# Quantum computing with spin systems

IPPP Quantum seminar

Nicholas Chancellor*

July, 2020

*for more about me see http://nicholas-chancellor.me and contained links

# My Background

Bachelor in Engineering Physics from Colorado School of Mines

PhD from University of Southern California in Physics

Post-doc at UCL performing remote experiments on D-Wave quantum annealers

Post-doc at Durham hybrid quantum/classical (Viv Kendon)

Awarded EPSRC UKRI Innovation fellowship in June 2018 - currently PI of own group, more on next slide

# My group

**Main funding: UKRI Innovation fellowship * June 2018-June 2021**

- Focus on applied quantum computing in the near term
- Theory, remote experiments, and use cases
- Hired PDRA (Jie Chen), to help with use cases $\rightarrow$ not a physicist, expert on queueing and network theory
- One graduate student (Laur Nita) + collaboration with Viv Kendon and her grad students + undergrad project students

**Other funding**

- EPSRC NQIT (quantum computing) hub project partnered with D-Wave Systems (with VK) + impact acceleration
- Co-I on HQCS (successor to NQIT) hub
- Quantum annealer machine time funded by BP

*see https://gtr.ukri.org/projects?ref=EP%2FS00114X%2F1

# Our work not discussed here (time constraints)

1. How to design quantum error correction codes without knowing quantum mechanics
   - IEEE transactions on information theory 66, 1, pp. 130-146 (2020) (see also: Quantum Sci. Technol. 3 035010 (2018), ar$\chi$iv:1903.10254 )
2. Mapping optimization problems to Ising models
   - Coupler proposal Nature Partner Journals Quantum Information 3, 21 (2017)
   - Max-k-SAT mapping Scientific Reports 6, 37107 (2016)
   - 'Domain wall' encoding for discrete variables Quantum Science and Technology 4, 045004 (2019)
3. Unstructured search with quantum walks and adiabatic algorithms
   - Atomic testbed proposal Phys. Rev. A 100, 032320 (2019)
   - Interpolation between algorithms Phys. Rev. A 99, 022339 (2019)
4. Many ongoing projects... including citizen science my graduate student spoke about at NQM (**Recently funded grant on this, I am PI**)

# Quantum computing

Big idea: harness the fundamental physics of discrete systems (quantum mechanics) to solve important problems

- We know it works in theory: quantum search of unstructured database with $N$ entries in a time proportional to $\sqrt{N}$
- This is not possible without using quantum mechanics (only option without QM is random guess or exhaustive search)

(011) (010)

(111)

(000)

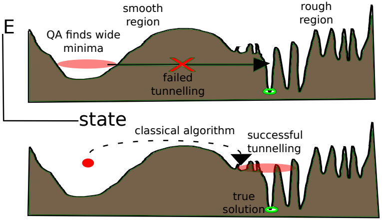
(001) (110)

...but how do we use real, imperfect, quantum machines to solve problems people care about?

# Applied Quantum computing

How do we use real, imperfect, quantum machines to solve problems people care about?

1. Only use them for what they are good at do the rest classically hybrid quantum/classical algorithms




2. Find the right problems → need to be the right shape and size for near term the machines... and still be problems people care about

But first... some background on continuous time QC

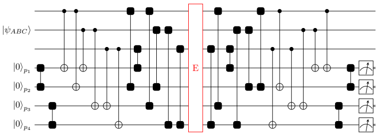
# Two different approaches to quantum computing

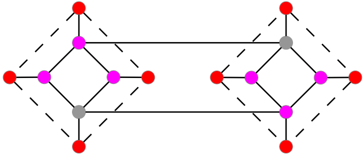
'Gate' based quantum computing

- Discrete quantum operations on qubits
- Construct 'circuits' out of these gates
- Detect and correct errors to reduce effect of noise

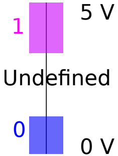
Continuous time

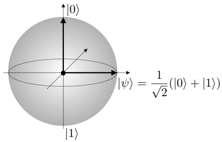
- Map problems directly to physical system
- Allow quantum physics to help search solution space
- Low temperature environment could help solve problems



# Why we focus on continuous time




Classical bits: fundamentally discrete $\rightarrow$ 0 or 1, nothing in between

Lends itself to a discrete *digital* description: bit flips either happen or they don't

