Continuous time quantum computing beyond adiabatic: quantum walks and fast quenches

Zaiku Group

Based mostly on arχiv:2007.11599*, but includes other work as well

Nicholas Chancellor

August 11, 2020

*Written jointly with Adam Callison, Max Festenstein, Jie Chen, Laurentiu Nita, and Viv Kendon

A brief note about terminology

For the purposes of this talk:

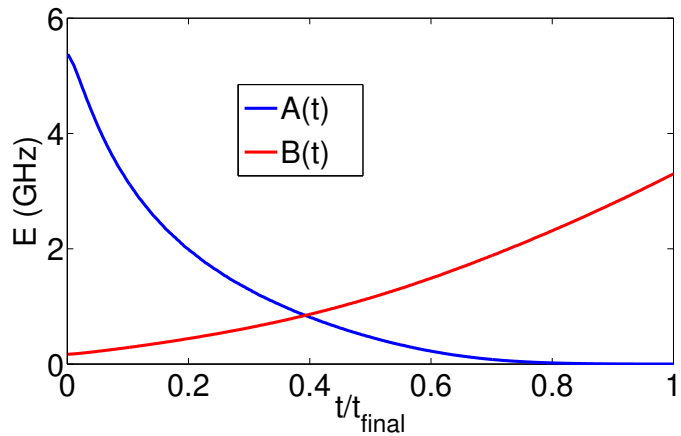
- Adiabatic quantum computation (AQC) $\rightarrow$ *closed system* protocols where an eigenstate is maintained via the adiabatic theorem of quantum mechanics
- Quantum Annealing (QA) $\rightarrow$ system is not well described by the adiabatic theorem, either because it is an open system or because evolution is much faster than adiabatic

The terminology is not standardized and different groups may use these terms differently

Adiabatic quantum computing

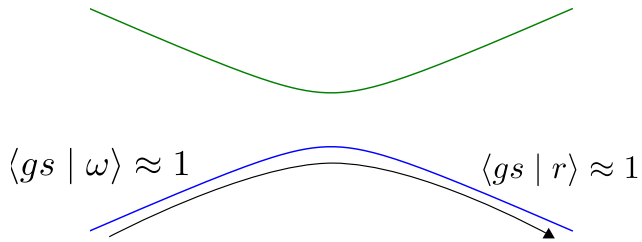
Traditional picture:

- 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

*For those unfamiliar with complexity theory, this is basically saying "let's assume that hard optimization problems exist", most experts believe P$\neq$NP

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 → 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

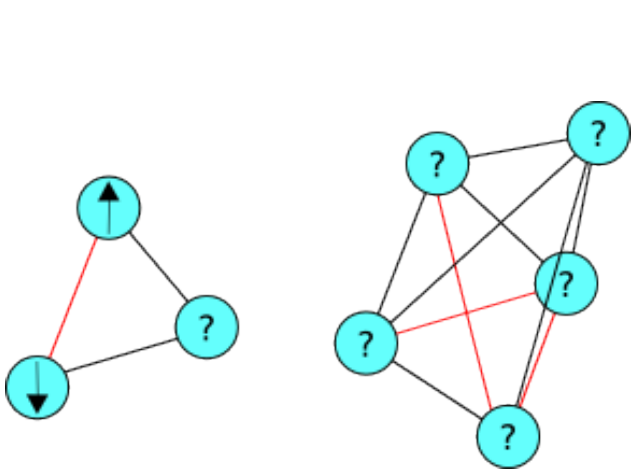
- *Total* runtime is still exponential in problem size
- Each run is short → exponentially many needed to hit right answer

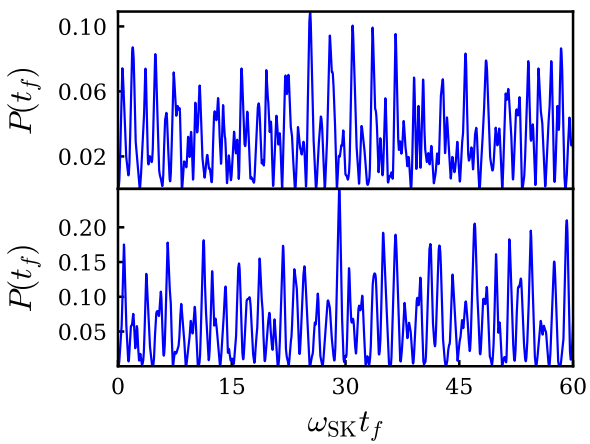
- Exponentially low success each run is *conceptually* unsatisfying...
- ... but much less demanding for coherence

Lottery

Example: continuous time quantum walk on spin glass

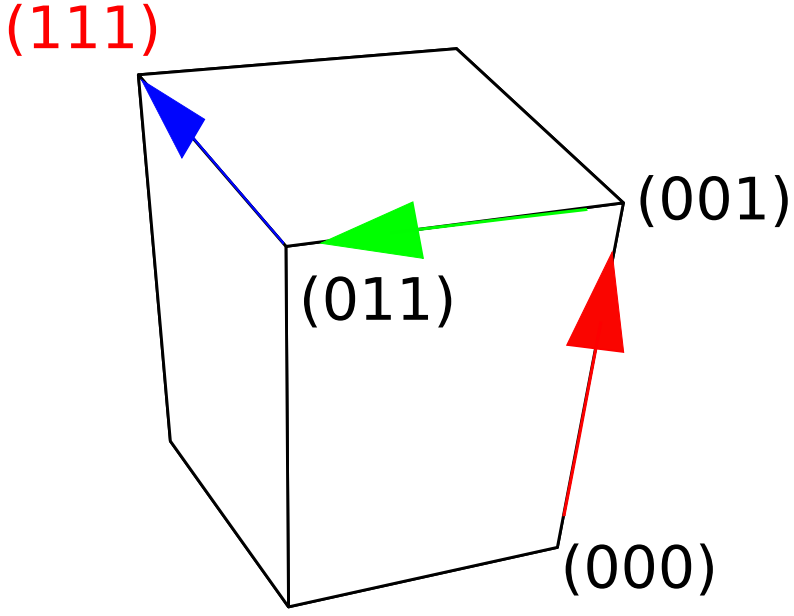
- Start with an equal positive superposition of all solutions, $|\omega\rangle = \frac{1}{\sqrt{N}} \sum_i |i\rangle$
  - Evolve with a fixed Hamiltonian $H_{\text{walk}} = \gamma H_{\text{hop}} + H_{\text{problem}}$
- $H_{\text{hop}} = -\sum_i \sigma_i^x \rightarrow$ superposition is ground state
-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

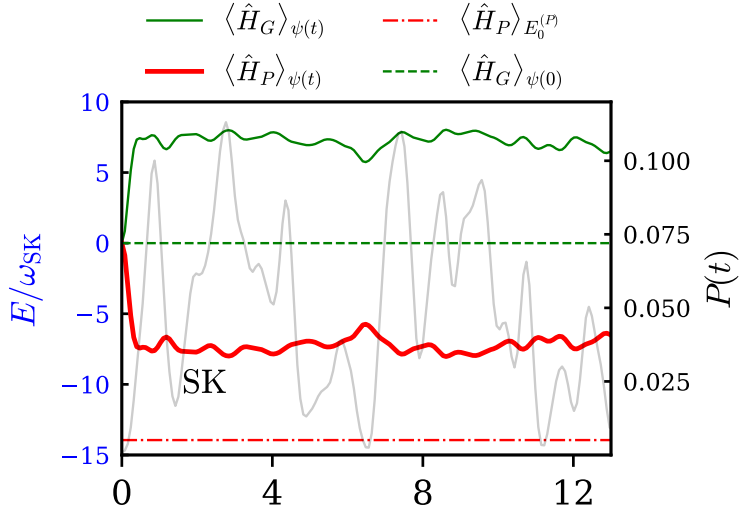




See Adam Callison et al 2019 New J. Phys. 21 123022 for details, work with Adam Callison, Viv Kendon, and Florian Mintert

