Exploring reverse annealing as a tool for hybrid quantum/classical computing

University of Zagreb QuantiXLie Seminar

Nicholas Chancellor

October 12, 2018

Talk structure

1. Background
   - Quantum computing: comparison between quantum annealing and 'gate' based QC
   - Quick overview on D-Wave quantum annealers
2. New Reverse annealing feature:
   - Enable hybrid quantum/classical algorithms
   - Experimental results showing reverse annealing searches solution space locally
3. Finding robust solutions with reverse annealing
   - Motivational example
   - Chip scale demonstration

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

Optimization and sampling

Solution space can roughly be thought of as a structured database with complex and unknown structure

- Finding optimal solutions to many real problems is important and difficult
- Similar to 'toy' database example, may be able to get the same kind of advantages
- Distributions of states which are some how 'optimal' may be used in machine learning

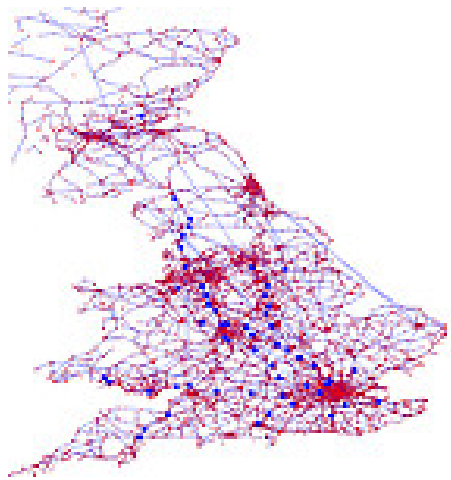


Image: public domain taken from wikimedia commons

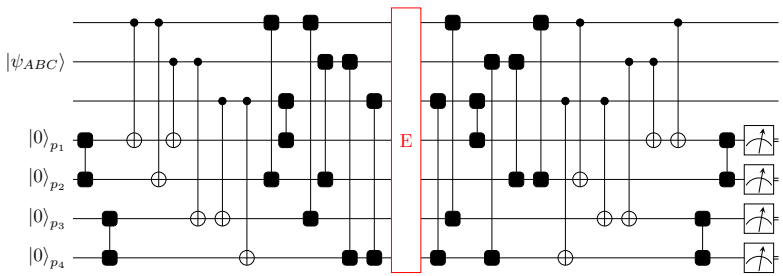
Two different approaches to quantum computing

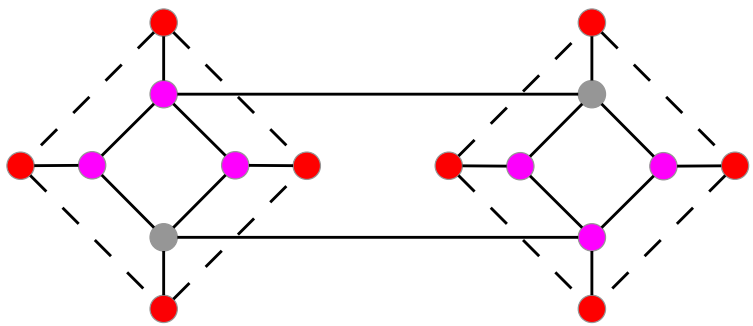
'Gate' based quantum computing

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

Quantum annealing

- Map optimization problem directly to energies of different states
- Allow quantum physics to help search solution space
- Low temperature environment helps solve problems





(some) Advantages and disadvantages of each

'Gate' based quantum computing

- Can simulate arbitrary quantum systems
- Error correction can get rid of all noise *in principle*
- Could simulate quantum annealing *in principle*

- Harder to build, largest device is tens of qubits
- All noise likely to be harmful rather than beneficial

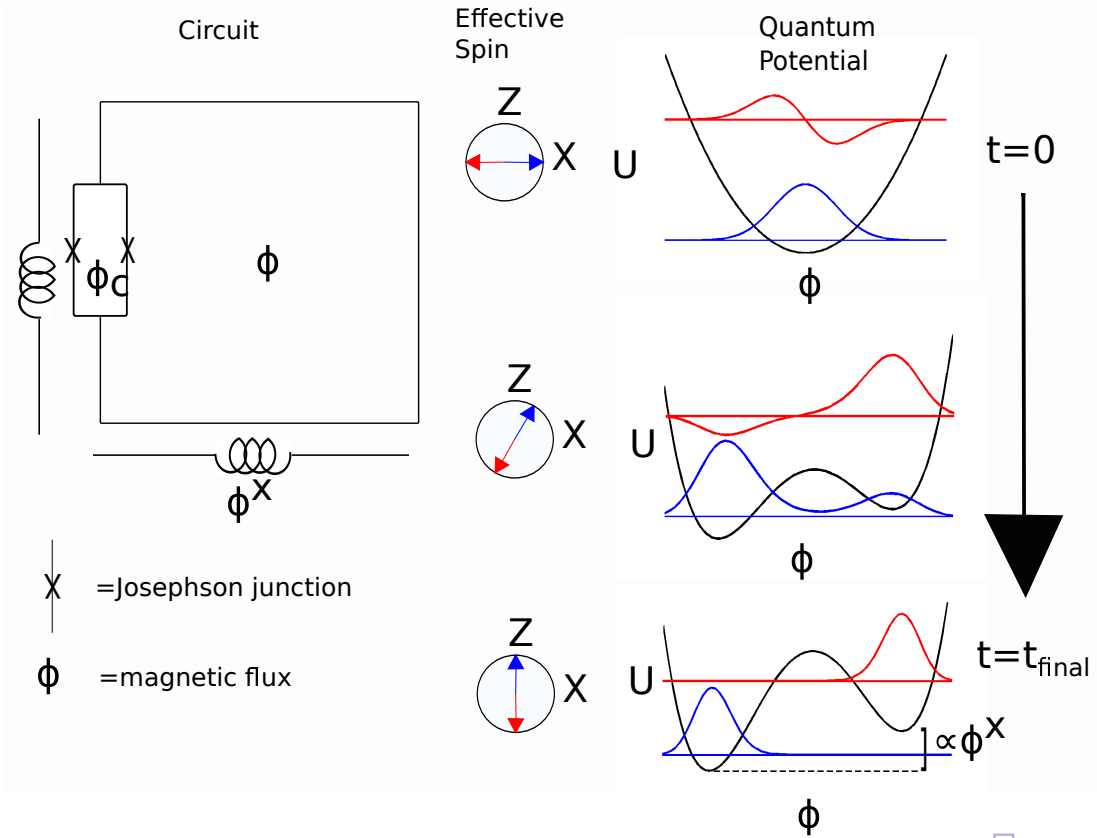
Quantum annealing

- Easier to build, largest device is thousands of qubits
- Tolerant to noise, in fact noise helps solve problems
- Naturally produces thermal distributions

- Unclear if error correction is feasible
- Cannot be used for some quantum algorithms

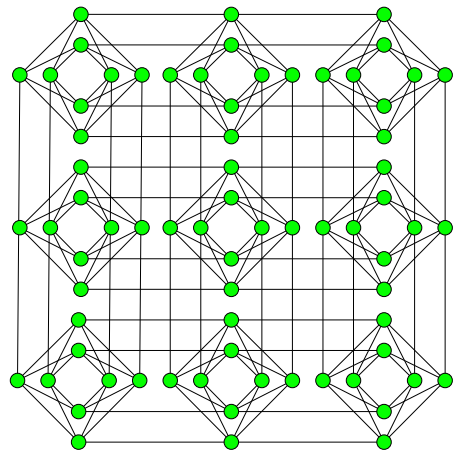
D-Wave Quantum annealing hardware

- Superconducting circuit devices with up to $2,048$ qubits in 16x16 'chimera' configuration
- Operates in a cryostat at ≈ 0.015 K (200x colder than interstellar space: 3K)
- Problem is programmed into 1 and 2 qubit penalty terms