Quantum bits: continuous rotations are possible

Breaking operations up into discrete chunks is not natural $\rightarrow$ an (exact) bit flip is just as hard as any other rotation

Bonus feature: applied gate based algorithms similar to continuous time operations $\rightarrow$ cont. time algorithms have implications for gate based

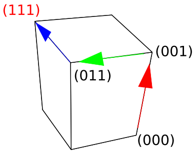
# Getting physics to solve hard problems $\rightarrow$ transverse field Ising model

Hamiltonian:

$$H = -A(t)\sum_i^n X_i + B(t)\left(\sum_i^n h_i Z_i + \sum_{i,j}^n J_{ij} Z_i Z_j\right)$$

What this means in pictures:

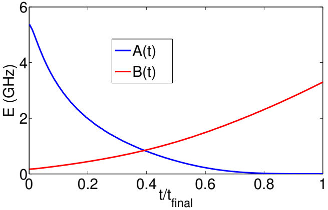
$\sum_i^n X_i \rightarrow$ Bit flips, hops state through $n$ dimensional hypercube




$\sum_i^n h_i Z_i + \sum_{i,j}^n J_{ij} Z_i Z_j \rightarrow$ Ising spin glass, defines interesting problem to be solved (as bitstring energies) can map hard optimisation problems to this (happy to explain if interest)

# Actually solving problems

Quantum Hamiltonians generalize classical Monte Carlo algorithms ex. simulated annealing




$$H = -A(t) \sum_{i}^{n} X_i + B(t) \left( \sum_{i}^{n} h_i Z_i + \sum_{i,j}^{n} J_{ij} Z_i Z_j \right)$$

- Parameter sweeps can be used to solve problems
- Low temperature dissipation can help too

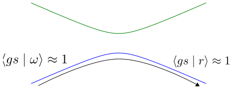
Adiabatic theorem of QM guarantees right answer if running slowly enough... but this is probably not the best approach

# Understanding protocols (closed system for simplicity)

Real machines are noisy, hard to keep coherent
Unless $P = NP^\star$ (when quantum machines included), all quantum algorithms for NP-hard problems must do one or both:

1. Protect from all noise for exponential time
2. Succeed with exponentially low probability

Most theory is in the adiabatic limit, succeeds with probability $\approx 1$ $\rightarrow$ remain coherent exponentially long, not practical

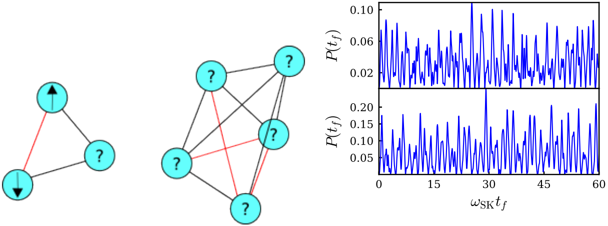



Need theory/numerics to understand experimentally achievable protocols $\rightarrow$ run many times with low probabilities

$^\star$Saying $P \neq NP$ is basically saying that hard optimisation problems exist, most computer science experts believe $P \neq NP$

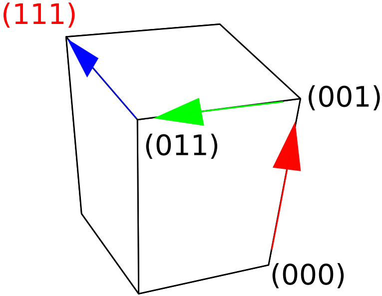
# Example: continuous time quantum walk on spin glass

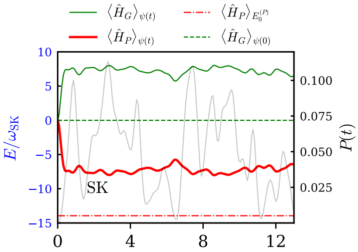
- Start with an equal positive superposition of all solutions, $|\omega\rangle = \frac{1}{\sqrt{N}} \sum_i |i\rangle$
  - Evolve with a fixed Hamiltonian $H_{\text{walk}} = \gamma H_{\text{hop}} + H_{\text{problem}}$
- $H_{\text{hop}} = -\sum_i \sigma_i^x \rightarrow$ superposition is ground state
- $H_{\text{problem}} = \sum_{i,j} J_{i,j} \sigma_i^z \sigma_j^z + \sum_i h_i \sigma_i^z$ where $h_i$ and $J_{i,j}$ drawn from the same Gaussian distribution
- Measure after random short period of time, repeat many times




