Fun with Adiabatic Quantum Computing

Jason Chang

NSD Staff Meeting
March 2020

What is a universal quantum computer?

DiVincenzo Criteria

- A scalable system with well-characterized qubits
- Ability to initialize state of qubits to a simple fiducial state, such as $|000\ldots\rangle$
- A universal set of quantum gates
- Decoherence time much longer than gate-operation time
- A qubit specific measurement

What is a universal quantum computer?

DiVincenzo Criteria

- A scalable system with well-characterized qubits
- Ability to initialize state of qubits to a simple fiducial state, such as $|000\ldots\rangle$
- A universal set of quantum gates
- Decoherence time much longer than gate-operation time
- A qubit specific measurement

What is a qubit?

- 2-level system with an energy gap
  - There is a state 0 and state 1
  - Ground-state and excited-state (gap)

How does this compare to a classical bit?

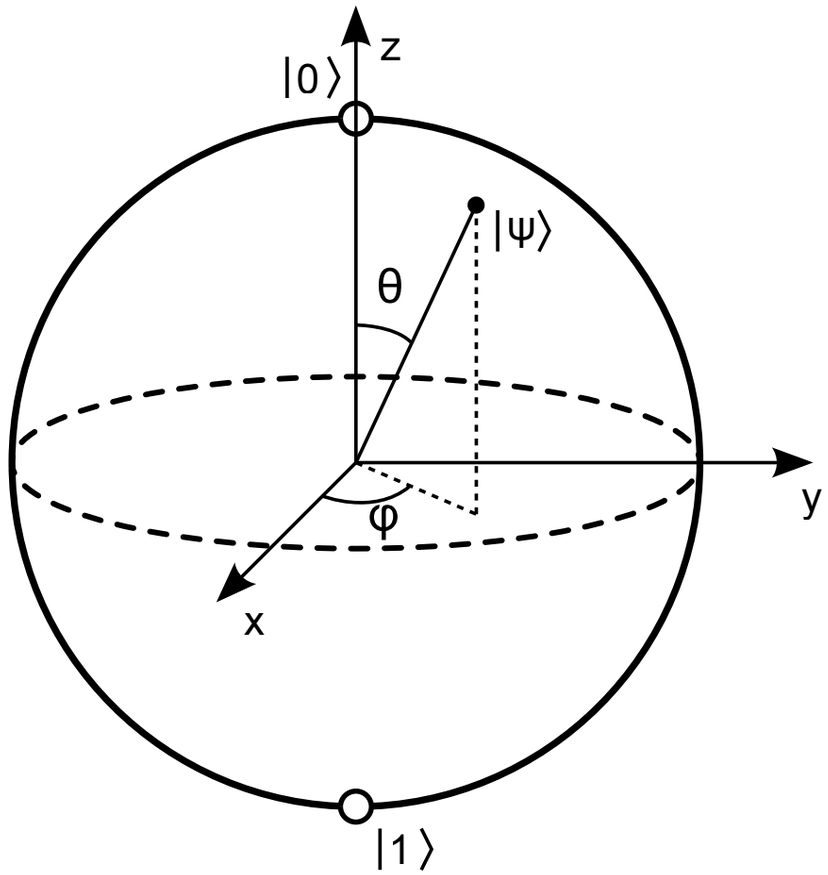
- A qubit can be anywhere on the 3-sphere surface

$$|\psi\rangle = c_1 |0\rangle + c_2 |1\rangle$$ **qubit superposition**

$$|c_1|^2 + |c_2|^2 = 1$$ **total probability 1 of getting 0 or 1 upon measurement**

$$|0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$ **matrix representation of qubit**

Comment: can generalize to *n* - level system



Bloch sphere
Visualization of 2-level system

What is a universal quantum computer?

DiVincenzo Criteria

- A scalable system with well-characterized qubits
- Ability to initialize state of qubits to a simple fiducial state, such as $|000\ldots\rangle$
- A universal set of quantum gates
- Decoherence time much longer than gate-operation time
- A qubit specific measurement

Need to initialize the set of qubits before a calculation

$$\left|000\ldots0\right\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \otimes \ldots \begin{pmatrix}1\\0\end{pmatrix}$$

With *n* qubits, the explicit matrix representation has length 2^n.
The Hilbert-space of *n* qubits is exponentially large.

The above example is a **pure state** (every qubit is at state 0).
A set of qubits can be in a **mixed state**

$$\left|\psi\right\rangle = \sum_{i_k=0,1} \alpha_{i_1,i_2,\ldots,i_n} \left|i_1 i_2 \ldots i_n\right\rangle$$

An algorithm can in parallel evaluate exponential number of possibilities.

What is a universal quantum computer?

DiVincenzo Criteria

- A scalable system with well-characterized qubits
- Ability to initialize state of qubits to a simple fiducial state, such as $|000\ldots\rangle$
- A universal set of quantum gates
- Decoherence time much longer than gate-operation time
- A qubit specific measurement

Smallest set of gates that can rotate a qubit to any point on the Bloch sphere?

Example: Infinitesimal rotations of SU(2) (special *unitary* group)

$dR = e^{-i\tau_a\theta^a} \simeq 1 - i\tau_a\theta^a$ **unitary rotation, reversible with exp(+iτθ)**

$$\tau_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \tau_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \tau_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Pauli matrices
x, y forms mixed states
z projects quantum to classical state

For circuit QC this + identity is the complete single qubit gate-operations.

Tensor-products of 2 of the following {I, τ_x, τ_y, τ_z} forms all 2-qubit gate-operations.
From here build CNOT (quantum XOR), AND, etc… gates.

I will not go much into circuit-based QC today.

What is a universal quantum computer?

DiVincenzo Criteria

- A scalable system with well-characterized qubits
- Ability to initialize state of qubits to a simple fiducial state, such as $|000\ldots\rangle$
- A universal set of quantum gates
- Decoherence time much longer than gate-operation time
- A qubit specific measurement

Some very practical concerns

QC needs to be coherent

- Coherence implies the system does not lose information (probability is always 1).
- Ideal quantum systems are completely isolated, but not real computers.
  - Interaction with environment can leak information and lose “quantumness”.
- Coherence time >> gate-operation time for the QC to be useful.

Need to be able to read the solution

- The state of each qubit needs to be read.
- For classical problems, the solution is a pure state.
- Tricky if QC is used for quantum simulations (solution is a 2^n superposition state).

Paradigms of quantum computing

Universal Quantum Computers

- **Circuit-based QC (quantum gate arrays)**
- **Topological QC (braiding anyons to get phase information)**

- **Adiabatic QC**

- **Quantum annealing**

In my opinion these names are partially motivated by marketing also…

What are some examples of different quantum computers?

Circuit-based universal quantum computers

- Google, IBM, Rigetti, Amazon, many DOE funded quantum testbeds, etc…

Topological quantum computer

- Microsoft

Quantum annealers* (QA)

- D-Wave

* QA as produced by D-Wave can *only* solve classical problems. Adiabatic QC is in principle polynomially equivalent to other universal QC paradigms.

Adiabatic theorem

A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it **slowly enough** and if there is a **gap** between the eigenvalue and the rest of the Hamiltonian's spectrum.

Note: The inverse-gap dictates algorithm run-time

Is there a quantum computer that can prepare a trivial ground state and adiabatically evolve the quantum system to a non-trivial final Hamiltonian?

If so we can calculate complicated problems.

Note: NO gate operations.

What does a quantum annealer solve then?