How to actually solve problems with these devices: Optimization (traditional approach)

1. Map problem to one and two body terms of the appropriate form (Ising model) Optimality of solution $\rightarrow$ energy
2. Map to chimera graph by strongly linking qubits together to form 'logical' qubits (3x3 chimera shown below)



3. Quantum dynamics finds low energy states, run many times and take lowest energy solution

Each run is independent and starts from equal superposition 'state of maximal ignorance' could do better by using information from previous runs

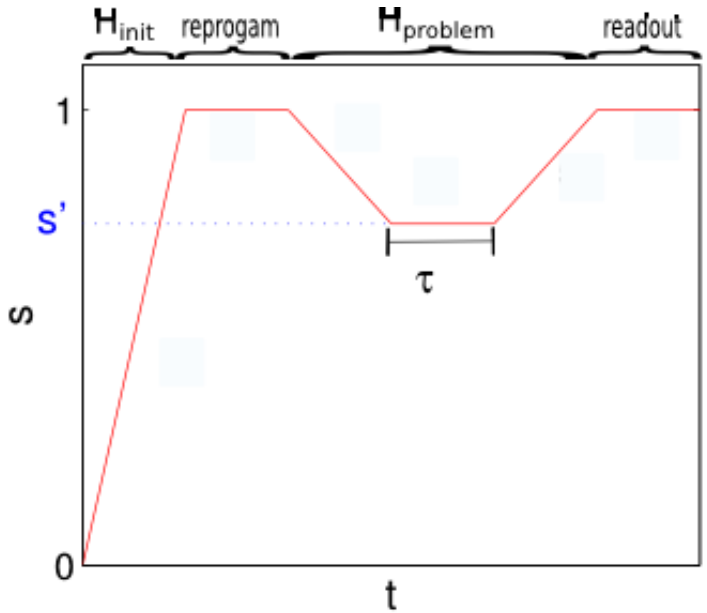
Why hybrid? Why reverse annealing?

Why hybrid (quantum/classical)?

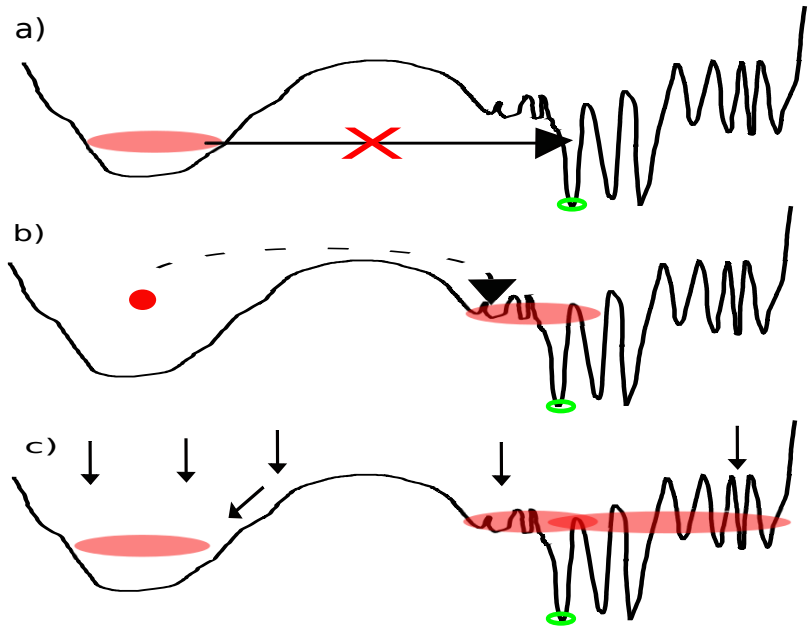
- Many good classical optimization algorithms already exist
- Need every advantage we can get to take advantage of early quantum hardware

Why reverse annealing?

- Easy inclusion of previously found solutions in algorithm calls (search range controlled by parameter s': $s' = 1 \rightarrow$ no search, $s' = 0 \rightarrow$ traditional annealing)
- Flexible: can be used with most existing techniques
- Now available on D-Wave devices



Cartoon example: energy landscape with rough and smooth features



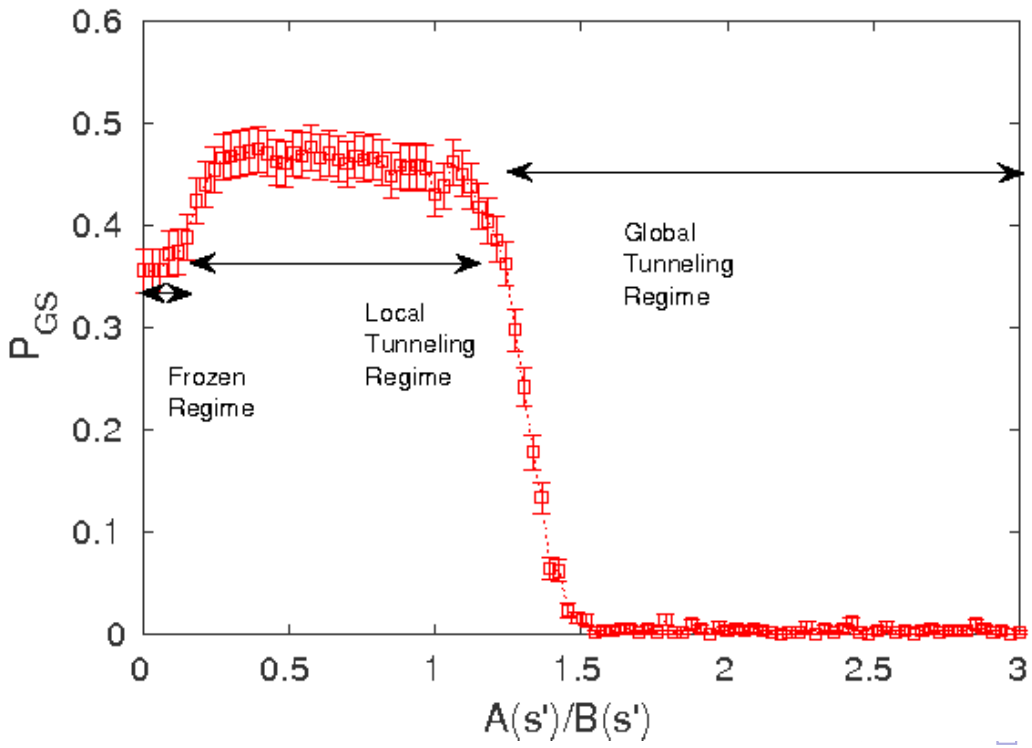
a) QA gets stuck in broad local minima and cannot tunnel to correct minima

b) Classical algorithms can easily explore the broad features, while the annealer can explore the rough ones

c) Even random initialization can improve solution probabilities, may hit rough region by chance

Simple quantum Monte Carlo example

- Toy problem with broad false minima which stymies traditional quantum annealing
- Reverse annealing starting in a random state can find true ground state for large s'via local search
- Behaves like traditional annealing for too small s' values
- see NJP 19, 2, 023024 (2017) for details