See Callison et. al. 2019 New J. Phys. **21** 123022 for details, work with Adam Callison, Viv Kendon, and Florian Mintert

# How is this a 'walk'? How does it find solutions?




- $H_{\mathrm{hop}}$ effectively forms a hypercube with a bitstring at each vertex, probability amplitude 'walks' between different states
- $H_{\mathrm{problem}}$ contributes phases which guide the walk

Energy is conserved $\langle H_{\mathrm{walk}} \rangle_{t=0} = \langle H_{\mathrm{walk}} \rangle_{t>0}$ since the system starts in the ground state of $H_{\mathrm{hop}}$:
$\langle H_{\mathrm{problem}} \rangle_{t>0} - \langle H_{\mathrm{problem}} \rangle_{t=0} = \langle H_{\mathrm{hop}} \rangle_{t=0} - \langle H_{\mathrm{hop}} \rangle_{t>0} \leq 0$

Walk seeks out 'good' solutions!

# Rapid quenches?

The energy conservation argument given previously can be extended to any monotonic (closed system) quench

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

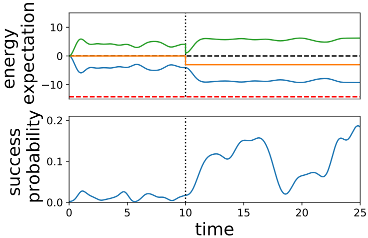
Sketch of proof:

1. Trotterize time evolution: $A(t) \rightarrow A(t+\delta t)$ and $B(t) \rightarrow B(t+\delta t)$ 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 quantum walk $H_{eff}(\gamma(t)) = \gamma(t)\, H_d + H_{\text{problem}}$
3. In rescaled version $\gamma(t) \geq \gamma(t+\delta t) \therefore$ $\langle H_{eff}(\gamma(t))\rangle_{\psi(t)} - \gamma(t)\, n \geq \langle H_{eff}(\gamma(t+\delta t))\rangle_{\psi(t)} - \gamma(t+\delta t)\, n$
4. Because $\langle H_{eff}(\gamma(t))\rangle_{\psi(t)} \geq -\gamma(t)\, n \ \forall_t$, $\langle H_{\text{problem}}\rangle_{\psi(t)} \leq 0 \ \forall_t$

Unpublished work with Adam Callison, Viv Kendon, and others

# Intuitive example: two stage quantum walk

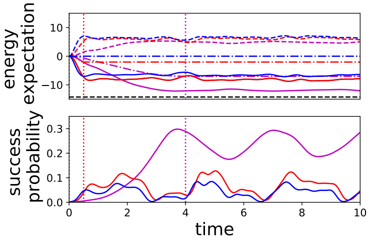
Perform a quantum walk at $\gamma_1$, and than use result as an input state for a second walk at $\gamma_2 < \gamma_1$




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

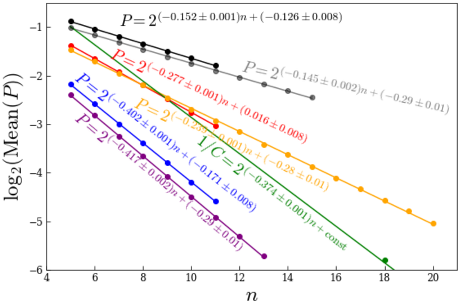
# Pre-annealed quantum walk, single spin glass example

Perform an anneal before the walk to dissipate energy




- Vertical dashed line is end of pre-anneal
- Longer pre-anneal lowers $\langle H_{\mathrm{problem}} \rangle$ (solid lines top plot) and raises success probability
- How does this affect scaling?
- Stop in paramagnetic regime and avoid exponentially small gaps in spin glass