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





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

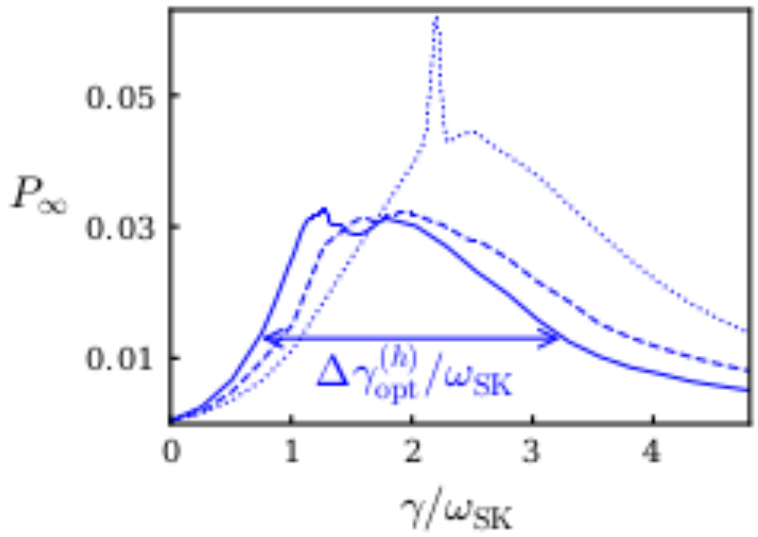
Energy is conserved $\langle H_{\mathrm{walk}} \rangle_{t=0} = \langle H_{\mathrm{walk}} \rangle_{t>0}$ since the system starts in the ground state of H_{hop}:
$\langle H_{\mathrm{problem}} \rangle_{t>0} - \langle H_{\mathrm{problem}} \rangle_{t=0} = \langle H_{\mathrm{hop}} \rangle_{t=0} - \langle H_{\mathrm{hop}}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

Walk seeks out 'good' solutions!

How much to walk? Choosing the γ parameter

$$H_{\text{walk}} = \gamma H_{\text{hop}} + H_{\text{problem}}$$

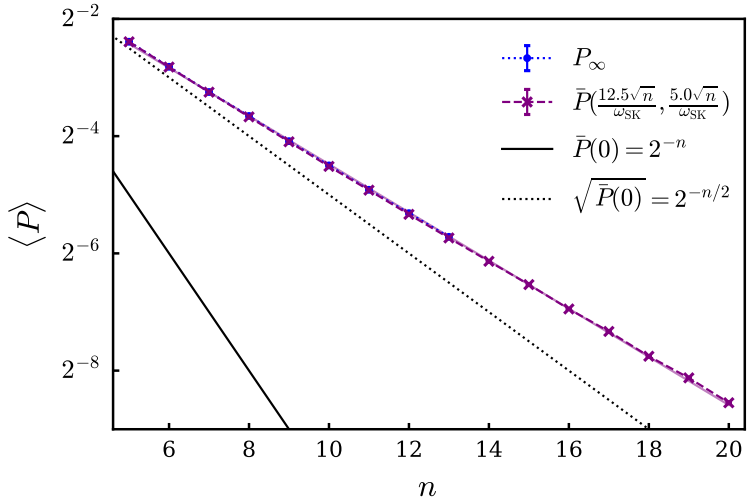
Still have one undefined parameter, γ, how do we set it? does it need to be set precisely? How do we make sure we are not 'cheating'?



P_∞ is the long time average success probability

Short answer: yes, γ does not have to be precisely set to find solutions effectively, and we can find a heuristic to choose it (without cheating). For details about the heuristic see Adam Callison et al 2019 New J. Phys. 21 123022

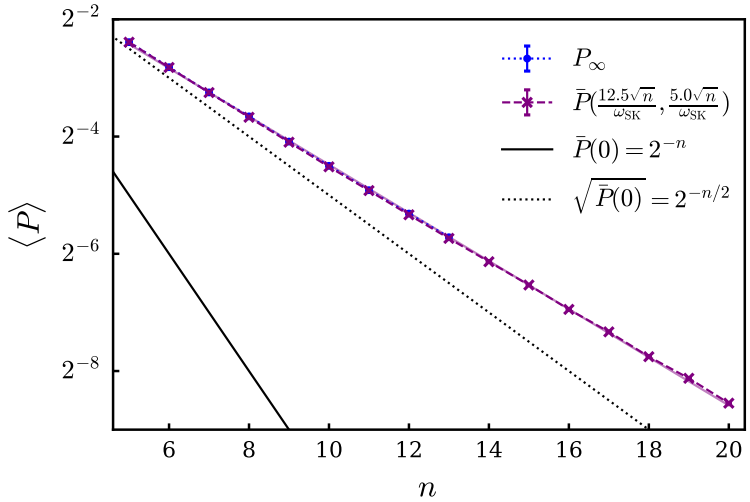
How well this works, numerically extracting scaling



- $N = 2^n$ possible bitstrings, one correct solution, runtime scales as inverse probability
- Scaling of $\frac{1}{N^{0.417}}$ better than both classical guessing ($\frac{1}{N}$) and $\frac{1}{\sqrt{N}}$ unstructured (Grover like) quantum search

The structure of the problem (correlations in bitstring energies) is playing a role in the computational mechanism, otherwise could not beat $\frac{1}{\sqrt{N}}$ scaling

A note on runtimes of single walk



There are actually two curves on top of each other in the above plot:

1. Infinite time average
2. Average over a time $\propto \sqrt{n}$ *

$\therefore$ The system is equilibrating quickly

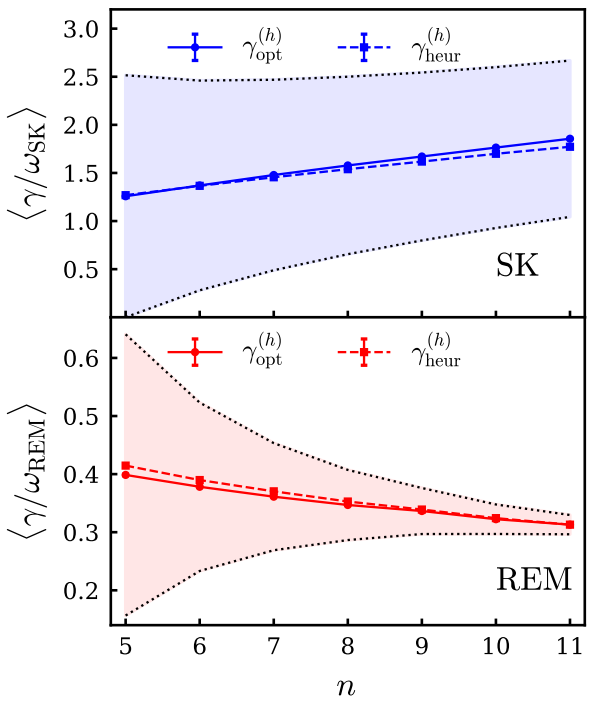
Implies that the walk is probably occurring in the precursor to a paramagnetic phase, rather than a spin glass phase $\rightarrow$ spin glass phase would equilibrate slowly

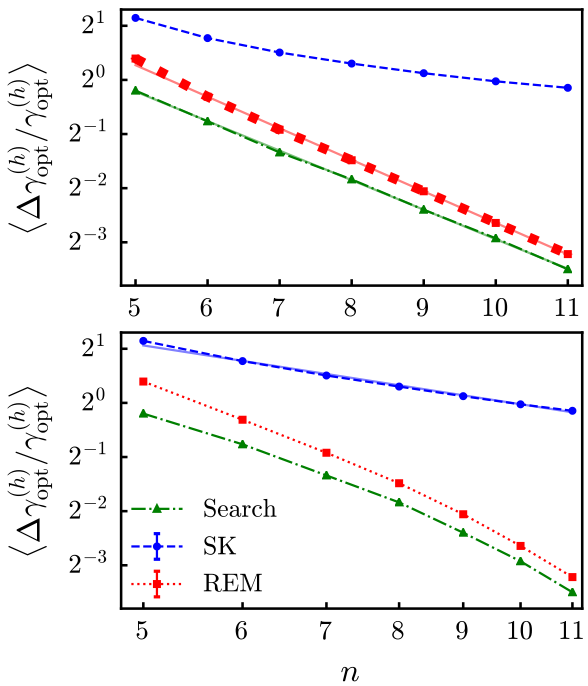
*yes n, the number of qubits *not* $N = 2^n$

Compare to problems without correlations

Random Energy Model (REM): each bitstring is assigned a random independent energy

- Dynamics become dominated by a single close avoided crossing, require fine tuned γ, technically difficult, may not be possible to find correct value
- Requires single long run for high success probability $\rightarrow$ need long coherence time