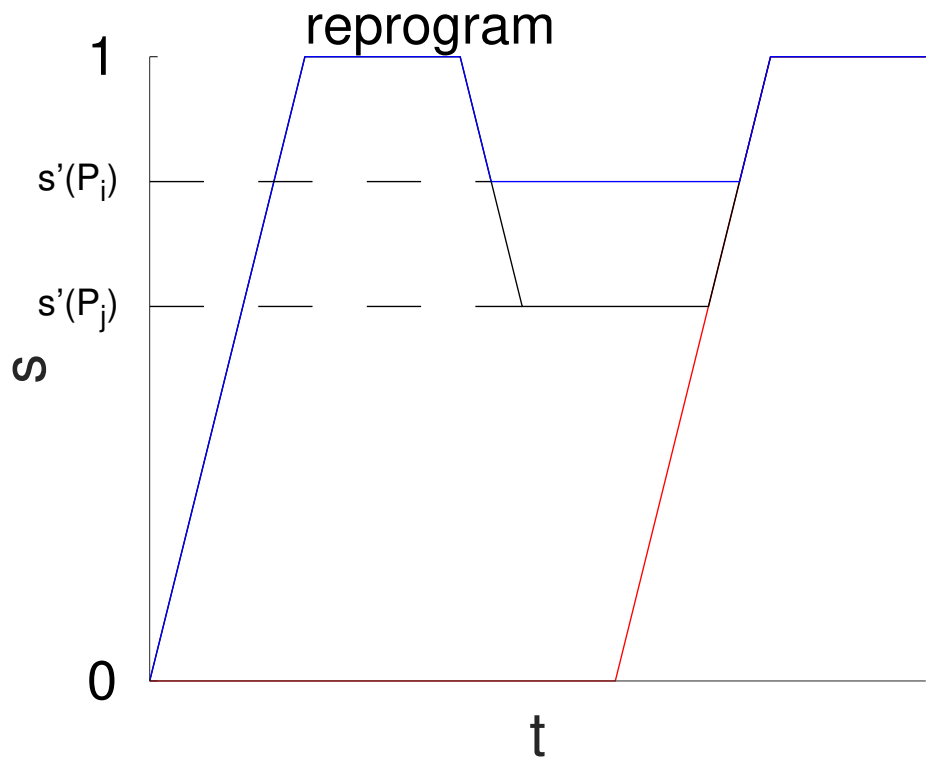


A huge number of possible reverse annealing algorithms

1. Simple version 1: search locally around best classical solution
   - Any improvement is an immediate win
   - But only likely to find solutions 'near' best classical
2. Simple version 2: search locally around randomly chosen state
   - May avoid a broad false minima
3. Monte Carlo like algorithms (see NJP 19, 2, 023024 (2017))
   - Transverse field parameter s' controls tradeoff between exploration and exploitation similar to temperature in Monte Carlo
   - Quantum analogues of many known classical algorithms
4. Genetic algorithms (see arχiv:1609.05875)
   - Compose guess from two or more known solutions
   - Most general version requires more controls than currently available

Extension: anneal qubits differently $\rightarrow$ local uncertainty

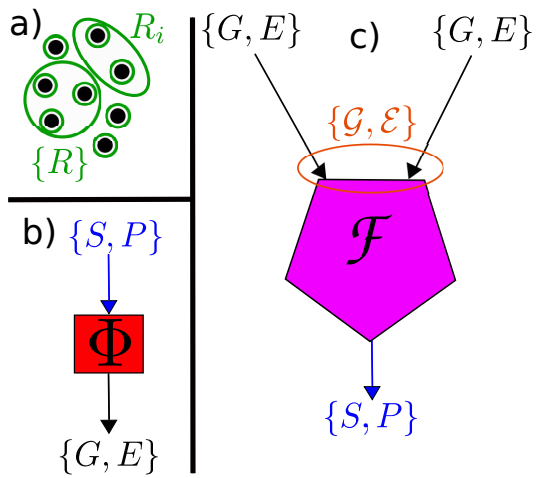
What if we are more sure about some parts of our guess then others? $\rightarrow$ anneal different qubits back to different points



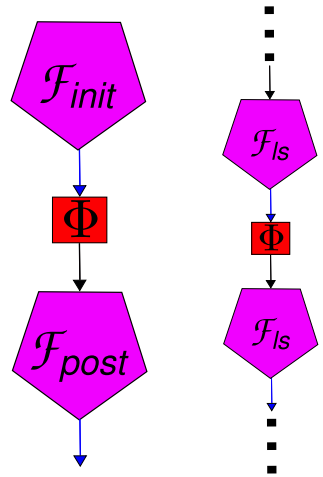
An extreme version of this, which excluded qubits where a value was expected with high certainty has already been done H. Karimi and G. Rosenberg Quantum Inf. Proc. 16(7):166 (2017) and H. Karimi and G. Rosenberg Phys. Rev. E, 96:043312

Representing this graphically: Inference Primitive Formalism

- Represent quantum annealing call as an inference primitive Φ, takes state guess $S \in \{-1, 1\}$ and uncertianty values $P \in [0, 0.5]$, outputs list of states G and energies E
- Processing function $\mathcal{F}$ represents classical processing $\rightarrow$ takes any number (including zero) of annealer outputs (found states G and energies E) and finds new guess S and uncertainty values P
- Easily generalized to multi-body drivers representing uncertainty on clusters of qubits



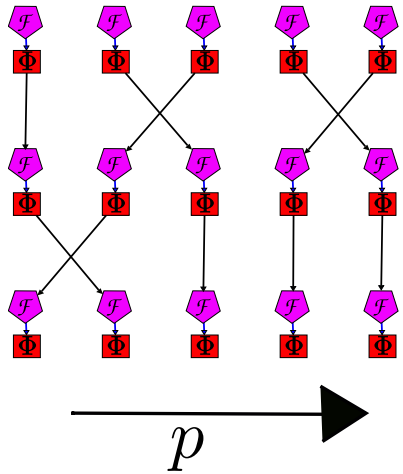
Basic Examples: traditional QA, and repeated local search in this formalism

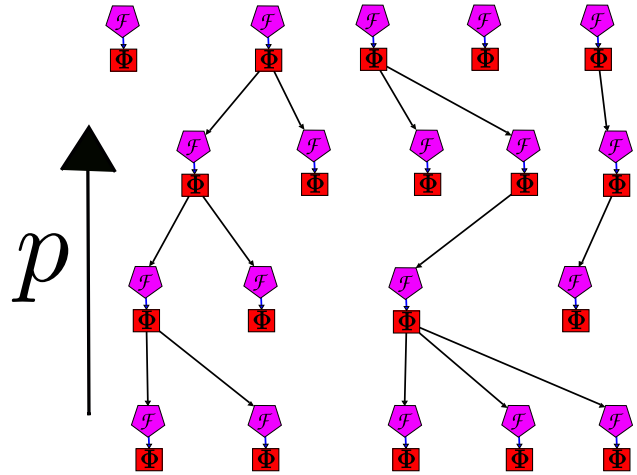


- Traditional QA (left) represented by initialization processing function which takes no inputs and gives complete uncertainty ($P_i = 0.5 \forall i$) on all qubits, followed by post processing function
- Repeated local search (right) from running annealer many times and using the output as an input to the next processing function

More advanced algorithms: Parallel tempering and Population annealing analogues

- Processing function $\mathcal{F}$ returns lowest energy state as guess and gives all qubits the same uncertainty $P_i = p \forall i$
- Assign effective temperature T to each p value and either:
  1. exchange using Parallel tempering rules (left)
     $P_{swap}(i,j) = \min\left[1, \exp\left(\left(\frac{1}{T(i)} - \frac{1}{T(j)}\right)(E_i - E_j)\right)\right]$
  2. kill or replicate states using population annealing rules (right)
     $\bar{N}(E) = \frac{1}{Q}\exp\left(\left(\frac{1}{T_{old}} - \frac{1}{T_{new}}\right)E\right)$