# Scaling boost from pre-annealing




- Blue and Magenta quantum walk (two different ways of choosing $\gamma$)
- Red and Gold Pre-annealed walks with $\gamma$ values from regular QW
- Black and Gray Pre-annealed quantum walk with more optimal $\gamma$
- Green Effective scaling for classical branch-and-bound (for comparison)

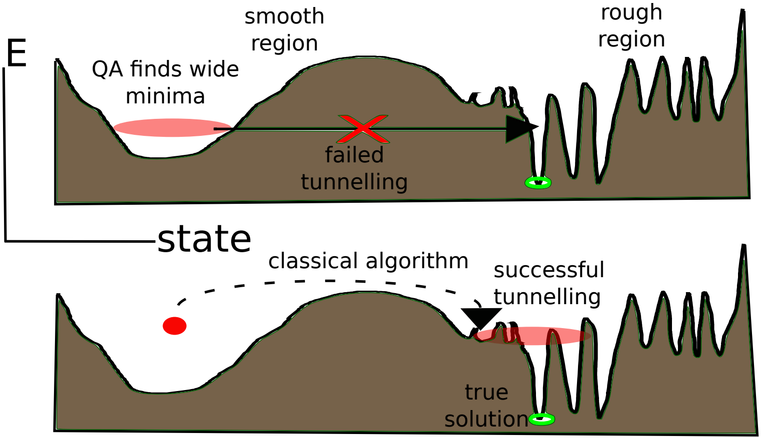
# Pre-annealed quantum walk beats classical state of the art

- Thanks to Zoe Bertrand (summer project student at Durham) for optimal branch-and-bound (BnB) implementation
- Scaling exponent less than half of state-of-the-art classical
- Comparable to quantum branch-and-bound scaling exponent found in arXiv:1906.10375 ours: 0.145, theirs 0.186

However...

- Our techniques are not hybrid like the quantum BnB (i.e. do not use classical tricks on top of quantum)
- Room for improvement as a subroutine in hybrid quantum classical? (maybe even combining with quantum BnB)

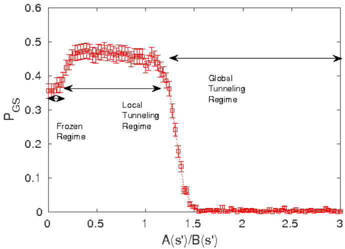
# Hybrid quantum/classical algorithms



# A subroutine for hybrid quantum/classical optimization

Basic requirement: needs to be able to incorporate outside information to solve problem

- One way to do this $\rightarrow$ search preferentially around candidate solution




How to do this experimentally: (dissipative) Reverse annealing

- Seed in guess solution on D-Wave quantum annealer
- Quantum fluctuations plus dissipation search locally
- Pioneered by me in New J. Phys. 19 023024 (2017)

# Obligatory slide: D-Wave controversy

Two separate controversies:

1) Are the dynamics actually quantum? Yes!

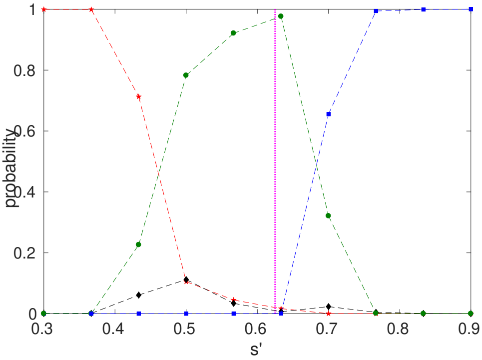
- Lots of evidence, most striking is simulation of extremely quantum KT phase transition Nature 560 456–460 (2018)
- Classical models reproduce **some** behaviours, expected $\rightarrow$ mean field approximation

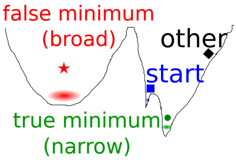
2) Can it ~~beat~~ improve classical computing? Open question

- No conclusive speedup demonstrated yet
- Not what this talk is about

- Currently largest scale device to study algorithmic application of quantum mechanics
- **Good science can be done regardless of answer to question 2!**

# Experimental biased search on a D-Wave device




- Unpublished experimental work by me
- $s'$ parameter controls amount of bias
- Able to find nearby (correct) solution a moderate value of $s'$ parameter, frozen at large $s'$, finds wrong solution at small $s'$