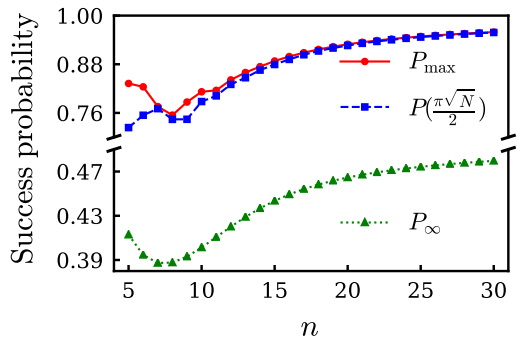




Compare to unstructured search

- Continuous time analog to Grover search
- Problem Hamiltonian is a single marked bitstring $|m\rangle\langle m|$:

$$H(s) = -(1-s)\sum_i \sigma_i^x + s(1-|m\rangle\langle m|)$$



- Same optimal speedup as gate model finds solution in time $\sqrt{N}$ rather than N from classical guessing/exhaustive search
- Exponentially sensitive to parameter setting (values of s)
- Succeeds with $O(1)$ probability after $\sqrt{N}$ runtime

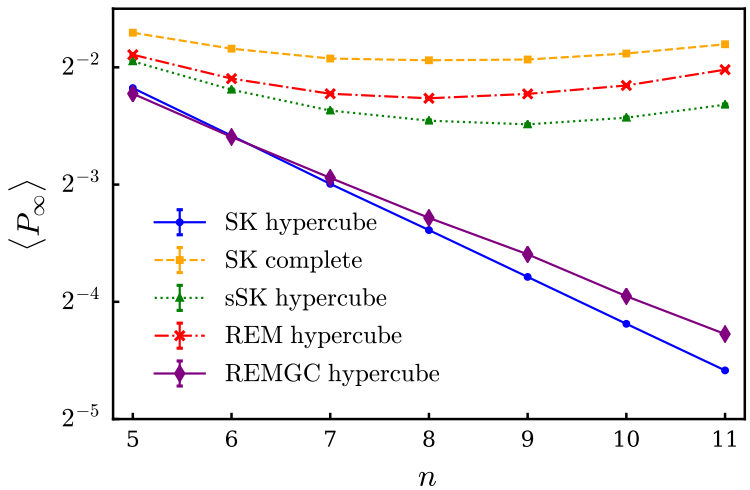
For a detailed study of AQC and QW, see Morley et. al. Phys. Rev. A 99, 022339 (2019), for practical implementation, see Dodds et. al. Phys. Rev. A 100, 032320 (2019)

(Why) is the effect of correlations interesting?

If the bitstring energies are uncorrelated no *classical* algorithm could do better than random guessing, why?

Energy of one bitstring tells nothing about energy of neighbours

Complementary to the search-like mechanism usually attributed to quantum algorithms



Not just a 'one off' difference between spin glasses and REM, but more general between correlated and uncorrelated energies

Beyond simple quantum walks

Study of quantum walks on spin glasses fruitful for understanding computational mechanisms, but scaling is not cutting edge

How do we build better algorithms on top of this result?

1. Add a (rapid) quench to dissipate some energy
   - Need theory which goes beyond adiabatic and works for rapid quenches
2. Use as a hybrid subroutine along with classical computation
   - Needs to be coherent and ideally compatible with the theory from point 1

Rapid quenches?

The energy conservation argument given previously can be 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 quantum walk $H_{eff}(\Gamma(t)) = \Gamma(t)\,H_{\text{drive}} + H_{\text{problem}}$
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 $\langle H_{eff}(\Gamma(t))\rangle_{\psi(t)} \geq -\Gamma(t)\,n \;\forall_t$, $\langle H_{\text{problem}}\rangle_{\psi(t)} \leq 0 \;\forall_t$

Details can be found in arχiv:2007.11599

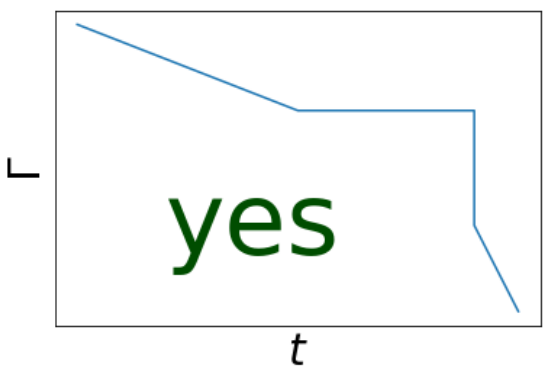
A very general result!

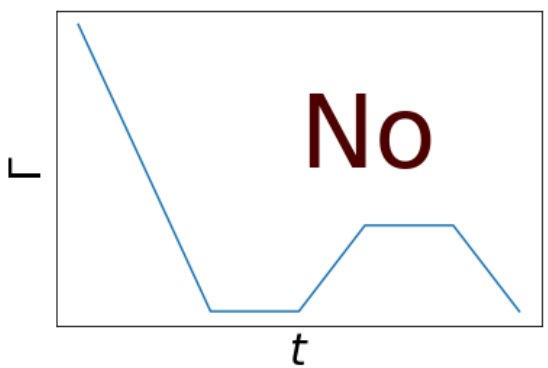
What is needed for result to hold:

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:

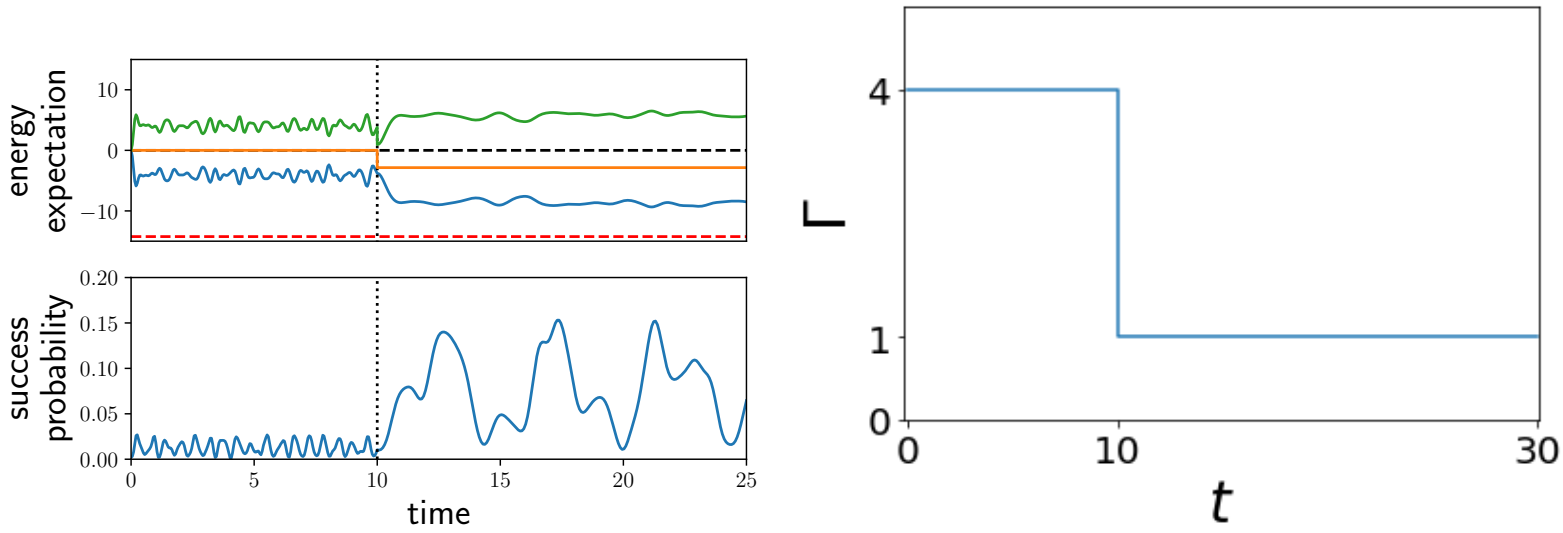
1. No limit on how fast algorithm runs
2. Discontinuities in $\Gamma(t)$ are ok
3. H_{drive} does not need to be diagonal in an orthogonal basis to H_{problem} $\rightarrow$ starting state can be biased





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 discussed later
3. Mechanism to understand dynamics very far from adiabatic limit

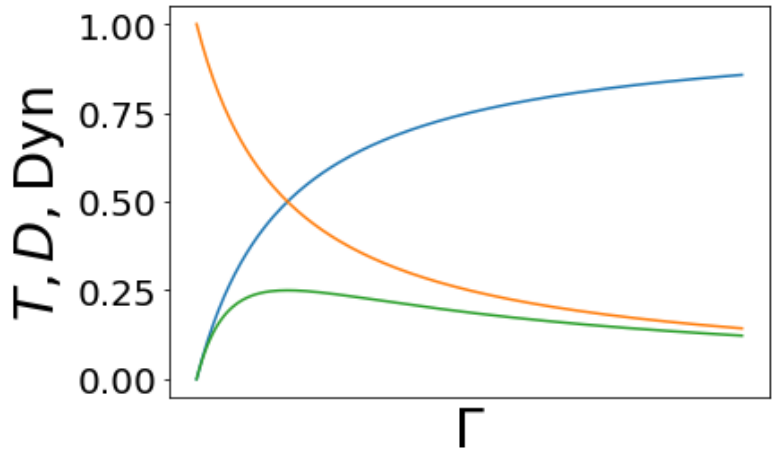
Couple with tools to quantify dynamics to make more powerful!