Adiabatic quantum computing

$$H(t) = A(t)H^{\text{init}} + B(t)H^{\text{fin}}$$

$$A(0) = 1 \qquad B(0) = 0$$
$$A(T) = 0 \qquad B(T) = 1$$

Quantum annealing

$$H^{\text{init}} = \sum_i \sigma_i^x \qquad H^{\text{fin}} = \sum_{ij} J_{ij}\sigma_i^z\sigma_j^z + \sum_i h_i\sigma_i^z$$

Condensed Matter > Statistical Mechanics

Quantum Annealing in the Transverse Ising Model

Tadashi Kadowaki, Hidetoshi Nishimori

(Submitted on 25 Apr 1998)

We introduce quantum fluctuations into the simulated annealing process of optimization problems, aiming at faster convergence to the optimal state. Quantum fluctuations cause transitions between states and thus play the same role as thermal fluctuations in the conventional approach. The idea is tested by the transverse Ising model, in which the transverse field is a function of time similar to the temperature in the conventional method. The goal is to find the ground state of the diagonal part of the Hamiltonian with high accuracy as quickly as possible. We have solved the time-dependent Schrödinger equation numerically for small size systems with various exchange interactions. Comparison with the results of the corresponding classical (thermal) method reveals that the quantum annealing leads to the ground state with much larger probability in almost all cases if we use the same annealing schedule.

This particular industry was inspired by this paper

Solving a programmable spin-glass model

$$H^{\text{init}} = \sum_i \sigma_i^x \qquad H^{\text{fin}} = \sum_{ij} J_{ij}\sigma_i^z\sigma_j^z + \sum_i h_i\sigma_i^z$$

The purpose is to solve the final Hamiltonian
One can map ALL classical NP-hard optimization problems to this model
(this is why it is interesting)

If we drop the transverse field…

- if J is a constant, we recover the *n*-dimensional **Ising Model**
  - beyond 1-dimension we get (anti-)ferromagnetic phase transitions
- If J is random, we get the **spin-glass model**

- **If J and h is programmable, we have a computer**

Solving a programmable spin-glass model

$$H^{\text{init}} = \sum_i \sigma_i^x \qquad H^{\text{fin}} = \sum_{ij} J_{ij}\sigma_i^z\sigma_j^z + \sum_i h_i\sigma_i^z$$

The purpose is to solve the final Hamiltonian
One can map ALL classical NP-hard optimization problems to this model
(this is why it is interesting)

If we drop the transverse field…

- if J is a constant, we recover the *n*-dimensional **Ising Model**
  - beyond 1-dimension we get (anti-)ferromagnetic phase transitions
- If J is random, we get the **spin-glass model**

- **If J and h is programmable, we have a computer**

These systems are solvable with **thermal** (classical) phase transitions
OR
at zero temperature **quantum** phase transitions

Solving a programmable spin-glass model

$$H^{\text{init}} = \sum_i \sigma_i^x \qquad H^{\text{fin}} = \sum_{ij} J_{ij}\sigma_i^z\sigma_j^z + \sum_i h_i\sigma_i^z$$

The purpose is to solve the final Hamiltonian
One can map ALL classical NP-hard optimization problems to this model
(this is why it is interesting)

If we drop the transverse field…

- if J is a constant, we recover the *n*-dimensional **Ising Model**
  - beyond 1-dimension we get (anti-)ferromagnetic phase transitions
- If J is random, we get the **spin-glass model**

- **If J and h is programmable, we have a computer**

These systems are solvable with **thermal** (classical) phase transitions
OR
at zero temperature **quantum** phase transitions

This is why quantum annealing is (and is not) a quantum computer

Systems of equations

Solutions to systems of equations are ubiquitous in science, engineering and mathematics.

Problem definition

Linear system of equations

index notation

$$P_{ij}^{(1)} x_j = P_i^{(0)}$$

matrix notation

$$\begin{pmatrix} P_{00}^{(1)} & P_{01}^{(1)} \\ P_{10}^{(1)} & P_{11}^{(1)} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} P_0^{(0)} \\ P_1^{(0)} \end{pmatrix}$$

Polynomial system of equations

$$P_{ij}^{(1)} x_j + P_{ijk}^{(2)} x_j x_k + \cdots = P_i^{(0)}$$

High-dimensional function spaces are challenging for classical algorithms.

Can quantum computing tackle this?

Examples of linear equations

Solving the Dirac equation for propagator for example

$$\not{D}\psi = S$$

$$P^{(1)}x = P^{(0)}$$

Linear and polynomial regression can be mapped to a system of equations.

Correlated least-squares loss function

$$\chi^2 \equiv \left[F(x,\{p\}) - \langle y \rangle\right]_i S_{ij}^{-1} \left[F(x,\{p\}) - \langle y \rangle\right]_j$$

Fit function

$$F(x_i,\{p\}) = \sum_{n=1}^{P} p_n f_n(x_i)$$

Minimizing loss function (assuming **linear**) yields a linear system of equations

$$P^{(1)} = \begin{pmatrix} f_0(x_i)S_{ij}^{-1}f_0(x_j) & \dots & f_0(x_i)S_{ij}^{-1}f_P(x_j) \\ \vdots & \ddots & \vdots \\ f_P(x_i)S_{ij}^{-1}f_0(x_j) & \dots & f_P(x_i)S_{ij}^{-1}f_P(x_j) \end{pmatrix} \quad P^{(0)} = \begin{pmatrix} f_0(x_i)S_{ij}^{-1}y_j \\ \vdots \\ f_P(x_i)S_{ij}^{-1}y_j \end{pmatrix} \quad P^{(1)}x = P^{(0)}$$

Mapping systems of equations to spin-glass

Rewrite system of equations $P_{ij}^{(1)} x_j + P_{ijk}^{(2)} x_j x_k + \cdots = P_i^{(0)}$

into a minimization problem $\mathrm{Min}\left[\frac{1}{2} P_{ij}^{(1)} x_i x_j + \frac{1}{3} P_{ijk}^{(2)} x_i x_j x_k + \ldots - P_i^{(0)} x_i\right]$

Map *R*-spin chain to a superposition of 2^R decimal numbers

$$x_i = a_i \sum_{r=0}^{R-1} 2^r \psi_{r,i} + b_i \qquad \text{where} \qquad \psi \in [0,1] \quad \text{(QUBO definition)}$$