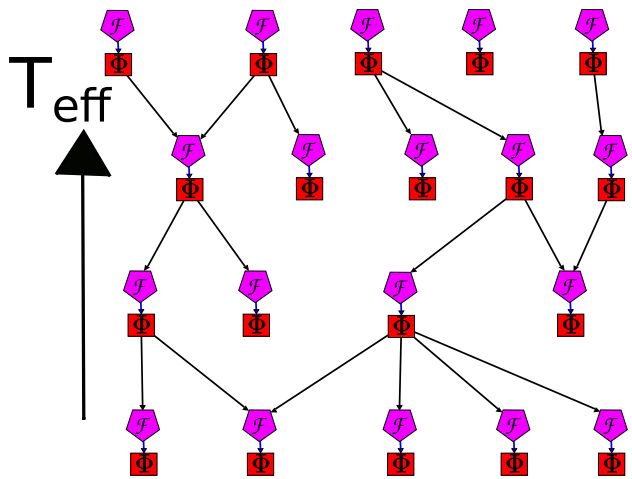
Even more advanced algorithms: Genetic algorithms

- A processing function which takes more than one input is a 'breeding' step of a genetic algorithm
- For instance could be thermally reweighted sum* (u indicates sum over unique states found)

$$S_i = \mathrm{sgn}(\sum_{j=1}^{N_u} G_j^{(u)} \exp(-\frac{E_j^{(u)}}{T_{eff}})),$$

$$P_i = \frac{1}{Z}(\sum_{j=1}^{N_u} \delta_{G_j^{(u)},-S_i} \exp(-\frac{E_j^{(u)}}{T_{eff}}))$$

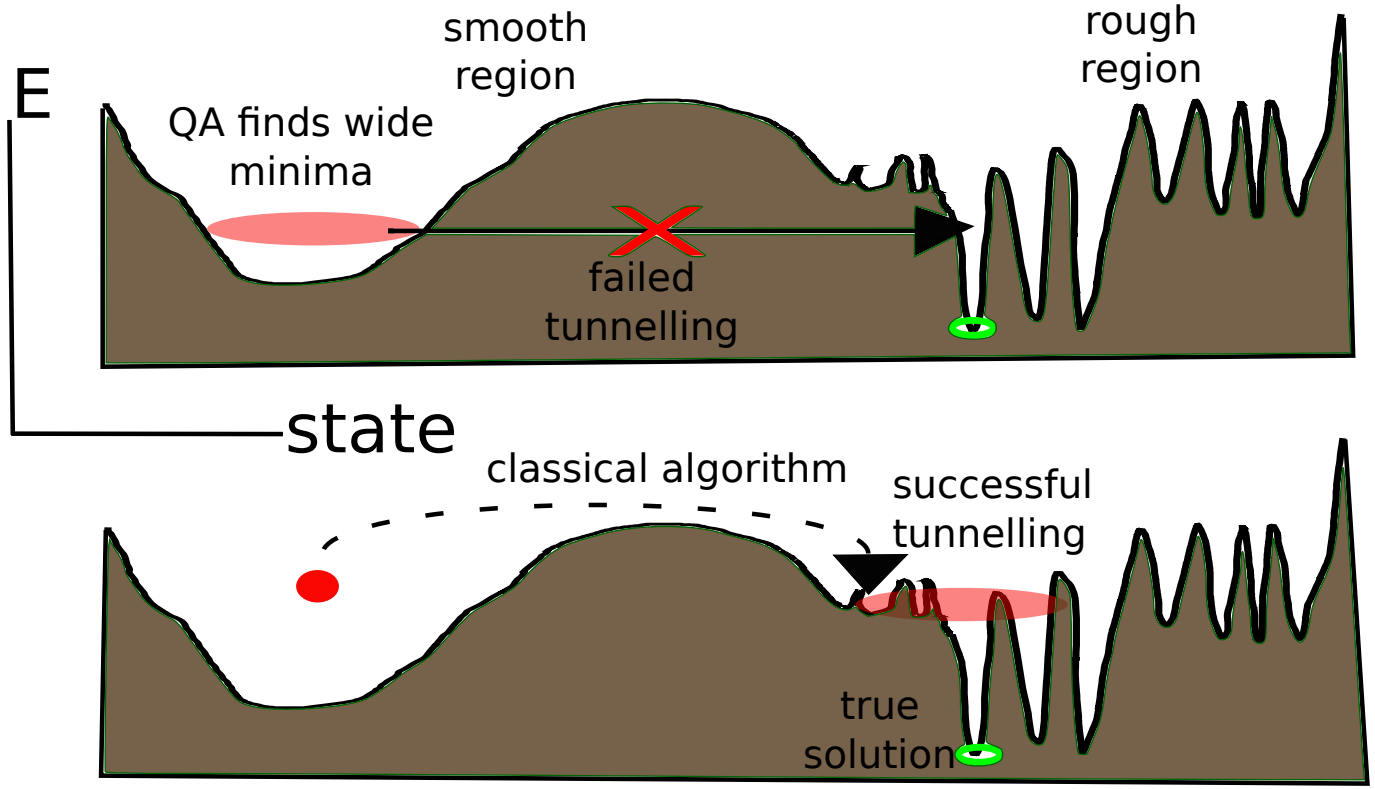
- Could be used to add crossbreeding to Population annealing analogue, as shown below



*see: arχiv:1609.05875 for details

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

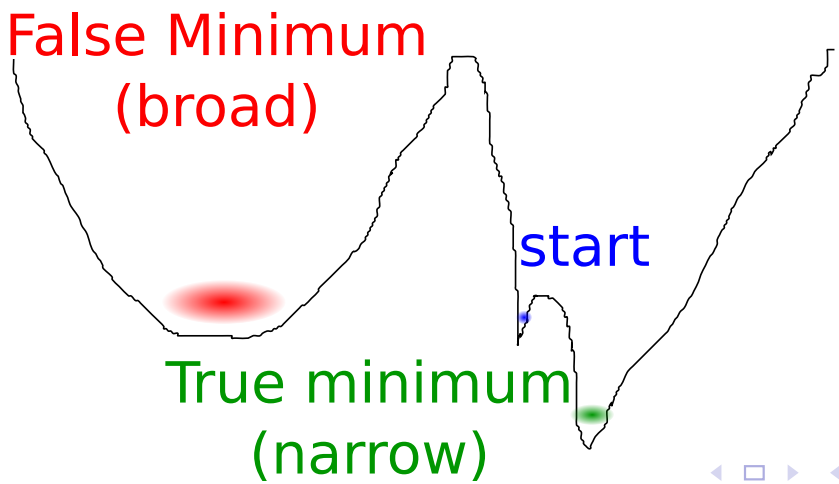
*For related work, see: A. Perdomo-Ortiz, et. al. Quant. Inf. Proc.10(1):33–52, (2011)., T. Graß and M. Lewenstein Phys. Rev. A **95**, 052309 (2017). and Kechedzhi et. al. arχiv:1807.04792 (2018).

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

Need to construct a problem Hamiltonian with the following properties:

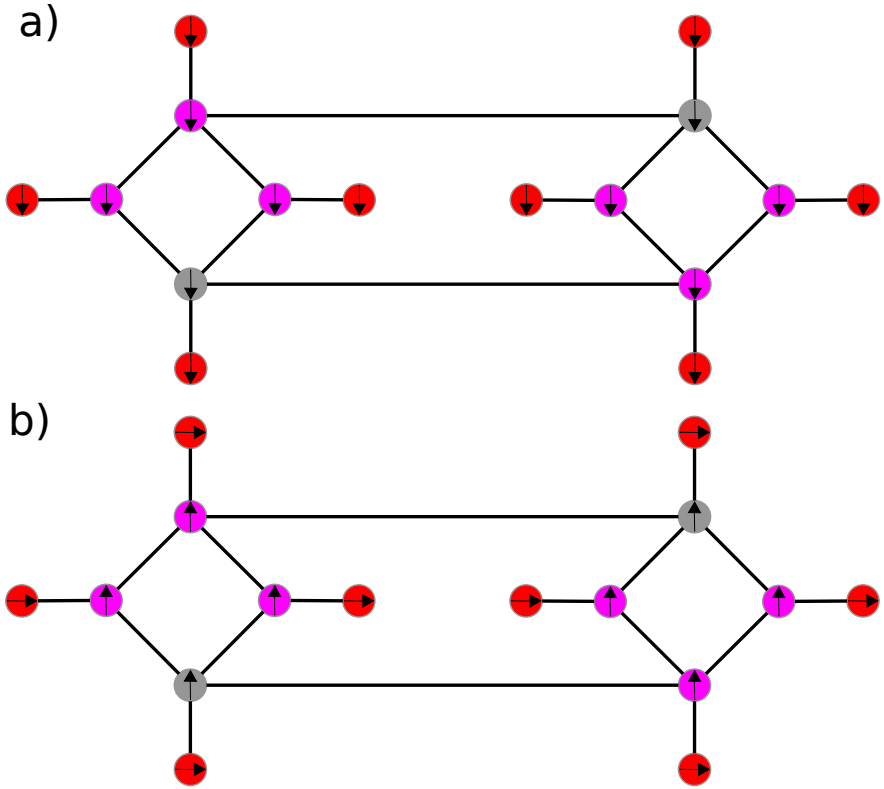
1. Wide false energy minimum which 'tricks' traditional quantum annealing algorithm
2. Relatively narrow true minimum energy
3. Local minimum near true minimum for start state