Quantifying dynamics in a two state subspace

Transfer coefficent, ability to 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

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

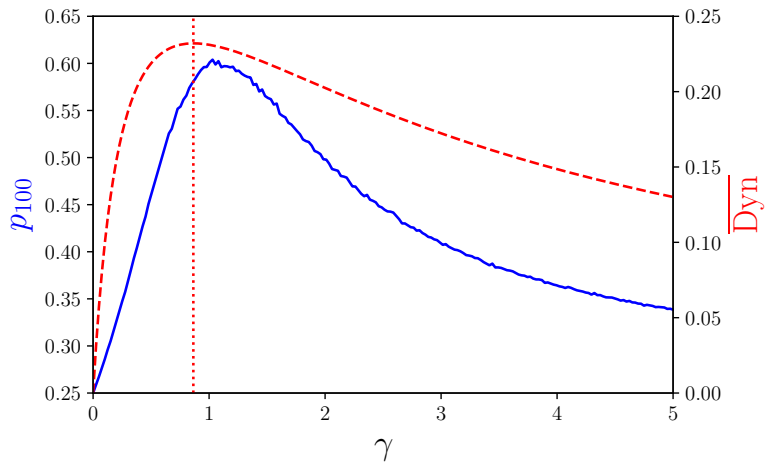


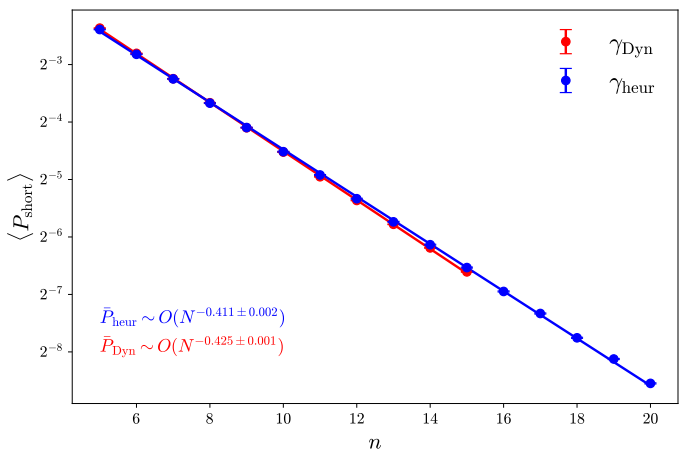
Estimating dynamics for the whole system

Sample over two level subsystems and find average Dyn value, $\overline{\mathrm{Dyn}} = \langle \mathrm{Dyn}^{(jk)} \rangle_{jk}$

- Can be estimated by statistical sampling even for large systems

Left: two qubit quantum walk example, p_{100} is average success probability for $t \leq 100$



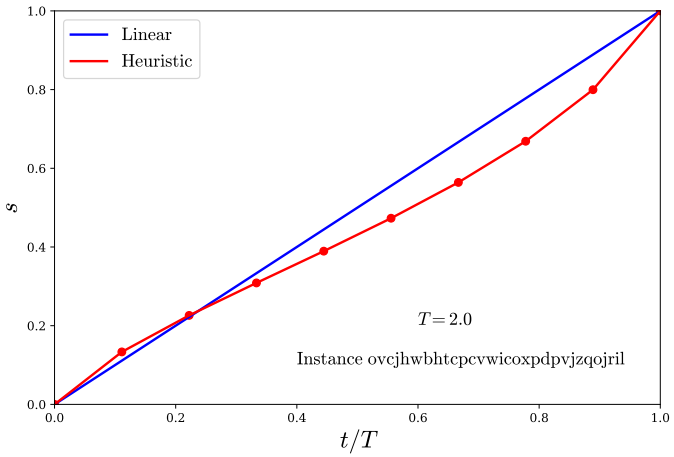


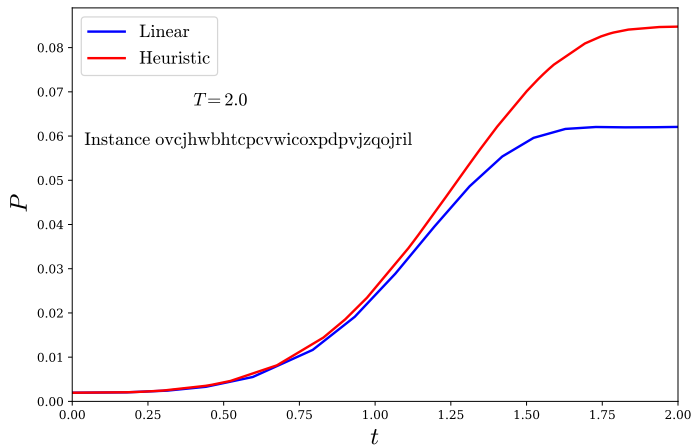
Right: Maximising $\overline{\mathrm{Dyn}}$ performs almost as well as the fine tuned heuristic from Adam Callison et al 2019 New J. Phys. 21 123022 for finding the best γ for quantum walk

Finding optimal quench 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 $\frac{\partial s}{\partial t} \propto \frac{1}{\text{Dyn}}$
- Compare to linear schedule $s \propto t$ for single SK instance

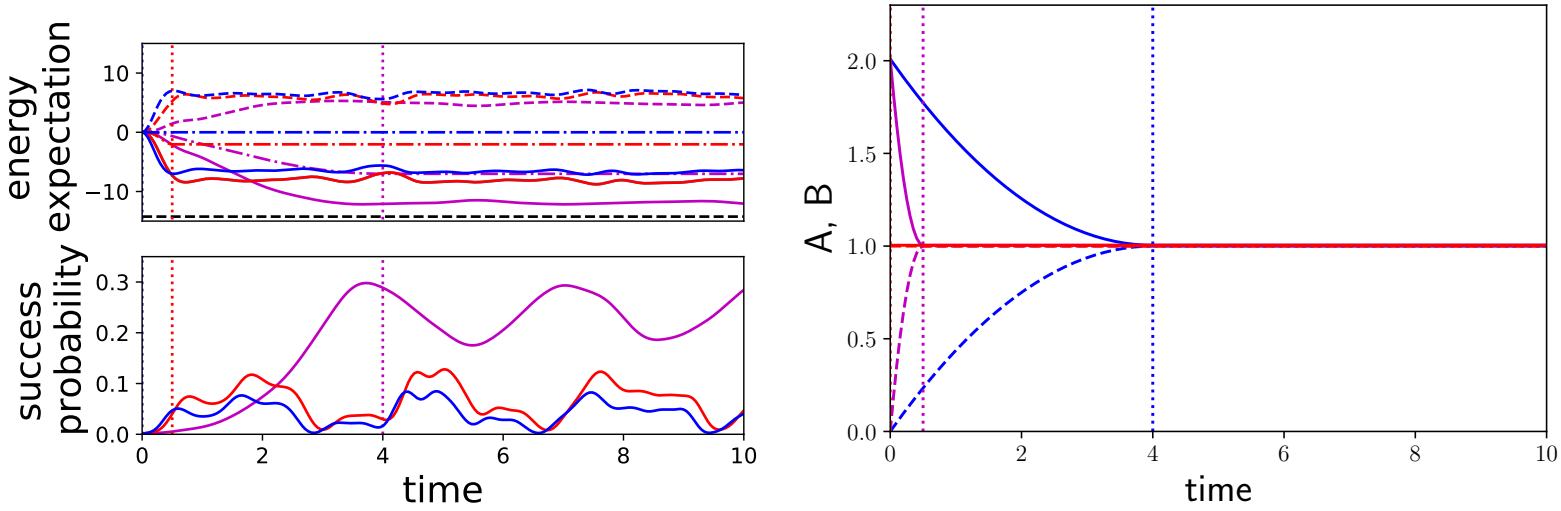




Heuristic performs better than linear schedule!