For a system of *N* linear equations the mapping is explicitly

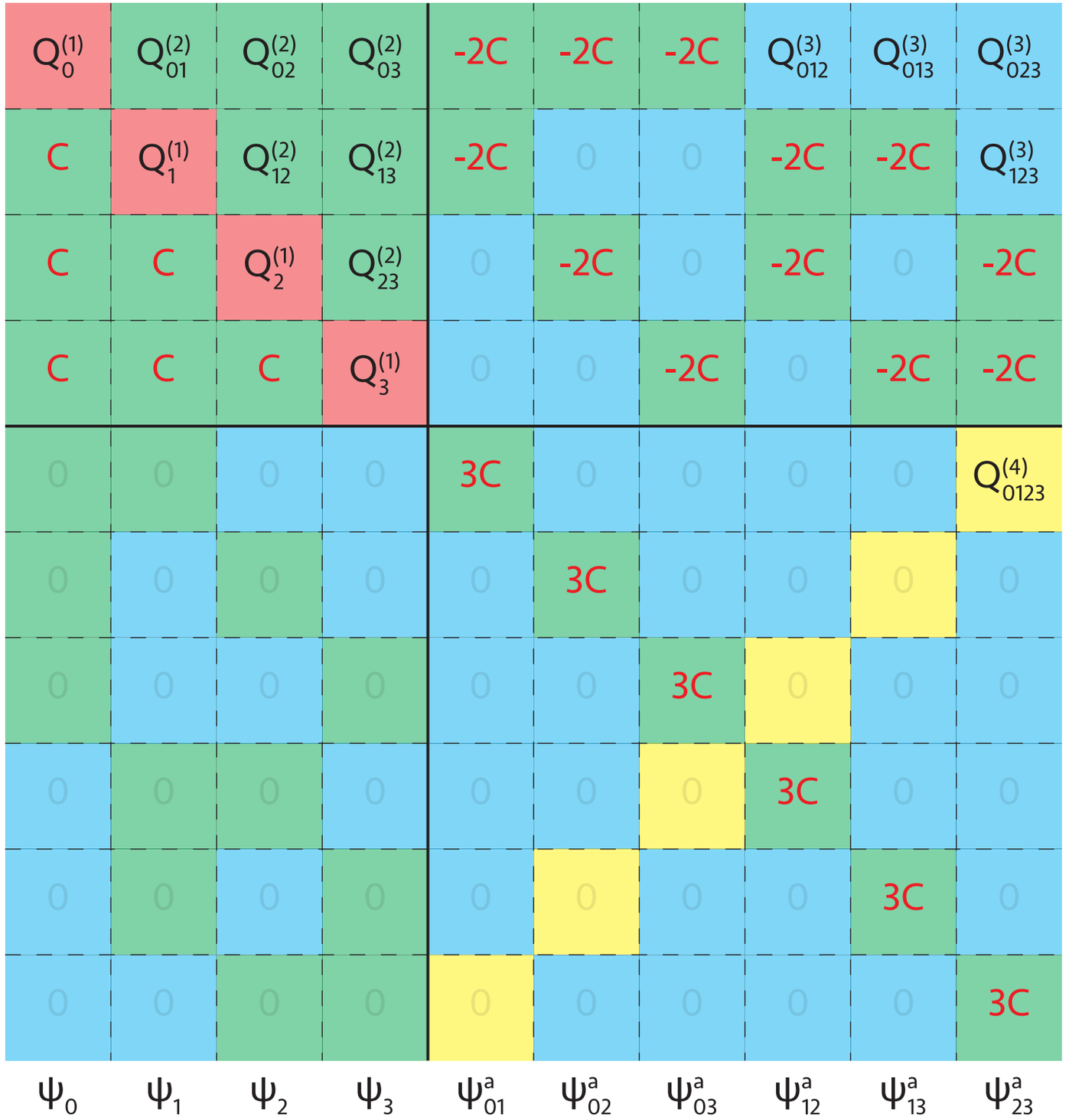
$$\begin{aligned} H^{\mathrm{QUBO}}(\psi) = & \begin{pmatrix} \psi_{1,1} \\ \vdots \\ \psi_{R,N} \end{pmatrix}^T \left[\begin{pmatrix} a_1^2 P_{11}^{(1)} & \cdots & a_1 a_N P_{1N}^{(1)} \\ \vdots & \ddots & \vdots \\ a_N a_1 P_{N1}^{(1)} & \cdots & a_N^2 P_{NN}^{(1)} \end{pmatrix} \otimes \begin{pmatrix} 2^0 2^0 & \cdots & 2^0 2^{R-1} \\ \vdots & \ddots & \vdots \\ 2^{R-1} 2^0 & \cdots & 2^{R-1} 2^{R-1} \end{pmatrix} \right. \\ & \left. -2 \begin{pmatrix} a_1 \left(P_1^{(0)} - b_1 \sum_i P_{1i}^{(1)} \right) & & \\ & \ddots & \\ & & a_N \left(P_N^{(0)} - b_N \sum_i P_{Ni}^{(1)} \right) \end{pmatrix} \otimes \begin{pmatrix} 2^0 & & \\ & \ddots & \\ & & 2^{R-1} \end{pmatrix} \right] \begin{pmatrix} \psi_{1,1} \\ \vdots \\ \psi_{R,N} \end{pmatrix} \\ = & \Psi_i Q_{ij} \Psi_j \end{aligned}$$

In principle a direct solve (global minimum) independent of condition number or sparseness.
Quantum computers can simultaneously evaluate an exponential number of solutions.
Note: PRL 103 150502 (2009) (HHL algorithm) outlines linear solver for gate-model QC.

Systems of Quadratic Equations

Solve toy example:

$$0 = 2x_0^2 + 3x_0x_1 + x_1^2 + 2x_0 + 4x_1 - 51$$
$$0 = x_0^2 + 2x_0x_1 + 2x_1^2 + 3x_0 + 2x_1 - 46$$



Systems of Quadratic Equations

Solve toy example:

$$0 = 2x_0^2 + 3x_0x_1 + x_1^2 + 2x_0 + 4x_1 - 51$$

$$0 = x_0^2 + 2x_0x_1 + 2x_1^2 + 3x_0 + 2x_1 - 46$$

Send to Quantum Annealer

$$H^{\text{QUBO}} = \left[P^{(0)} + P^{(1)}\left(\cdot\mathcal{B} + \circ\mathcal{A}\otimes\mathcal{R}\cdot\psi\right) + P^{(2)}\left(\cdot\mathcal{B} + \circ\mathcal{A}\otimes\mathcal{R}\cdot\psi\right)^2 + \ldots\right]^2$$

$Q_0^{(1)}$	$Q_{01}^{(2)}$	$Q_{02}^{(2)}$	$Q_{03}^{(2)}$	-2C	-2C	-2C	$Q_{012}^{(3)}$	$Q_{013}^{(3)}$	$Q_{023}^{(3)}$
C	$Q_1^{(1)}$	$Q_{12}^{(2)}$	$Q_{13}^{(2)}$	-2C	0	0	-2C	-2C	$Q_{123}^{(3)}$
C	C	$Q_2^{(1)}$	$Q_{23}^{(2)}$	0	-2C	0	-2C	0	-2C
C	C	C	$Q_3^{(1)}$	0	0	-2C	0	-2C	-2C
0	0	0	0	3C	0	0	0	0	$Q_{0123}^{(4)}$
0	0	0	0	0	3C	0	0	0	0
0	0	0	0	0	0	3C	0	0	0
0	0	0	0	0	0	0	3C	0	0
0	0	0	0	0	0	0	0	3C	0
0	0	0	0	0	0	0	0	0	3C
ψ_0	ψ_1	ψ_2	ψ_3	ψ^a_{01}	ψ^a_{02}	ψ^a_{03}	ψ^a_{12}	ψ^a_{13}	ψ^a_{23}
x_0		x_1		x_0^2		x_0x_1		x_1^2	

Systems of Quadratic Equations

Solve toy example:

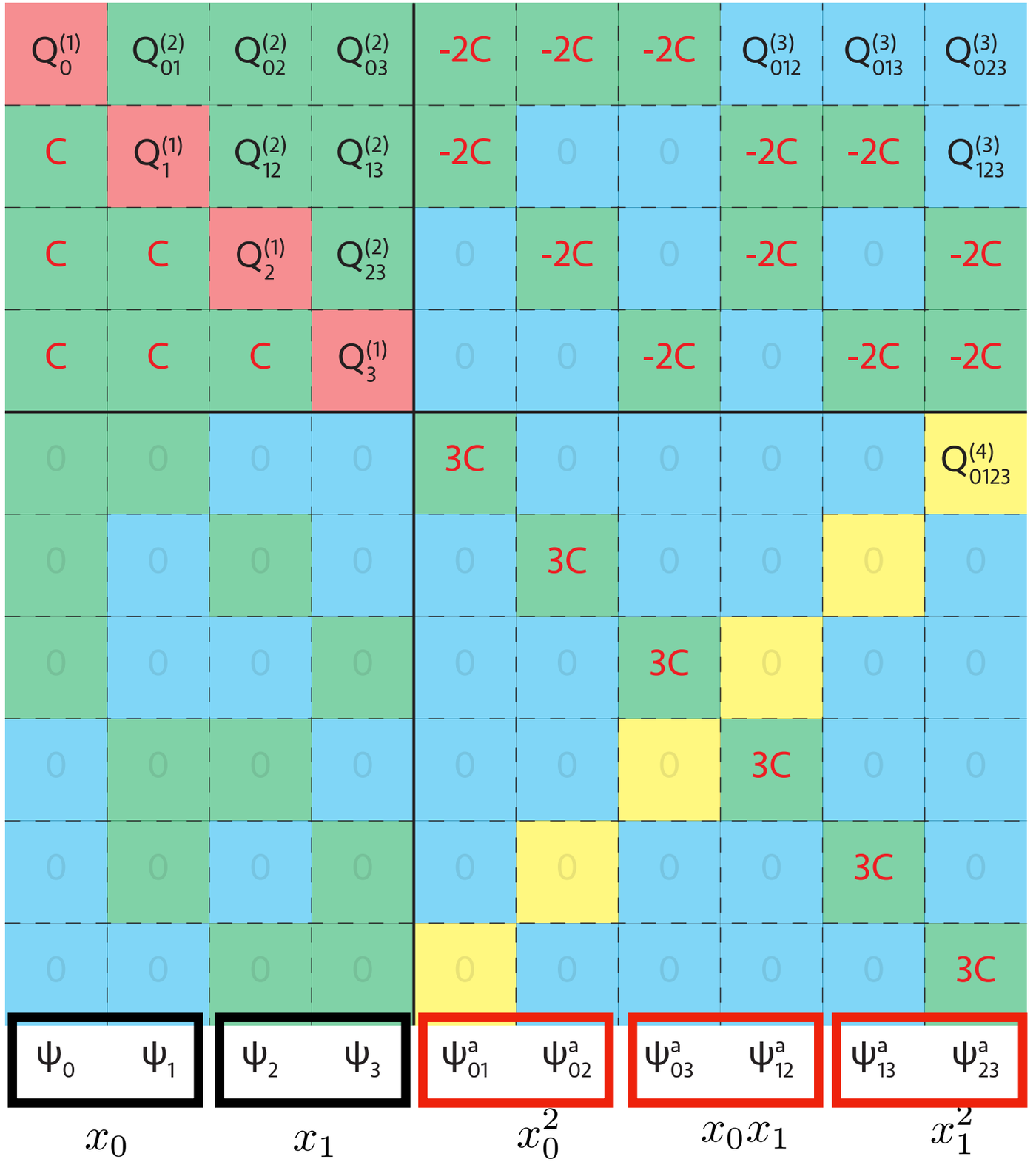
$$0 = 2x_0^2 + 3x_0x_1 + x_1^2 + 2x_0 + 4x_1 - 51$$
$$0 = x_0^2 + 2x_0x_1 + 2x_1^2 + 3x_0 + 2x_1 - 46$$

Send to Quantum Annealer

$$H^{\mathrm{QUBO}}$$
$$= \Big[P^{(0)} + P^{(1)} \left(\cdot\mathcal{B} + \circ\mathcal{A} \otimes \mathcal{R} \cdot \psi \right)$$
$$+ P^{(2)} \left(\cdot\mathcal{B} + \circ\mathcal{A} \otimes \mathcal{R} \cdot \psi \right)^2 + \ldots \Big]^2$$

Solution!

$$\psi = (\psi_0 \quad \psi_1 \quad \psi_2 \quad \psi_3$$
$$\psi^a_{01} \quad \psi^a_{02} \quad \psi^a_{03} \quad \psi^a_{12} \quad \psi^a_{13} \quad \psi^a_{23})$$
$$= (0 \quad 1 \quad 1 \quad 1$$
$$0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1)$$
$$\rightarrow (2 \quad 3)$$



Integer Linear Programming

After finishing that first paper, I realized that annealers were inefficient at dealing with real numbers but potentially quite good at dealing with integer numbers.

Classically optimizing on integer numbers in NP-hard.

Problem definition

Solve for	$\text{argmax}(c_i x_i)$	linear
Subject to	$A_{ij} x_j \leq b_i$	
	$x_i \geq 0$	
	$x_i \in \mathcal{Z}^n$	integer

Integer Linear Programming

After finishing that first paper, I realized that annealers were inefficient at dealing with real numbers but potentially quite good at dealing with integer numbers.

Classically optimizing on integer numbers in NP-hard.

Problem definition

$$\begin{aligned} \text{Solve for} \quad & \operatorname{argmax}(c_i x_i) \\ \text{Subject to} \quad & A_{ij} x_j \leq b_i \\ & x_i \geq 0 \\ & x_i \in \mathcal{Z}^n \end{aligned}$$

Tackle problem with slack variable

$$\begin{aligned} \text{Solve for} \quad & \operatorname{argmax}(c_i x_i) \\ \text{Subject to} \quad & A_{ij} x_j + s_i = b_i \\ & s_i \geq 0 \\ & x_i \geq 0 \\ & x_i \in \mathcal{Z}^n \end{aligned}$$

Integer Linear Programming

After finishing that first paper, I realized that annealers were inefficient at dealing with real numbers but potentially quite good at dealing with integer numbers.

Classically optimizing on integer numbers in NP-hard.

Problem definition

$$\begin{aligned} \text{Solve for} \quad & \operatorname{argmax}(c_i x_i) \\ \text{Subject to} \quad & A_{ij} x_j \leq b_i \\ & x_i \geq 0 \\ & x_i \in \mathcal{Z}^n \end{aligned}$$