- 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

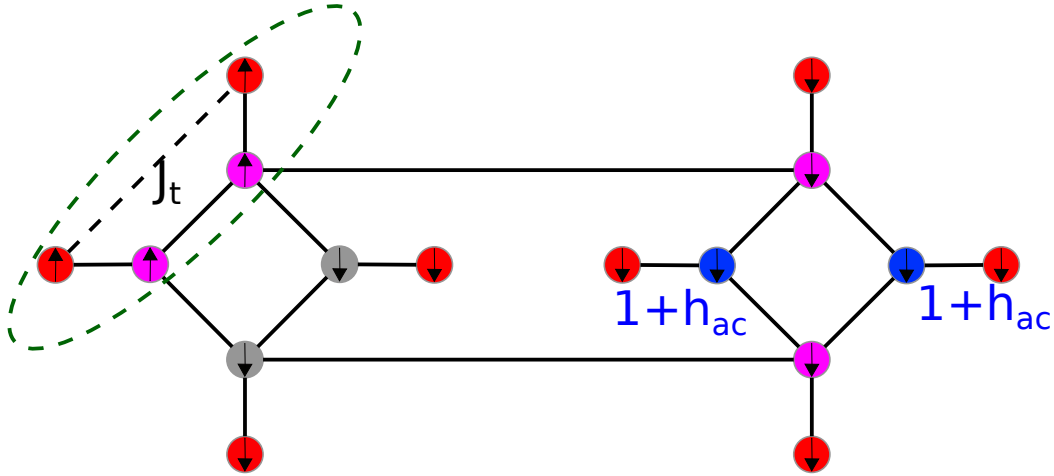
- Hamiltonians with features 1 and 2 are already known: free spin gadgets*
- Start with gadget from N. G. Dickson et. al. Nature Comm. 4, 1903 (2013)



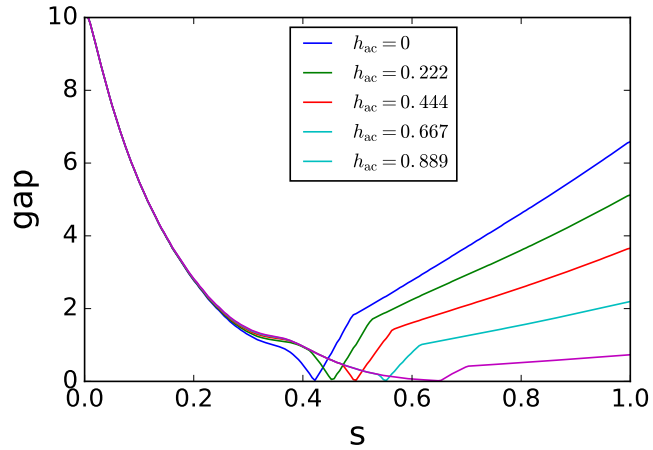
- a: unique ground state (red, h=+1 violet h=-1)
- b: 256-fold degenerate excited state → false minimum

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

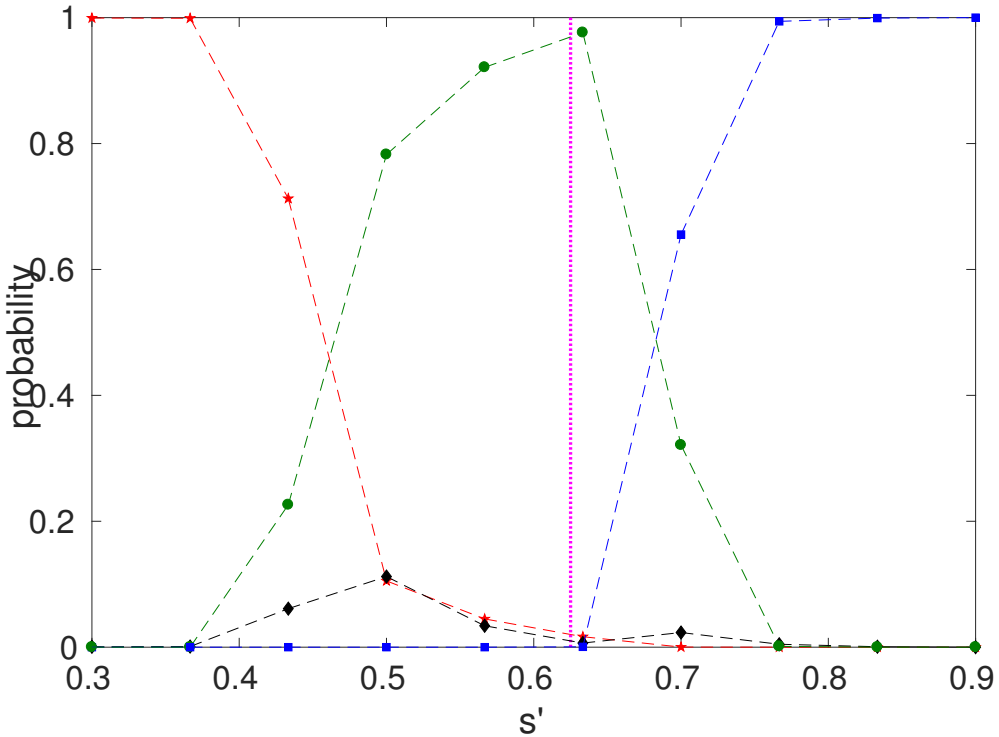
Add local minimum and make tunable

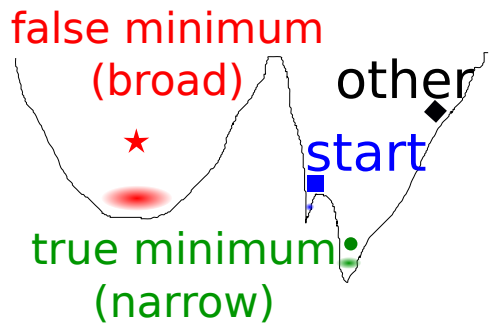


- Starting state shown by arrows, ground state except for circled spins flipped blue field is in - direction
- J_t controls barrier between start state and ground state.
- h_{ac} controls the value of s_{cross}