Experimental details in my AQC 2018 talk: https://www.youtube.com/watch?v=hSKCVESA-D8

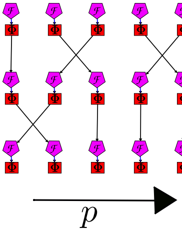
# Reverse annealing in algorithms (mostly work by others)*

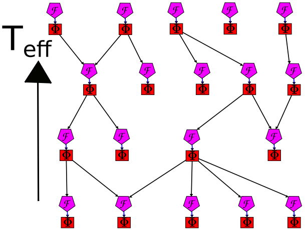
1. Start from one solution to find other solution (D-Wave whitepaper 14-1018A-A)
   - Finding other solution 150x more likely then forward
2. Search locally around classical solution (ar$\chi$iv:1810.08584)
   - Start from greedy search solution
   - Speedup of 100x over forward annealing
3. Iterative search (ar$\chi$iv:1808.08721)
   - Iteratively increase search range until new solution found
   - Forward annealing could not solve any, reverse solved most
   - See also: ar$\chi$iv:2007.05565
4. Quantum simulation(Nature 560 456–460 (2018))
   - Seed next call with result from previous
   - Seeding with previous state makes simulation possible
5. Monte Carlo and Genetic like algorithms
   - Quantum assisted genetic algorithm QAGA (ar$\chi$iv:1907.00707)
   - Finds global optima quickly where other methods struggle
   - Theoretical discussion (my work) (NJP 19, 2, 023024 (2017) and ar$\chi$iv:1609.05875)

*forward annealing= traditional non-hybrid method

# Hybrid quantum/classical, what's next?

1. More sophisticated algorithms
   - Except for QAGA, all experiments have been very simple algorithms
   - Move to more complex ones based on current state of art (particularly the state of the art for specific problems)
   - Develop theoretical framework: inference primitive → ar$\chi$iv:1609.05875




2. Understand and improve protocols
   - Understand how these protocols actually work under realistic conditions

# Applied quantum computing II: solving the right problem

Pretend we have an arbitrarily large perfect quantum computer $\rightarrow$ many algorithms and mappings, don't need to pick carefully

But this is not the real world $\rightarrow$ machines growing and improving slowly, **exciting but still limited**

Even picking the right problems to solve is non-trivial, needs input from the end users

- This is why I have hired a PDRA with a non-QC background
- Putting together workshop with ARC
- Work with startups on use cases, examples:
  1. Finance problems with Quantum Computing Inc.*
  2. Drug discovery with Kuano
  3. Ambulance dispatch with Applied Qubit (see: ar$\chi$iv:2006.05846 )

*Recently added to technical advisory board: https://www.hpcwire.com/off-the-wire/quantum-computing-announces-dr-nicholas-chancellor-as-technical-advisor/

# What makes a good early use case?

Early quantum computers may be powerful but relatively...

expensive

Needs to be a high value problem

Needs to be hard classically, otherwise why bother

small

Low processor throughput, quantum processor runs on 'small' (sub)problem (overall problem could still be high throughput)

NP-hard optimization problems and simulations of electrons are two examples which fit these criteria, there are others as well

Image: public domain taken from wikimedia commons

# What makes ~~good~~ **the best** early use cases?

Everything on the previous slide and...

Problem mapping overheads need to be low

Right size and shape of problem to map to existing machines or special purpose which could be built

Needs hardware *and* problem mapping expertise

structure of interesting instances needs to be understood

Needs application domain experts

Hybrid quantum/classical to get the best out of the machine

Classical algorithms where quantum subroutines can be incorporated

Needs domain *and* quantum expertise

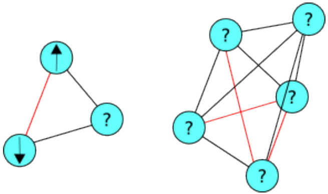
Fundamentally multidisciplinary

# From quantum to quantum-inspired

Quantum computing → very exciting could be game changing for computing but... requires quantum hardware advances

What if we don't want to wait for hardware? → quantum inspired

- Partially inspired by quantum annealing Fujitsu⋆ and Hitachi⋆ have built completely classical CMOS annealers
- Microsoft work on Quantum Monte Carlo