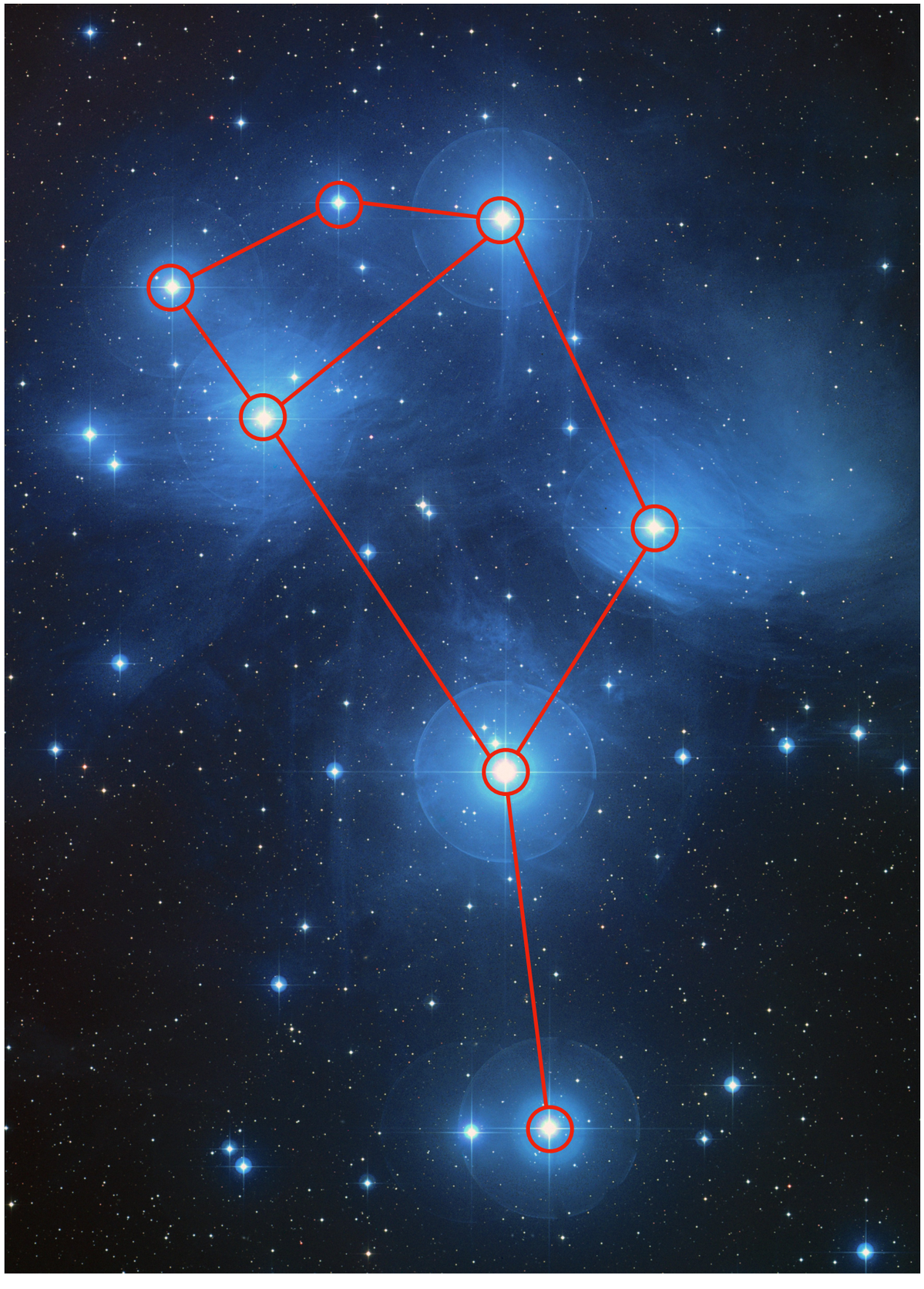
Tackle problem with slack variable

$$\begin{aligned} \text{Solve for} \quad & \operatorname{argmax}(c_i x_i) \\ \text{Subject to} \quad & A_{ij} x_j + s_i = b_i \\ & s_i \geq 0 \\ & x_i \geq 0 \\ & x_i \in \mathcal{Z}^n \end{aligned}$$

The possibility of mapping inequality constraints to annealing allows mapping of more general optimization problems

The Dominating Set Problem

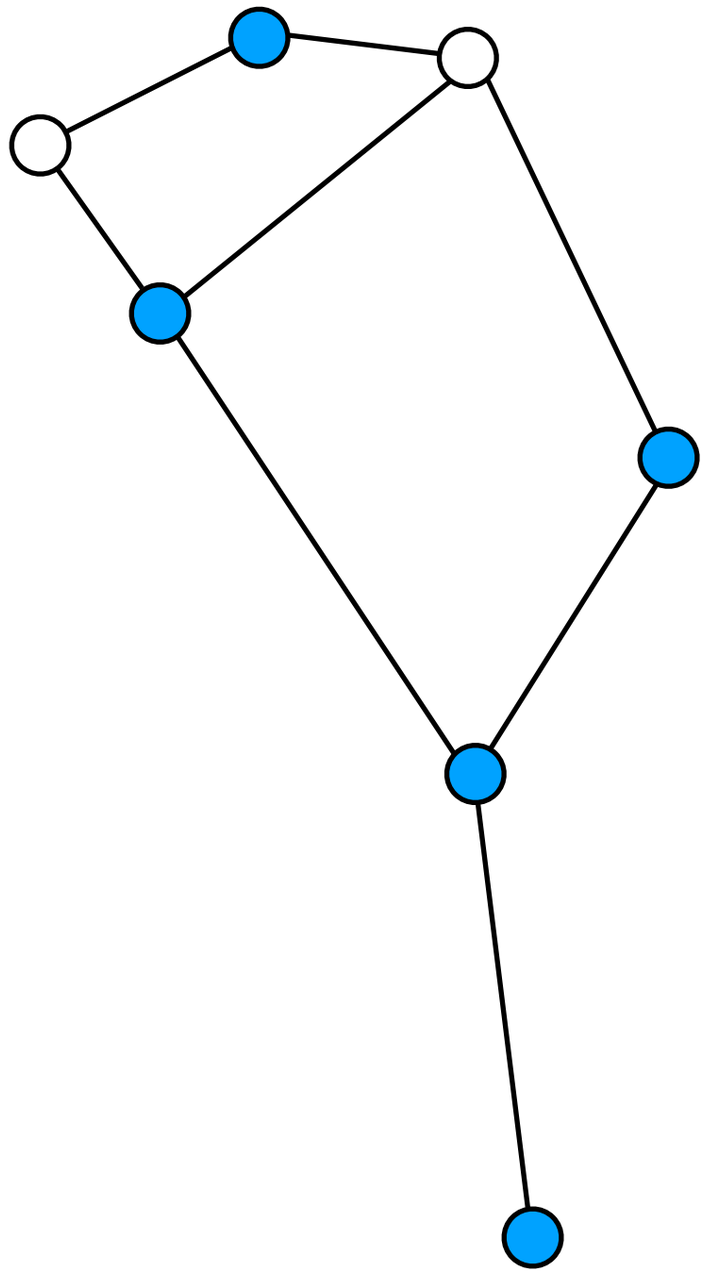
What is a graph?
Defined by vertices (V) and edges (E)
G(V, E)



The Dominating Set Problem

What is a graph?
Defined by vertices (V) and edges (E)
G(V, E)

Dominating Set
A set of vertices {v} in {V} such that
{v} + nearest neighbors = {V}