Experimental results *





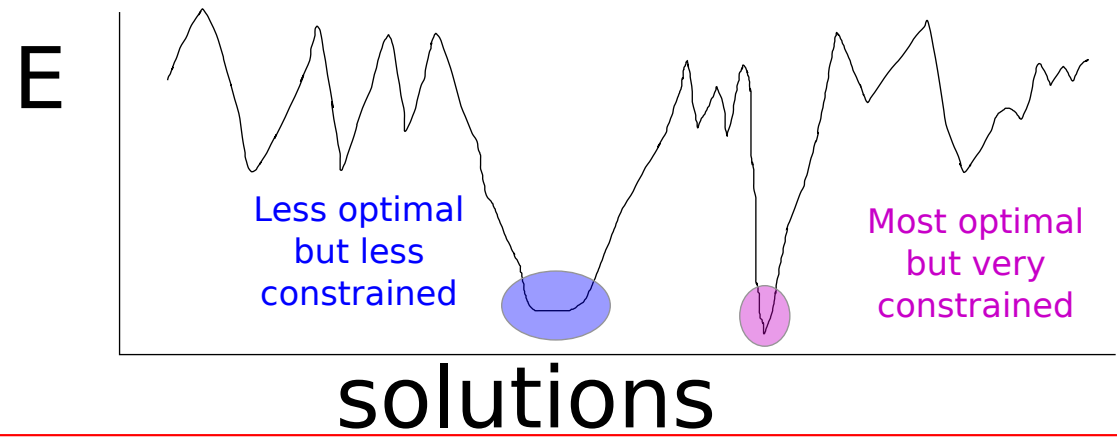
- $J_t = 0.2$, $h_{ac} = 0.95$ level crossing between true GS and false minima at magenta line
- Anneal at maximum allowed rate, wait time (τ) of $20\mu S$
- Frozen in starting state for small s', find true minimum at moderate s', trapped for large s'

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

Enhancing Robustness of Solutions using reverse annealing

Using quantum annealers to find solutions which are robust in the sense that they can be adjusted to a modified problem definition at little or no energy cost

- Simplest way this manifests is free spins $\rightarrow$ annealers known to find this feature
- If a good solution is already known, can we use an annealer to trade optimality for robustness?



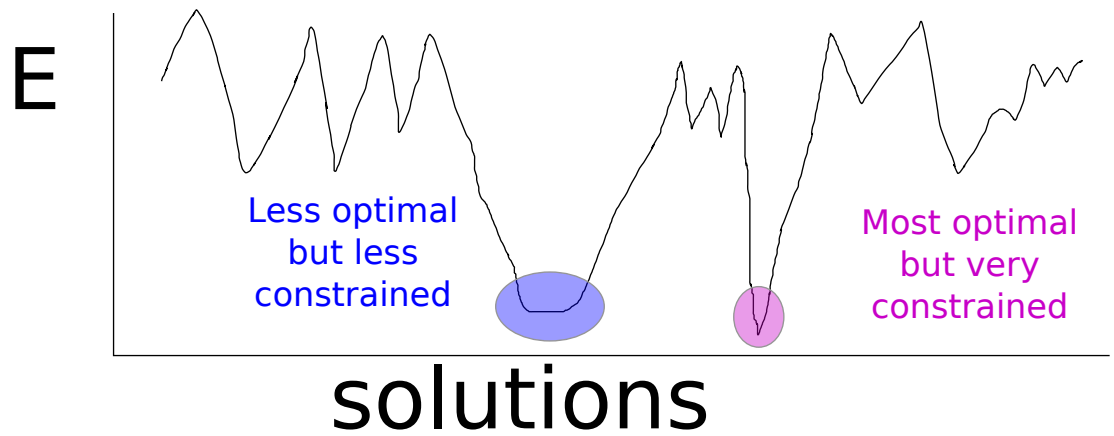
Joint work with Simon Benjamin, funded by BP and EPSRC

bp

EPSRC
Pioneering research and skills

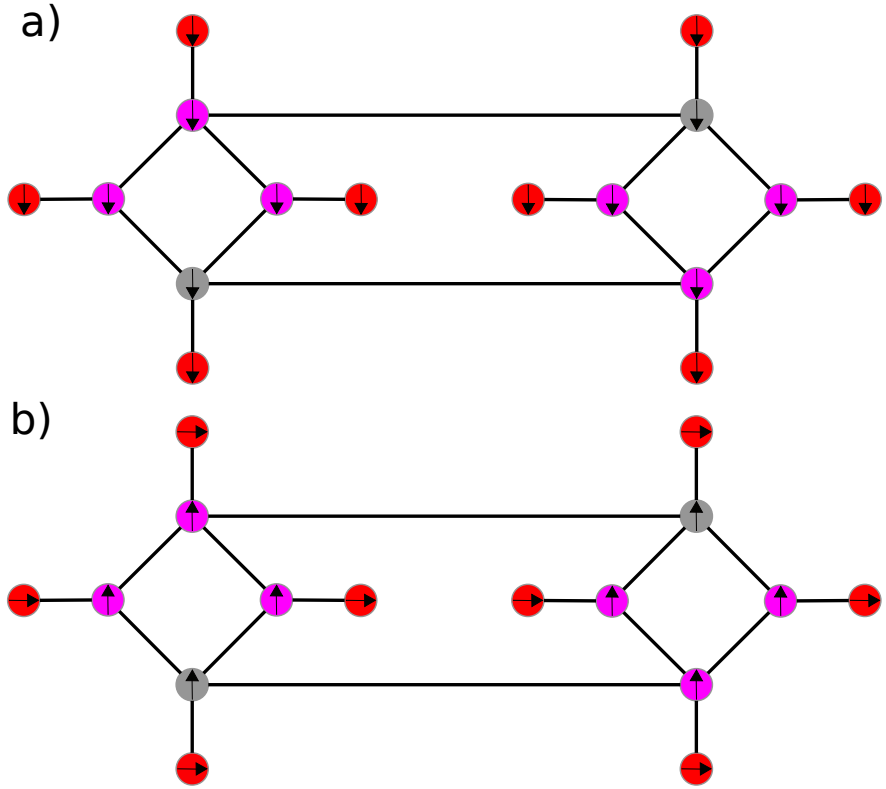
Why might we want this?

- Adjust solution if we later learn that our problem definition was slightly incorrect
- Penalty terms which are too expensive to encode on annealer could be implemented by adjustments in post-processing
  - Global non-linear constraints for instance are expensive to map
- Find 'template' solution which can be adjusted to solve many similar but not identical problems



A simple (motivational) example

Consider the same 16 qubit gadget from N. G. Dickson et. al. Nature Comm. 4, 1903 (2013) :



- a is the ground state but
- A D-Wave 2000Q with $1,280,000$ $5\mu s$ runs finds b $1,277,824$ times and a only 17 times

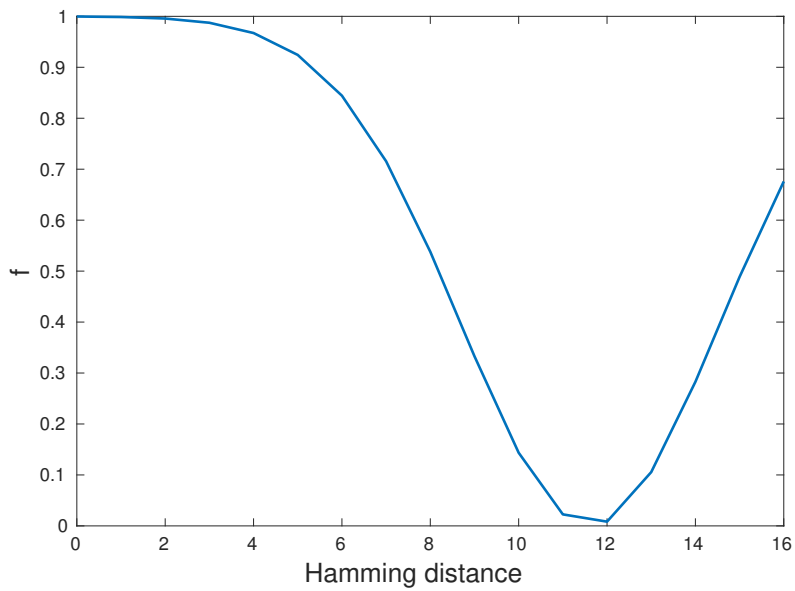
Simple test: add global penalty and do greedy search