Most of the work I have done will carry over to a quantum-inspired setting → looking into getting access to machines

⋆https://www.fujitsu.com/global/digitalannealer/

⋆https://www.hitachi.com/rev/archive/2017/r2017_06/r6-10/index.html

# More about quantum inspired... use cases

Early quantum inspired will be...

less expensive

Not necessarily high value, maybe still moderate value for ASIC implementations

Still hard with traditional methods $\rightarrow$ don't reinvent the wheel

not so small

Don't need to restrict to low throughput

Should consider for use cases which are not suitable for fully quantum treatment (and maybe some which are)

Image: public domain taken from wikimedia commons

# So we should just do quantum-inspired instead... No

Techniques and ideas likely mutually useful → what works well for q-inspired is likely to work well on full quantum

Proven advantages for being quantum (recall earlier slide)

(011) (010)

(111)

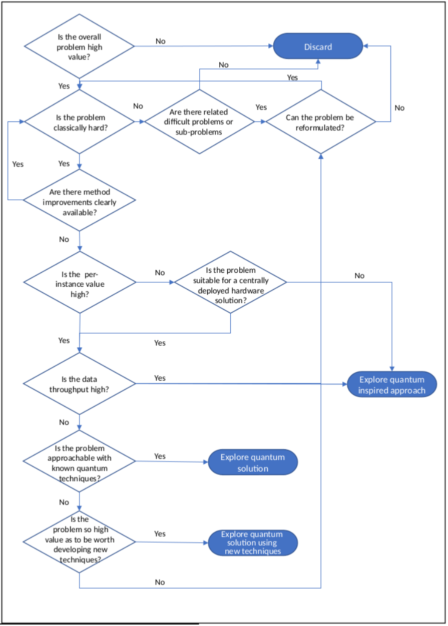
(000)

(001) (110)

should be treated as synergistic

- Quantum-inspired + other classical hetrotic → how we win today
- Full hybrid quantum/classical → how we win tomorrow
- Fully quantum with no hybrid → why do this? many sub-operations (ex. adding numbers) don't need quantum

# Putting it all together: use case methodology *




*From arχiv:2006.05846 collaboration with Applied Qubit

# Take away messages

Computing with quantum spin systems is an exciting interface between computer science and physics

Hybrid quantum/classical

Quantum machines should be used as subroutines $\rightarrow$ don't throw away all the good algorithms we already know

Developing better protocols

Quantum walks can find good solutions by energy conservation $\rightarrow$ can be extended to rapid (non-adiabatic) quenches

Competitive with state of the art quantum algorithms, but much more practical

Early use cases

Finding the best problems is fundamentally multidisciplinary $\rightarrow$ need non-quantum (application domain etc...) experts to contribute

Lots of work I didn't have time to talk about $\rightarrow$ see my webpage for more info http://nicholas-chancellor.me, or ask me

# Supplemental slides

# Context related to recent 'quantum supremacy*' result

Recent result posted by NASA: quantum supremacy in Google machine

What does this mean in simple language:

- Google machine appears to be *very* hard to simulate classically: evidence toward useful QC
- QS is not a demonstration of a useful application though

Where does our work fit with this...

- Finding useful applications is next logical step
- But we focus on a different kind of machine than the Google machine

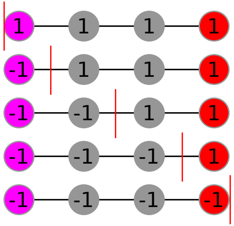
*Myself and many others in QC object to this use of the term 'supremacy' for a number of reasons, we are working toward getting the field to use an alternative term, see: Palacios-Berraquero et al Nature 576, 213 (2019)

# Bonus story: integer variable encoding

Want to encode discrete variables with more than 2 values into qubits

- Cumbersome to encode using traditional (one hot) method: $N$ value integer variable $\rightarrow$ $N$ qubit fully connected subgraph
- Better 'domain wall' encoding (see Quantum Science and Technology 4, 045004 (2019))
  " " $\rightarrow$ $N-1$ qubit linearly connected subgraph