Pre-annealed quantum walk, single spin glass example

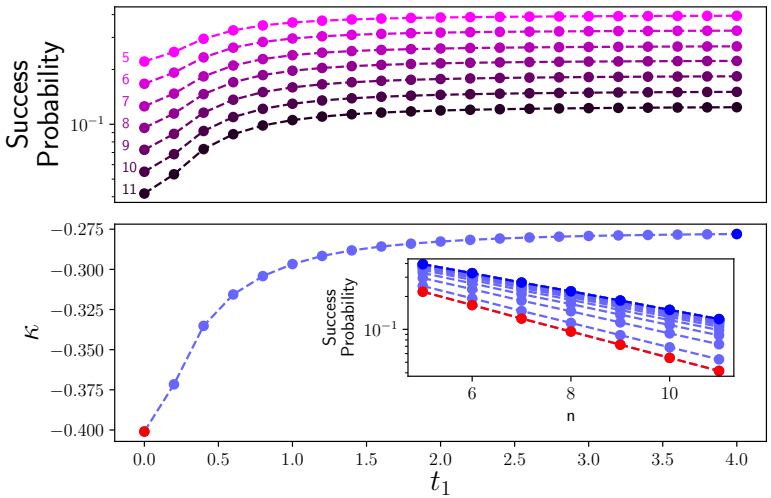
Perform an anneal before a quantum walk to dissipate energy



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

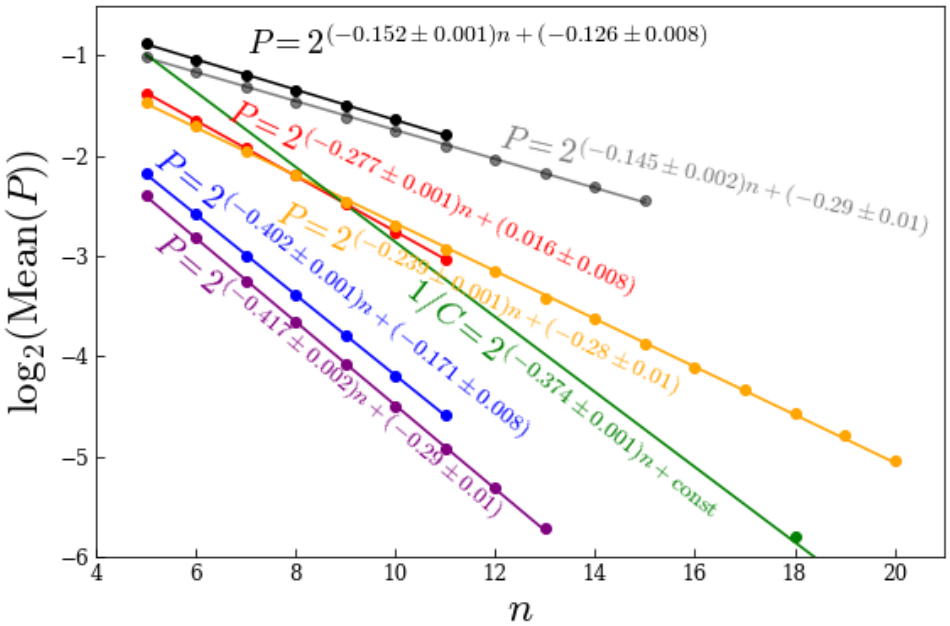
Scaling on spin glasses

Perform pre-annealed walk on spin glasses from Adam Callison et al 2019 New J. Phys. 21 123022 (these results can be found in arχiv:2007.11599)



- Top figure shows success probability versus pre-annealing time for different size spin glasses
- Bottom shows scaling with $p \propto 2^{\kappa n}$ when n is number of qubits (calculated based on inset)
- Pre-annealing not only improves success probability at one size, it improves scaling! $\rightarrow$ more on next slide

Scaling boost from pre-annealing



- Blue and Magenta quantum walk (two different ways of choosing γ) $\rightarrow$ a bit worse than classical state of the art
- Red and Gold Pre-annealed walks with γ values from regular QW $\rightarrow$ significantly better than classical state of the art
- Black and Gray Pre-annealed quantum walk (more optimal γ) $\rightarrow$ way better than classical state of the art
- Green Effective scaling for classical branch-and-bound (for comparison)

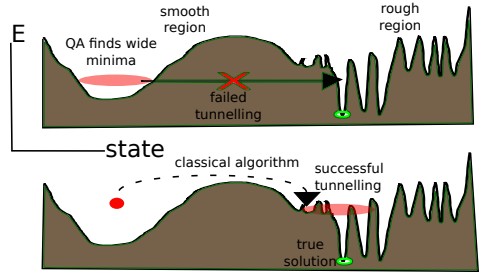
Pre-annealed quantum walk beats classical state of the art

- Thanks to Zoe Burtrand (summer project student at Durham) for optimal branch-and-bound (BnB) implementation
- Scaling exponent less than half of state-of-the-art classical (optimized version, currently unpublished)
- Comparable to quantum branch-and-bound scaling exponent found in arχiv: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 subroutines in Continuous time: a review



Known techniques:

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

Relies on dissipation, not suitable for coherent algorithms

'Mexican hat' schedule Perdomo-Ortiz et. al. Quantum Inf Process (2011) 10: 33. doi:10.1007/s11128-010-0168-z*

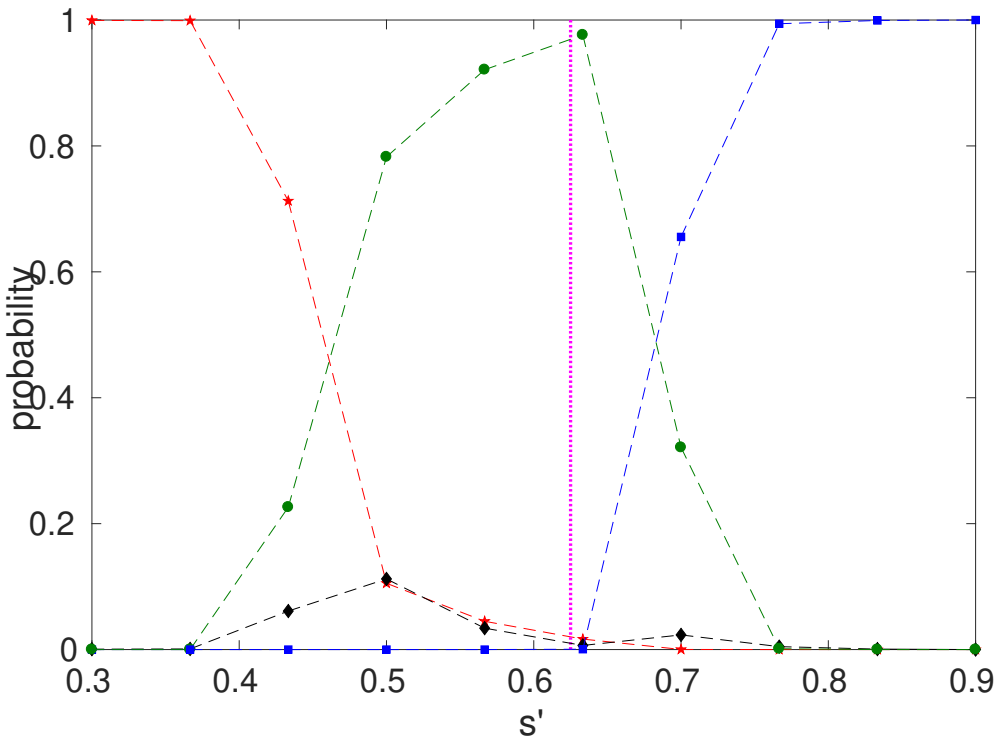
Involves three separate Hamiltonians, not compatible with rapid sweep proof in arχiv:2007.11599

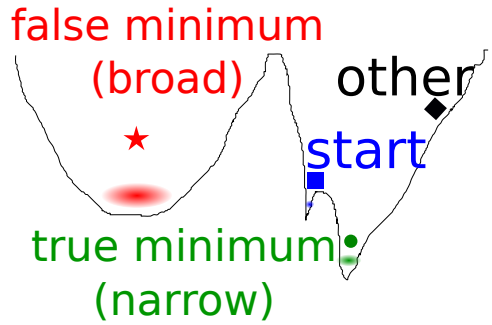
Biased driver Hamiltonian Chinese Physics Letters, 30 **1** 010302 and Tobias Graß Phys. Rev. Lett. 123, 120501 (2019)

Compatible with proof in arχiv:2007.11599, and can be used with quantum walk: this is the technique we want to focus on

*sometimes also called reverse annealing

Experimental biased search on a D-Wave device





- Level crossing, true GS and false minima at magenta line
- Anneal at maximum allowed rate, wait time (τ) of $20\mu S$
- Frozen in starting state for small s', find true minimum at moderate s', trapped for large s'