Global penalty:

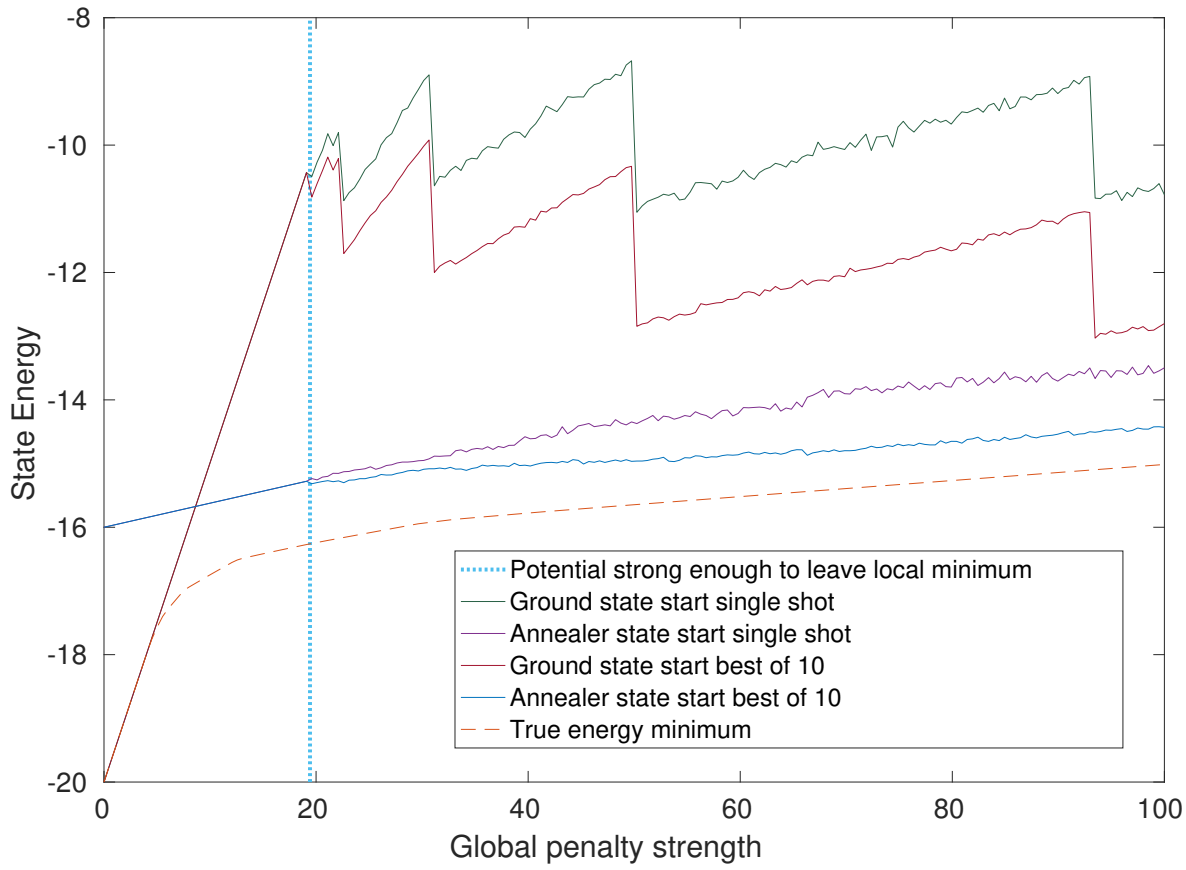
$$E(q) = E_{\mathrm{Ising}}(q) + g\, f[\mathfrak{h}(q, r)]$$

where:

- q is a bitstring representing the state
- g is the strength of the penalty
- $\mathfrak{h}$ is Hamming distance
- r is a random bitstring
- f is a single variable function:



Starting in true ground state vs. state annealer finds



The large degeneracy in the state the annealer finds allows for much more effective adjustment $\rightarrow$ higher energy but more robust

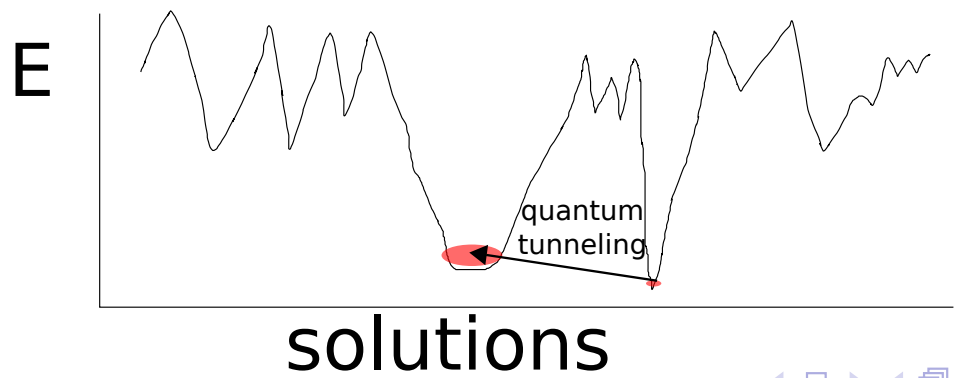
Reverse annealing to trade off optimality and robustness

Hypothetical situation:

- Already know the most optimal (planted) solution
- But we want more flexibility
- Are willing to 'pay' some optimality for a more flexible solution

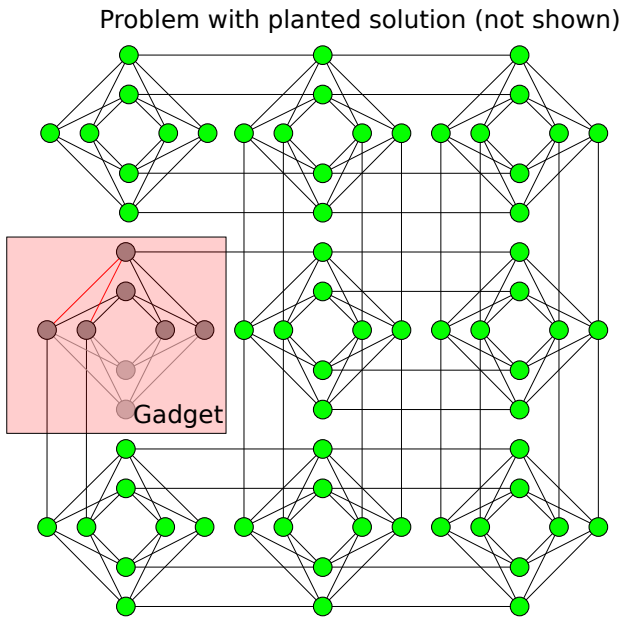
Algorithm:

1. Start reverse annealing in planted solution
2. Search over a set range
3. Repeat many times
4. Keep most optimal solutions with a given number of gadgets 'free'



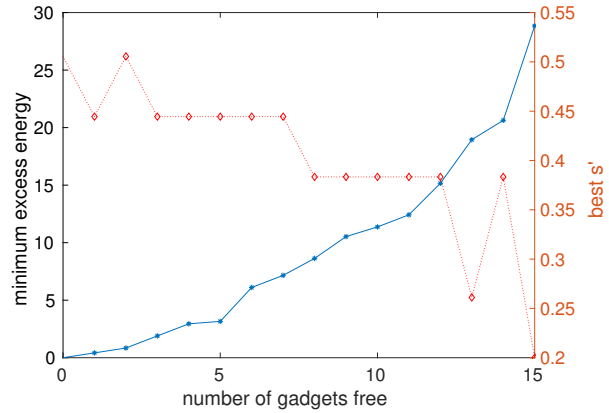
Free spin gadgets

- Use planted solution method from Hen et. al. Phys. Rev. A 92, 042325 (2015) to make 'hard' problems with all -1 and all $+1$ ground state
- Before constructing replace some unit cells with free spin gadgets
  - All spins fixed if 'outside' spins agree
  - Become free if they do not (but energy unchanged)
  - Energy penalty because has to leave planted solution