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



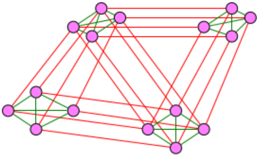
# Interactions between domain walls

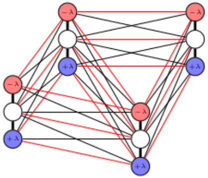
Ising chains with single domain wall $-1$ boundary condition to the left, $+1$ boundary to the right

- $\delta_i = \frac{1}{2}(Z_i + Z_{i-1})$, $\delta_i = 1$ iff domain wall between $i$ and $i-1$, 0 otherwise
- Products of $\delta_i$ on different chains are quadratic $\rightarrow$ arbitrary interactions between pairs of domain wall variables is qudratic
- 'virtual' Ising variables beyond end of chain $\rightarrow$ binary variable is special $N=2$ case of domain wall encoding

Use natural structure of problem to 'spread out' embedding

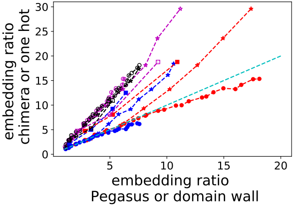
Four colouring example, 'layered' structure in Domain wall (right), no structure in one hot, (left)





# Domain wall encoding is a powerful tool for problem mapping

- Reduce number of qubits per variable by one
- Fewer connections within variable
- Structure tends to be better for embedding




- Red and blue → comparisons of domain wall versus one hot
- magenta and black → effect of more advanced ‘Pegasus’ hardware graph

Domain wall encoding can make as much of a difference as re-engineered hardware graph!

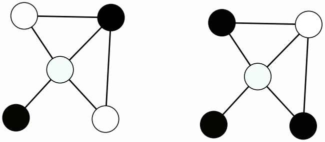
# Example of Ising problem mapping *

Have:

- Binary variables $Z_i \in \{-1, 1\}$
- Minimisation over Hamiltonian made of single and pairwise terms $\mathrm{H}_{\mathrm{Ising}} = \sum_i h_i Z_i + \sum_{j>i} J_{i,j} Z_i Z_j$

Want:

- Maximum independent set: how many vertexes on a graph can we colour so none touch? $\rightarrow$ NP hard



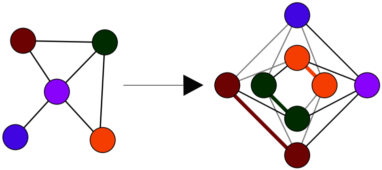
Method:

1. For an edge between vertex $i$ and $j$ add $Z_i + Z_j + Z_i Z_j \rightarrow$ penalizes colouring ($Z = 1$) adacent vertexes
2. Add $-\lambda Z_i$ to reward coloured vertexes ($0 < \lambda < 1$)

*Taken from the notes of a physics level 3 computing project I wrote, full notes at: http://nicholas-chancellor.me/QOpt_project_final.pdf

# Minor embedding

- Strong 'ferromagnetic' $(-Z_i Z_j)$ coupling energetically penalizes variables disagreeing
- If strong enough than entire 'chain' acts as a single variable
- Mathematically corresponds to mapping one graph to graph minors of another



Can embed arbitrary graphs into quasi-planar hardware graph with polynomial ($n^2$ for fully connected) overhead $\rightarrow$ Ising model restricted to hardware graph is also NP-hard

In practice this leads to a large overhead $\rightarrow$ important consideration for solving real problems potential bonus story if time and interest

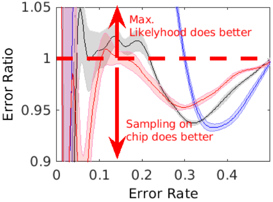
# Continuous time quantum computing

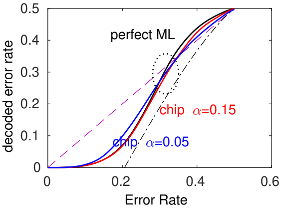
Physical system maps interesting computer science problem

Physics of system can be leveraged algorithmically to solve problems$^\star$: powerful marriage of physics and CS

Example: Maximum entropy inference on a *physical* quantum annealer NC et. al. Scientific Reports vol. 6, 22318 (2016)

- Thermal states maximize entropy $\rightarrow$ can be used to decode communications
- Superconducting quantum device produces (approximately) these distributions, can beat less powerful classical techniques




$^\star$of course there are many details here I don't have time to discuss