatic effect on freeze time