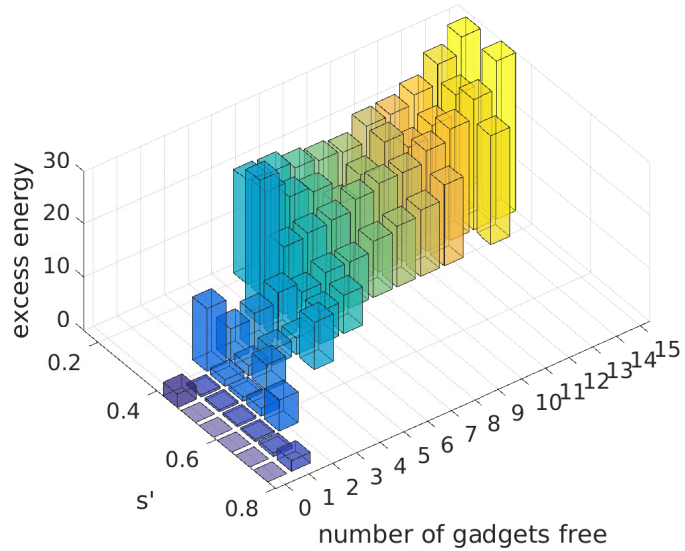


The tradeoff

What is the best excess energy we can find with a given number of gadgets free?

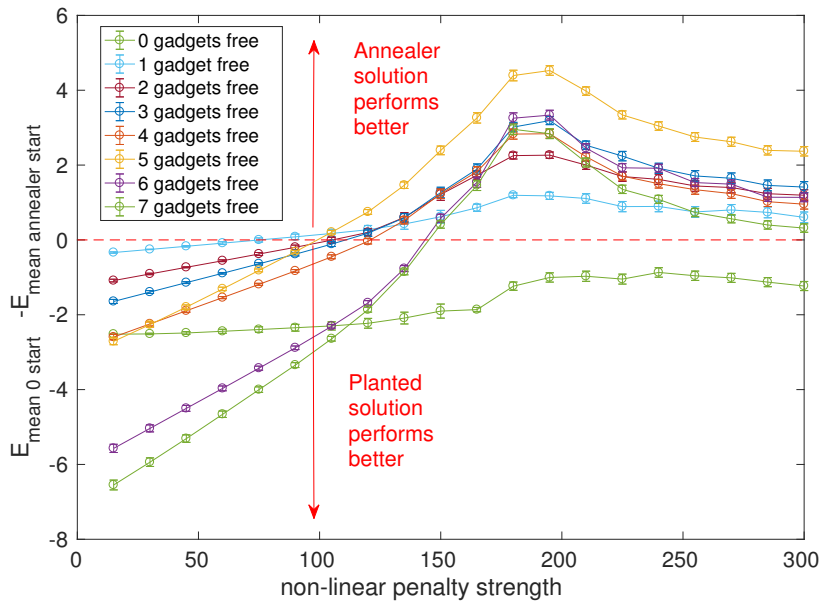


(smallest s' value taken in the event of a tie)



Putting new solutions to the test

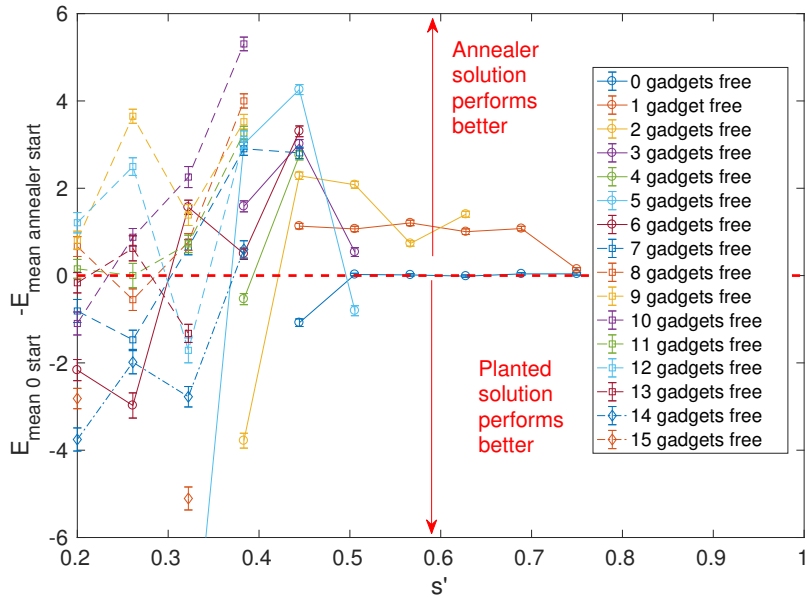
- Choose $s' = 0.4444$ dataset $\rightarrow$ contains some of the best solutions
- Choose $10,000$ different instances of non-linear penalties
- Perform greedy search in each case and compare with planted solution
- Compare for different penalty strengths



Crossover where annealer solution becomes the better choice

Comparing performance at different s' values

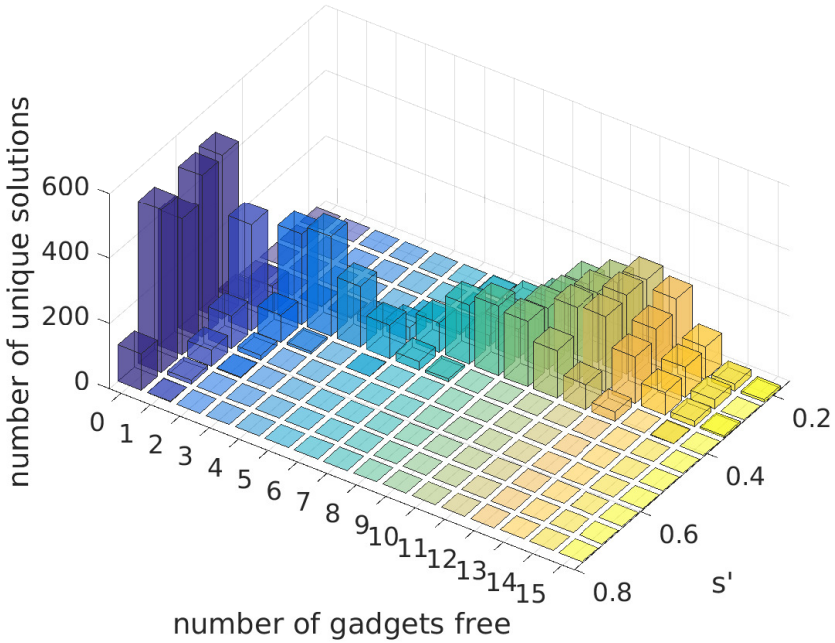
- Choose non-linear penalty strength of 195
- Examine performance of solutions found at different values of s'



- Best performance at intermediate values of s'
- Smaller values of s' better for finding solutions with smaller number of 'free' gadgets

What's next?

- Sample over more problem instances
- Run test with 'no free variable' gadgets for comparison
- More data analysis



Take away messages

Quantum annealing and gate base quantum computing

- Advantages and disadvantages to each
- Quantum annealing technology more mature

Hybrid computing with annealers

- Many more algorithmic possibilities using reverse annealing to search locally in solution space
- feature actually exists on D-Wave devices, can actually be tested *experimentally* right now

Solution robustness

- Optimality isn't the only concern in the real world
- Natural dynamics of quantum annealers mean they can be used to trade off optimality for robustness