I can explain details of this experiment if people are interested

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 (arχiv:1810.08584)
   - Start from greedy search solution
   - Speedup of 100x over forward annealing
3. Iterative search (arχiv:1808.08721)
   - Iteratively increase search range until new solution found
   - Forward annealing could not solve any, reverse solved most
   - See also: arχiv: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 (arχiv:1907.00707)
   - Finds global optima quickly where other methods struggle
   - Theoretical discussion (my work) (NJP 19, 2, 023024 (2017) and arχiv:1609.05875)

*forward annealing= traditional non-hybrid method

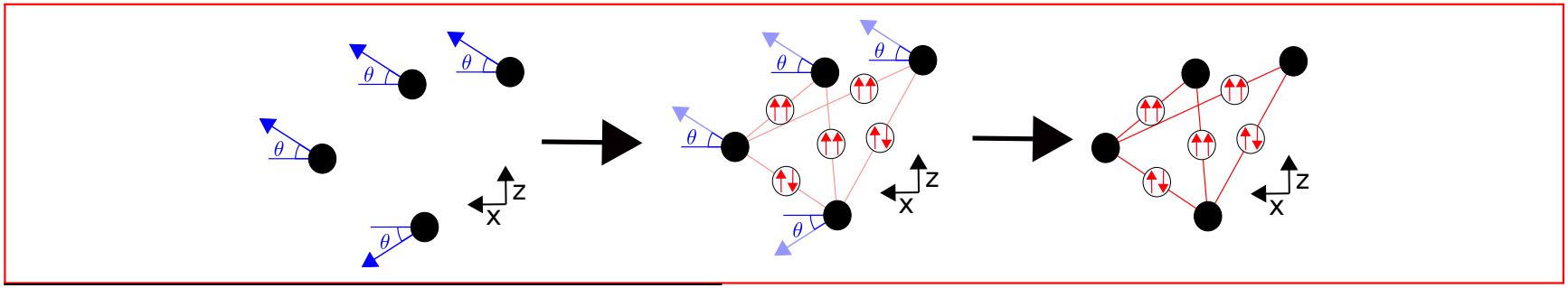
Biased driver Hamiltonian: our work*

Define driver Hamiltonian using fields which are not (completely) transverse $H_d = \sum_{i=1}^{n} -\cos(\theta)\sigma_i^x - g_i \sin(\theta)\sigma_i^z$

- Start in ground state of H_d:
  $|\psi(t=0)\rangle = \bigotimes_{i=1}^{n} \frac{1}{\sqrt{2+2\,g_i\cos(\theta)}}[(1+g_i\cos(\theta))|0\rangle + \sin(\theta)|1\rangle]$
- Starting state biased toward classical bitstring g, $g_i \in \{-1, 1\}$
- Closed system with monotonic sweep (including QW), time evolution improves the guess (on average):

$$\langle H_{\text{problem}} \rangle_{\psi(t)} \leq \langle H_{\text{problem}} \rangle_{\psi(0)}$$

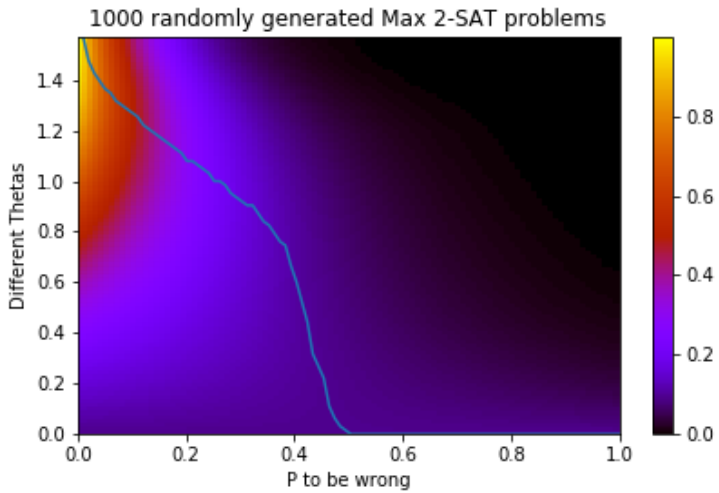
- Ground state is optimal solution so adiabatic theorem holds and dissipation can assist as well
  Can use AQC, QW and QA mechanisms simultaneously



*unpublished work with Laur Nita, Jie Chen, Adam Callison, Viv Kendon and Matthew Walsh.

Quantum walk with biased driver: proof-of-concept

- Consider a guess where each bit has an independent probability P of being wrong
- How good does the guess need to be before biasing (parametrized by θ, $\theta = 0$, no bias) improves the solution?



- Colour axis is success probability, line is optimal, result for eight qubit max-2-sat,
- hybrid techniques become useful right around 50% success probability, becomes significant around 45%
- Preliminary work by Laur Nita (PhD student)

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: arχiv: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

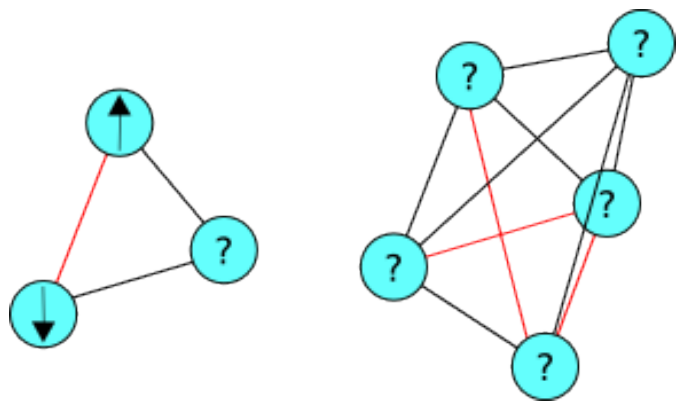
Fundamentally multidisciplinary

From quantum to quantum-inspired

Quantum computing $\rightarrow$ very exciting could be game changing for computing but... requires quantum hardware advances

What if we don't want to wait for hardware? $\rightarrow$ 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 $\rightarrow$ 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 $\rightarrow$ what works well for q-inspired is likely to work well on full quantum

Proven advantages of being quantum (ex: unstructured search)

(011) (010)

(111)

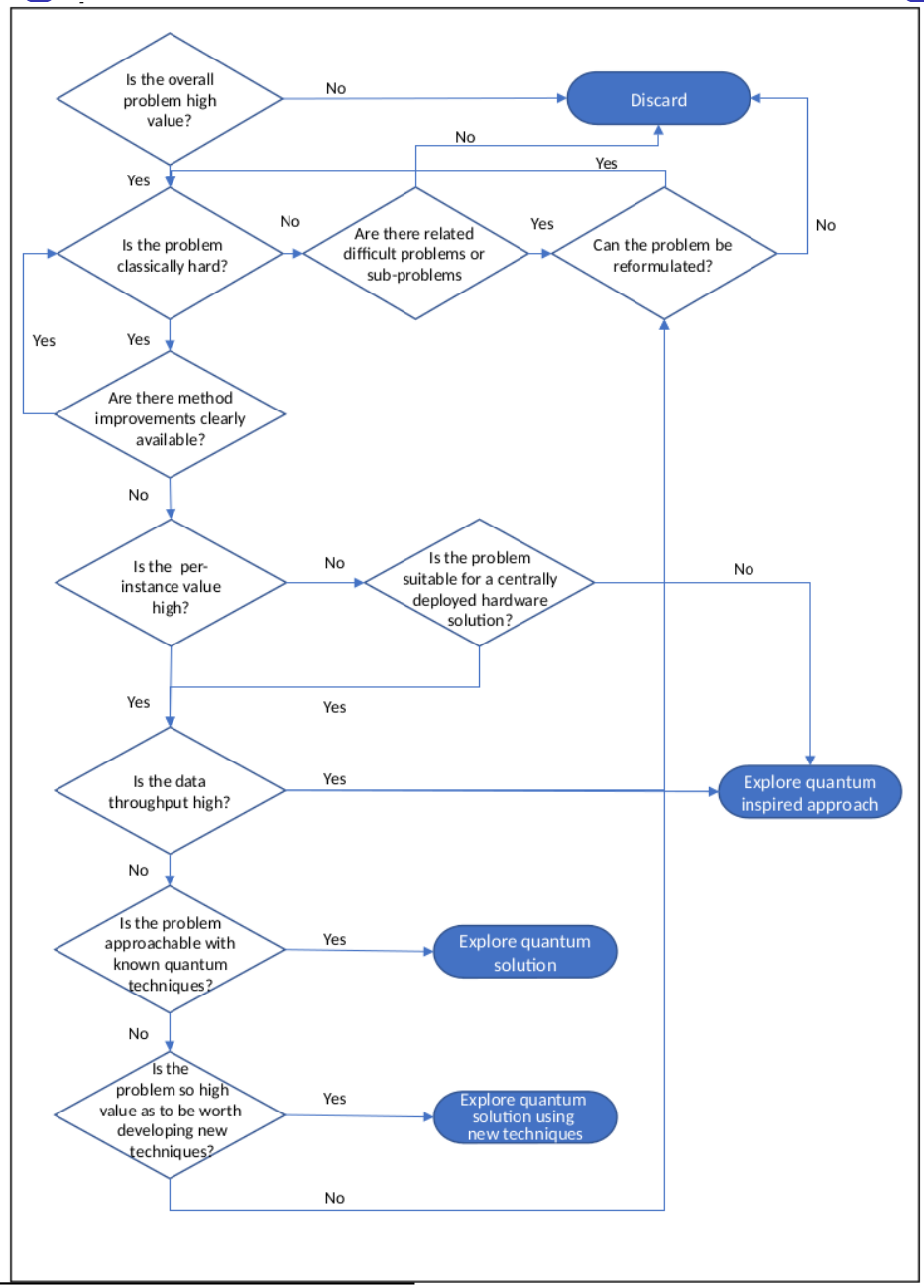
(000)