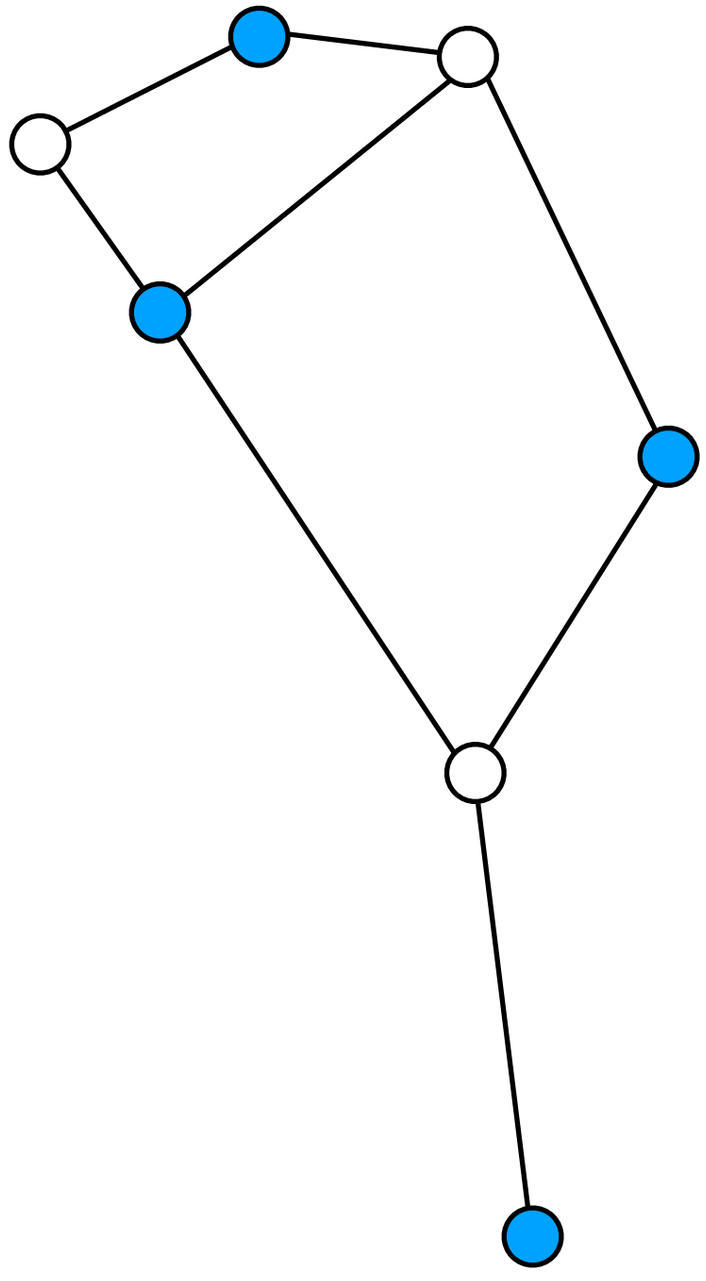
Domination number = 5

The Dominating Set Problem

What is a graph?
Defined by vertices (V) and edges (E)
G(V, E)

Dominating Set
A set of vertices {v} in {V} such that
{v} + nearest neighbors = {V}

Minimal Dominating Set
Set which can not be reduced by removing a vertex



Domination number = 4

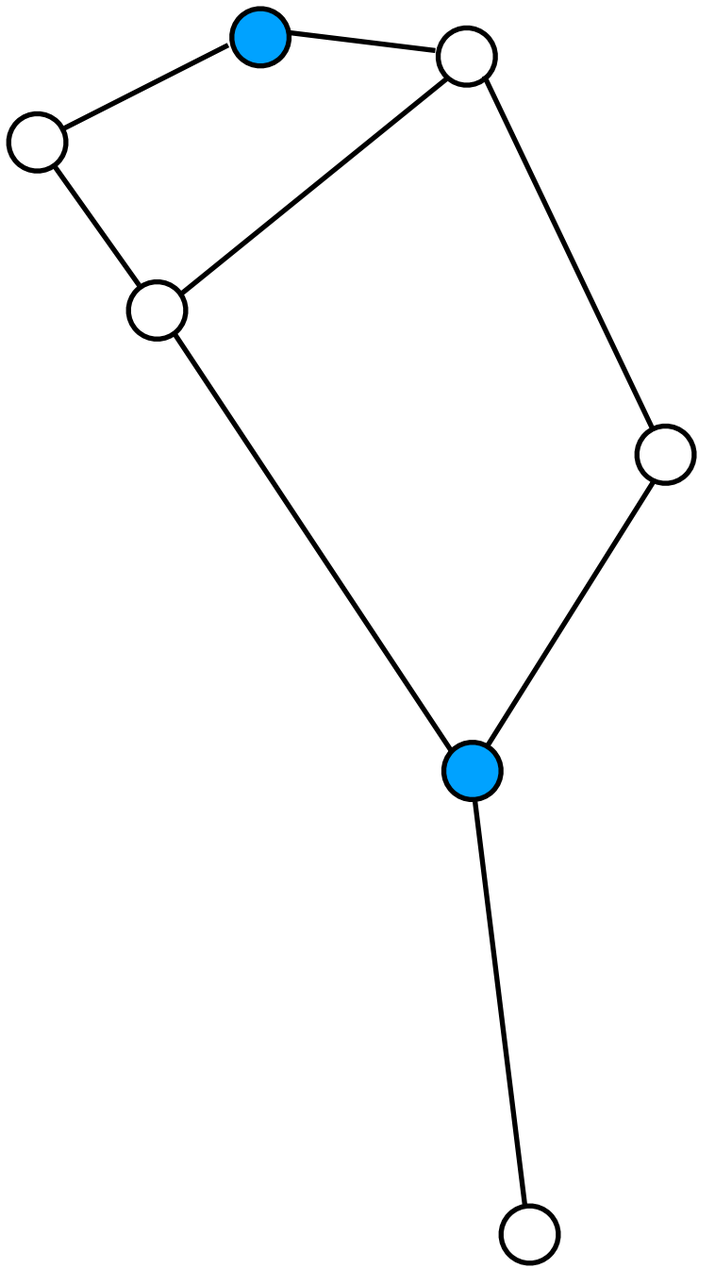
The Dominating Set Problem

What is a graph?
Defined by vertices (V) and edges (E)
G(V, E)

Dominating Set
A set of vertices {v} in {V} such that
{v} + nearest neighbors = {V}

Minimal Dominating Set
Set which can not be reduced by removing a vertex

Minimum Dominating Set
Set with smallest domination number



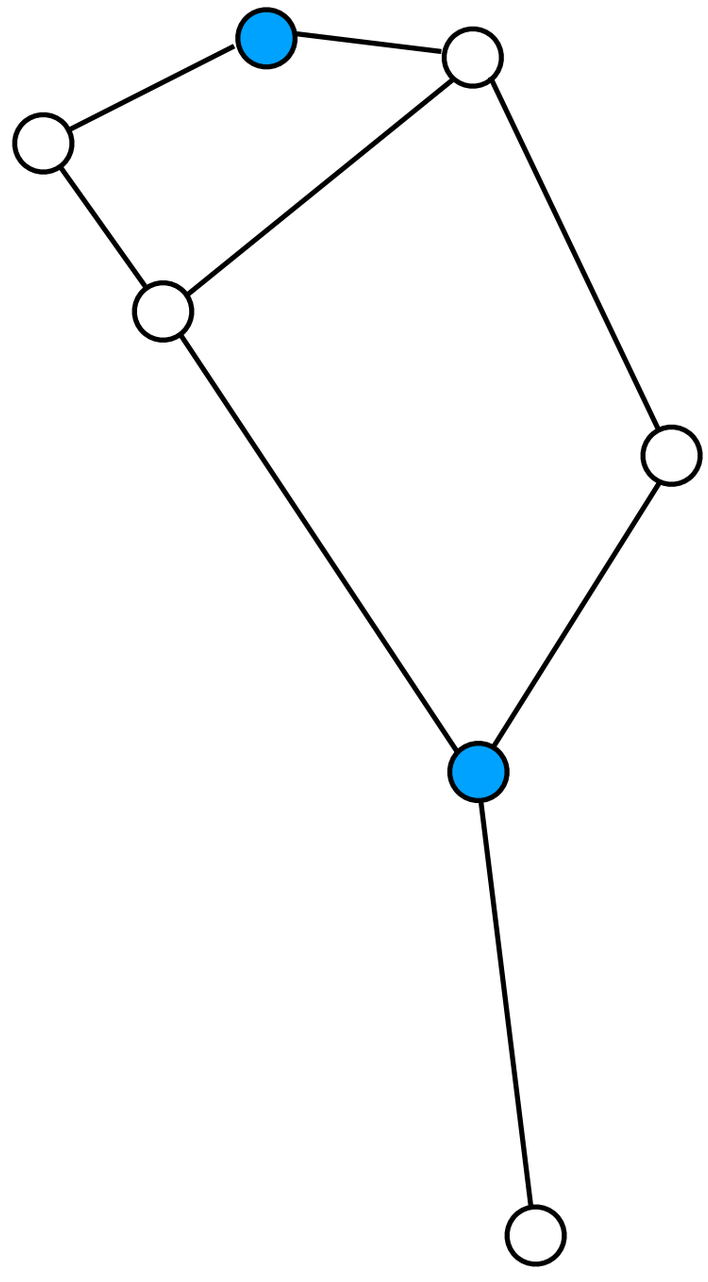
Domination number = 2

Dominating Set as Integer Programming

Problem definition

$$\begin{aligned} \text{Solve for} \quad & \min\left(\sum_{i \in V} x_i\right) \\ \text{Subject to} \quad & x_i + \sum_{j \in \mathcal{N}_i} x_j \geq 1 \\ & x_i \in \{0, 1\} \end{aligned}$$