(001) (110)

should be treated as synergistic

- Quantum-inspired + other classical hetrotic $\rightarrow$ how we win today
- Full hybrid quantum/classical $\rightarrow$ how we win tomorrow
- Fully quantum with no hybrid $\rightarrow$ 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 home messages

Algorithms with exponentially low success probability in a single run

- Unless $P = NP$ all algorithms will have exponentially low success, exponential single run time, or both
- Only need to be coherent *for single run* $\rightarrow$ much less demanding for hardware (lower precision needed as well)
- Less psychologically satisfying, but no other real drawback

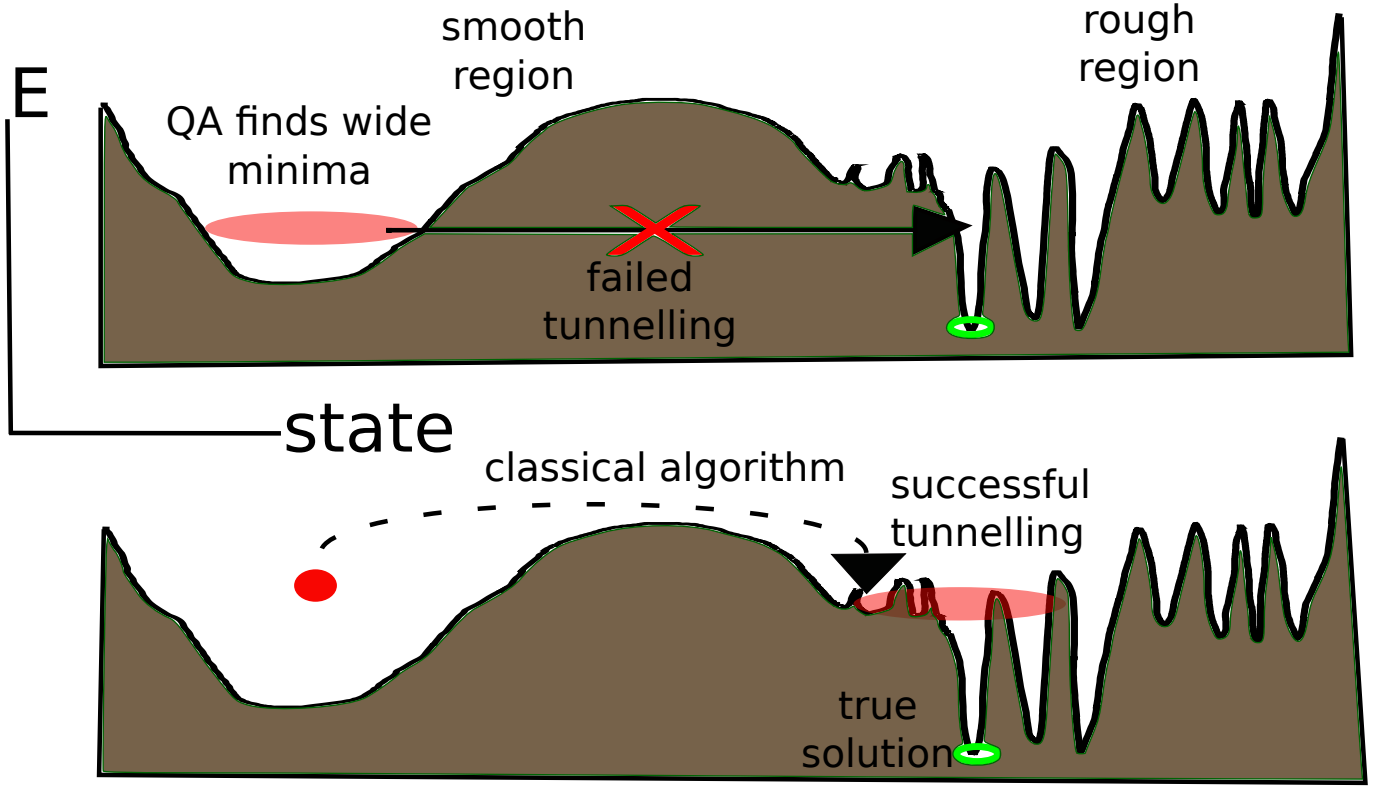
Quantum walks on spin glasses

- Correlations in energy landscape play a role, allow better than $\sqrt{N}$ runtime
- Behaves differently from simple search, less demanding for control precision (SS not good model for all computation!)
- Pre-annealing $\rightarrow$ performance competitive with state of art
- Working on hybrid subroutines

Bonus story: more on the reverse annealing experiment

What do all reverse annealing algorithms have in common?

All rely on the concept of local search, more likely to find solutions close (in Hamming distance) to starting state, than far away



Important to show *experimentally* that

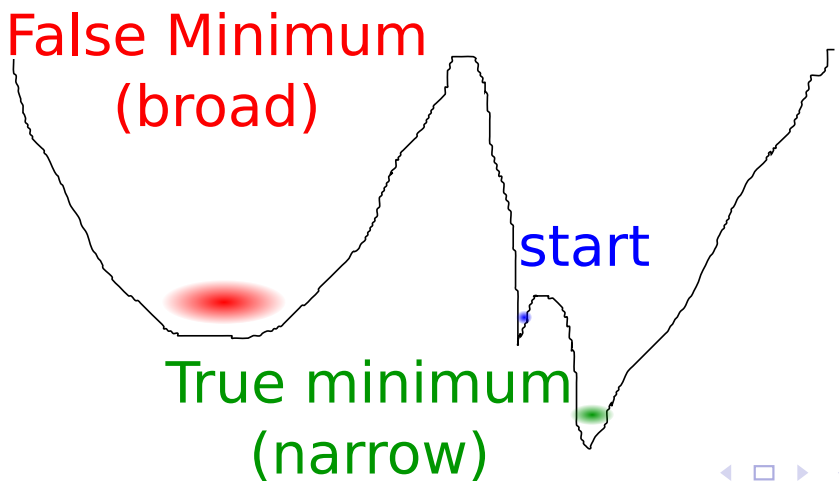
reverse annealing = local search

How to test this (proof-of-principle version)

Need to construct a problem Hamiltonian with the following properties:

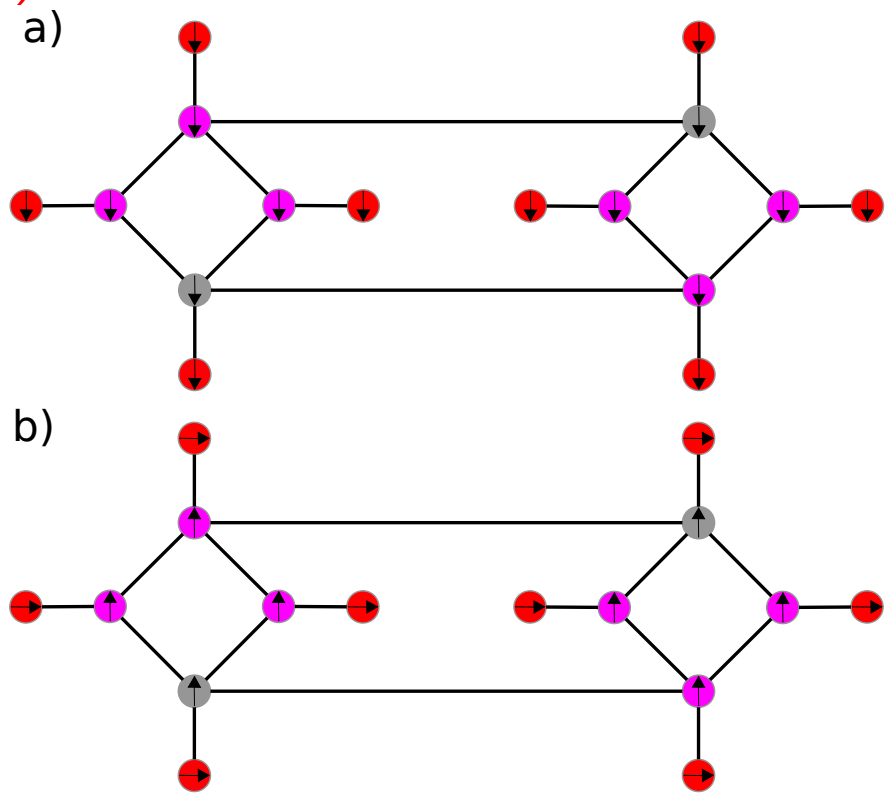
1. Wide false energy minimum which 'tricks' traditional quantum annealing algorithm
2. Relatively narrow true minimum energy
3. Local minimum near true minimum for start state

- For some value of the annealing parameter s_{cross}, fluctuations will lower the energy of the false minima below the true minima
- Local search demonstrated by tunnelling to true minimum while $s < s_{\mathrm{cross}}$



Constructing proof-of-principle Hamiltonian

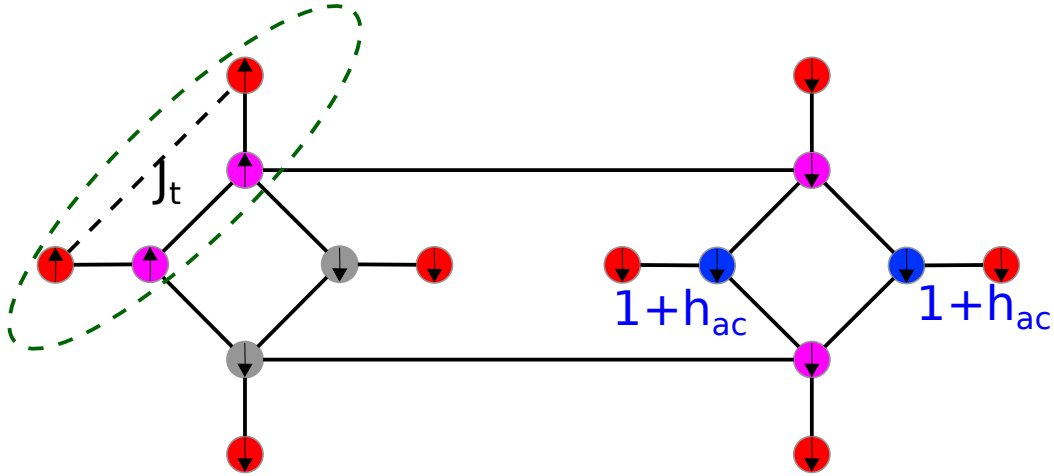
- Hamiltonians with features 1 and 2 are already known: free spin gadgets*
- Start with gadget from N. G. Dickson et. al. Nature Comm. 4, 1903 (2013)



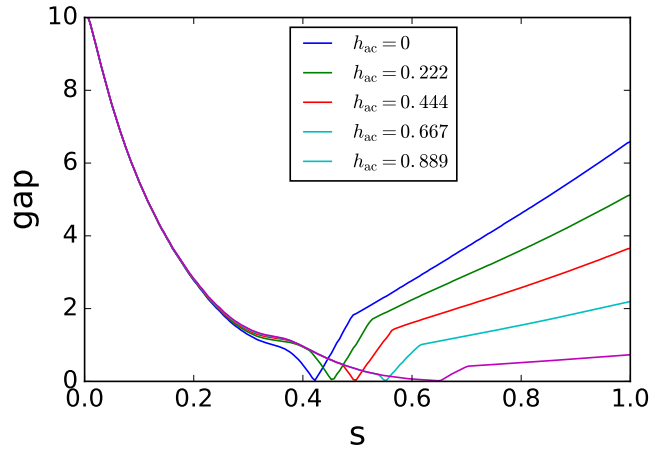
- a: unique ground state (red, h=+1 violet h=-1)
- b: 256-fold degenerate excited state → false minimum

*See for instance: S. Boixo et. al. Nature Comm. 4, 3067 (2013)

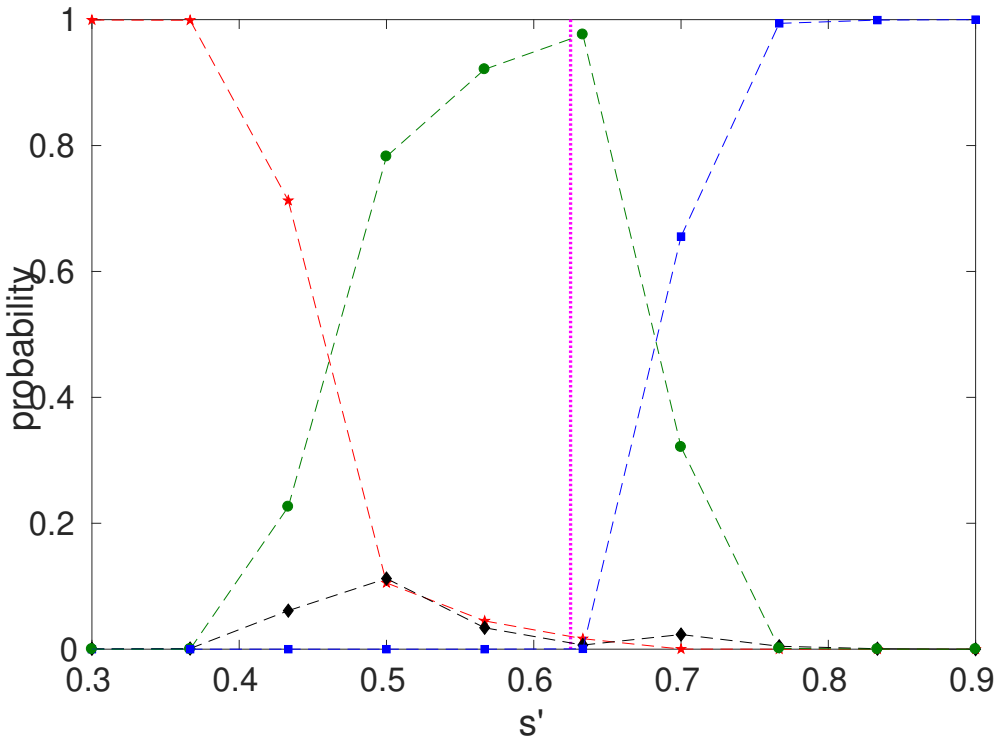
Add local minimum and make tunable

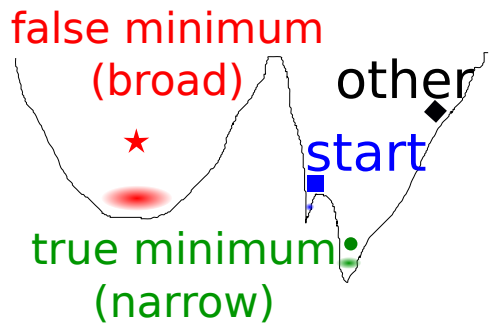


- Starting state shown by arrows, ground state except for circled spins flipped blue field is in - direction
- J_t controls barrier between start state and ground state.
- h_{ac} controls the value of s_{cross}



Experimental results *





- $J_t = 0.2$, $h_{ac} = 0.95$ level crossing between true GS and false minima at magenta line
- Anneal at maximum allowed rate, wait time (τ) of $20\mu S$
- Frozen in starting state for small s', find true minimum at moderate s', trapped for large s'

*For different proof-of-principle results, see: D-Wave white paper on Reverse Quantum Annealing for Local Refinement of Solutions

Future work

Have demonstrated the principles behind local search now to demonstrate it is a useful algorithmic tool

- Can we find better solutions starting from the best traditional QA can find (simple and immediate demonstration that reverse annealing is useful)
- Use problems with known solutions to test range over which reverse annealing can search
- Test sampling applications: balancing tradeoff between memory and search range