$\mathcal{N}_i$ = number of nearest neighbors for V_i



Dominating Set as Integer Programming

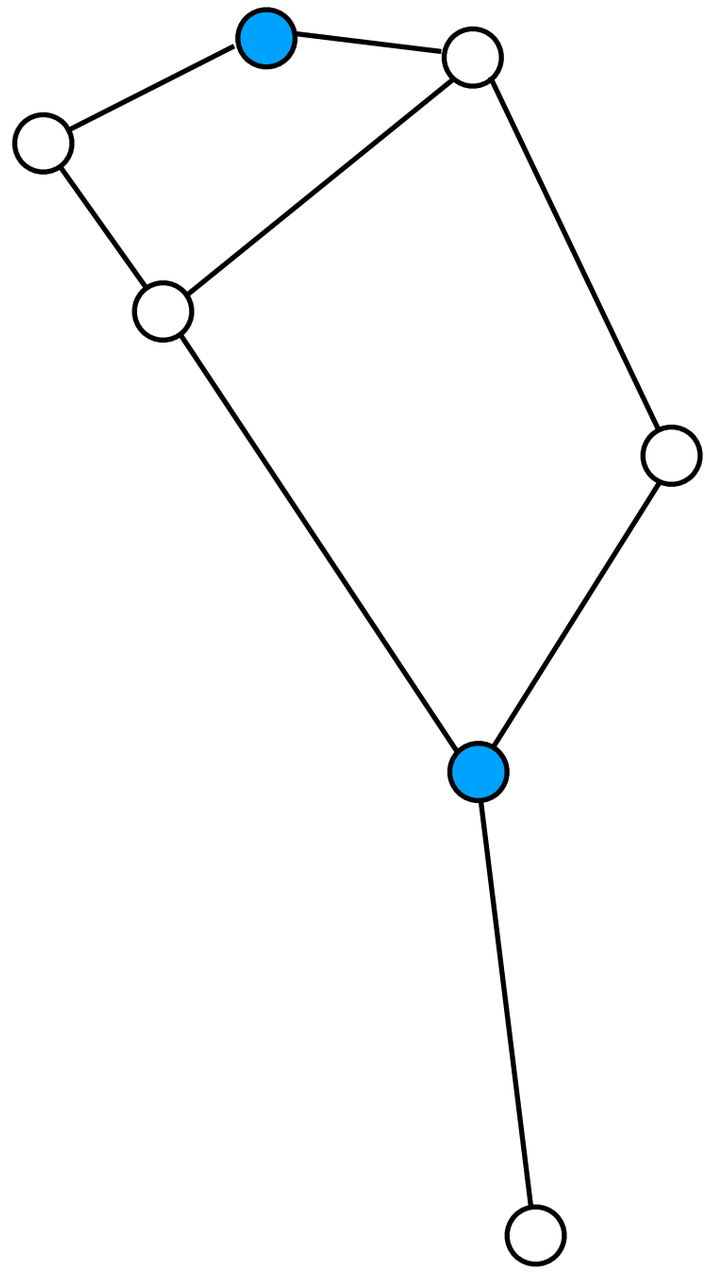
Problem definition

$$\text{Solve for} \quad \min\left(\sum_{i \in V} x_i\right)$$

$$\text{Subject to} \quad x_i + \sum_{j \in \mathcal{N}_i} x_j \geq 1$$

$$x_i \in \{0, 1\}$$

$\mathcal{N}_i = \text{number of nearest neighbors for } V_i$



Map to slack variables

$$\text{Solve for} \quad \min\left(\sum_{i \in V} x_i\right)$$

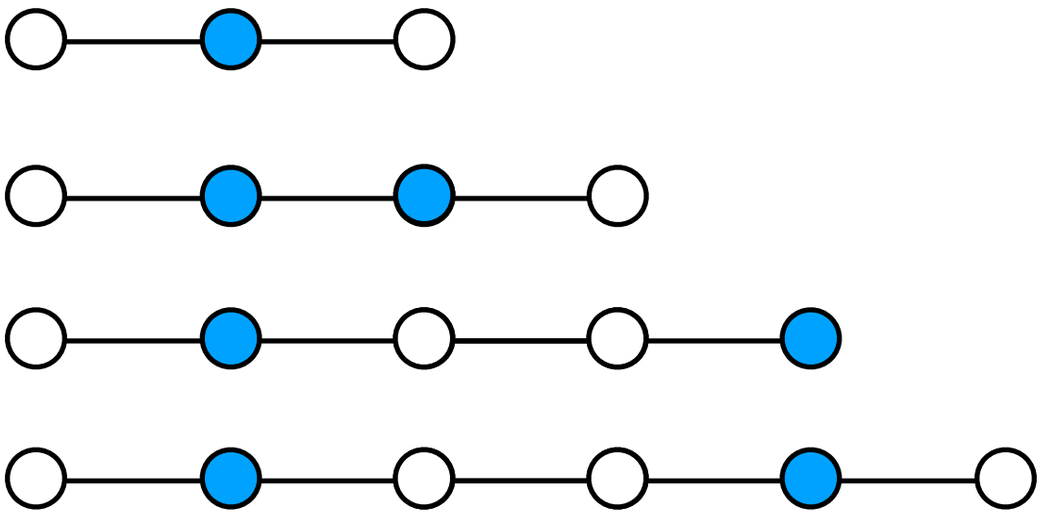
$$\text{Subject to} \quad x_i - s_i + \sum_{j \in \mathcal{N}_i} x_j = 1$$

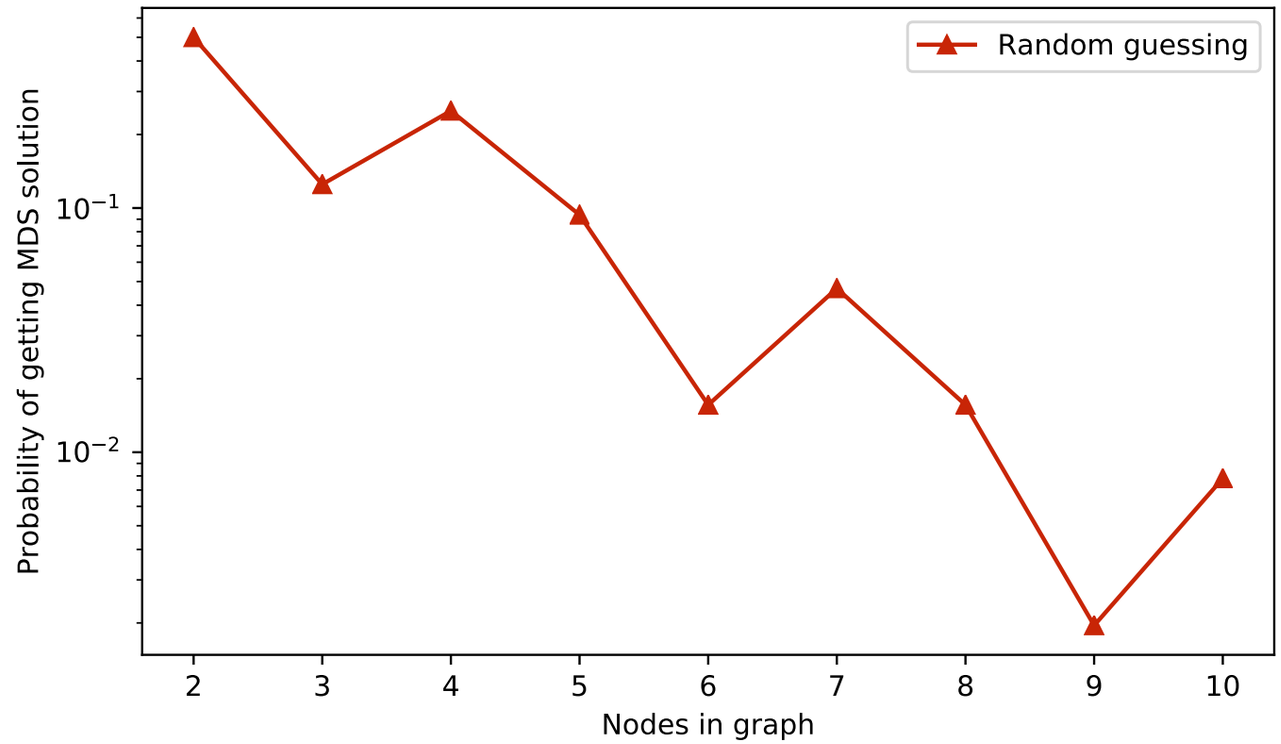
$$0 \leq s_i \leq \mathcal{N}_i$$

$$x_i \in \{0, 1\}$$

Minimum Dominating Set of a 1D Graph

Study the simplest graph possible





For graph G(v):

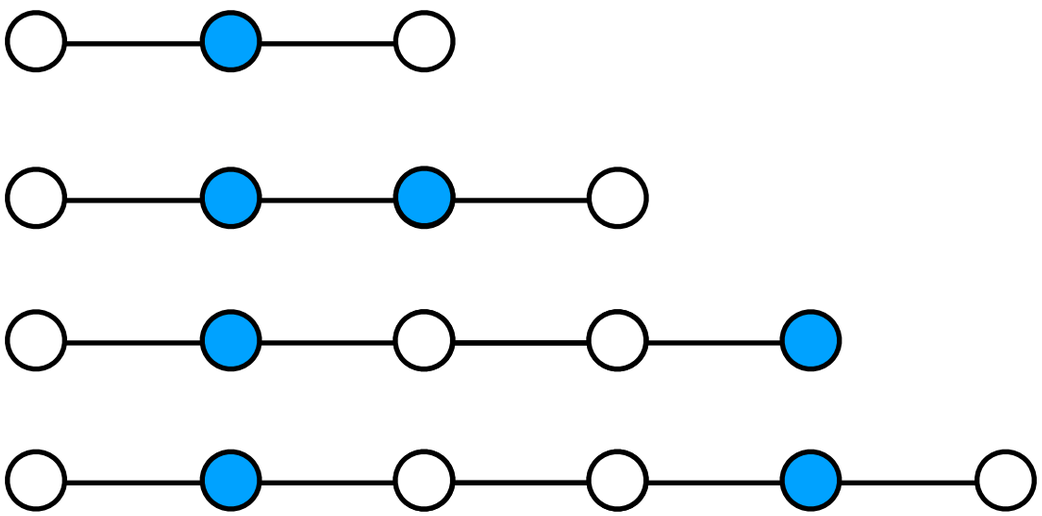
Domination number is ceiling(v/3)

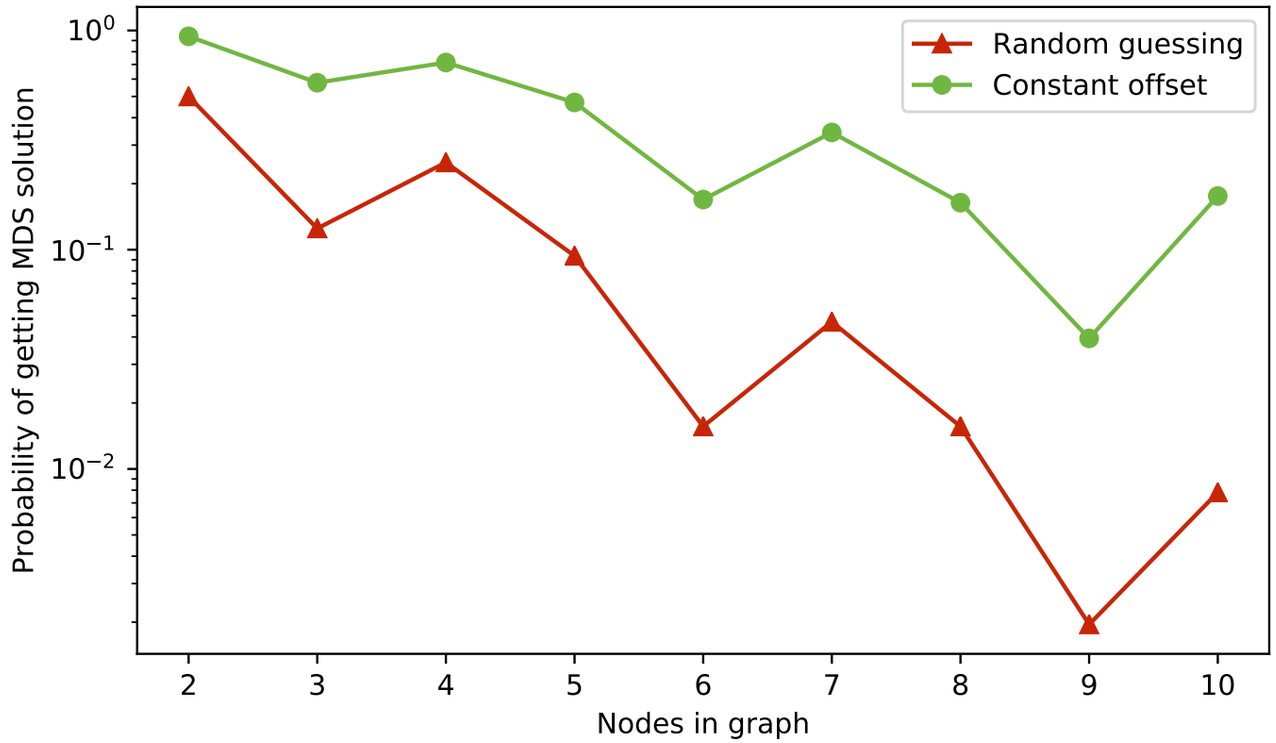
Degeneracy of ground state is
v%3 = 0 is 1
v%3 = 1 is 2*[floor(v/3)+1]
v%3 = 2 is floor(v/3) + 2

For a general graph, the MDS solution can only be obtained by searching every combination.

Minimum Dominating Set of a 1D Graph

Study the simplest graph possible



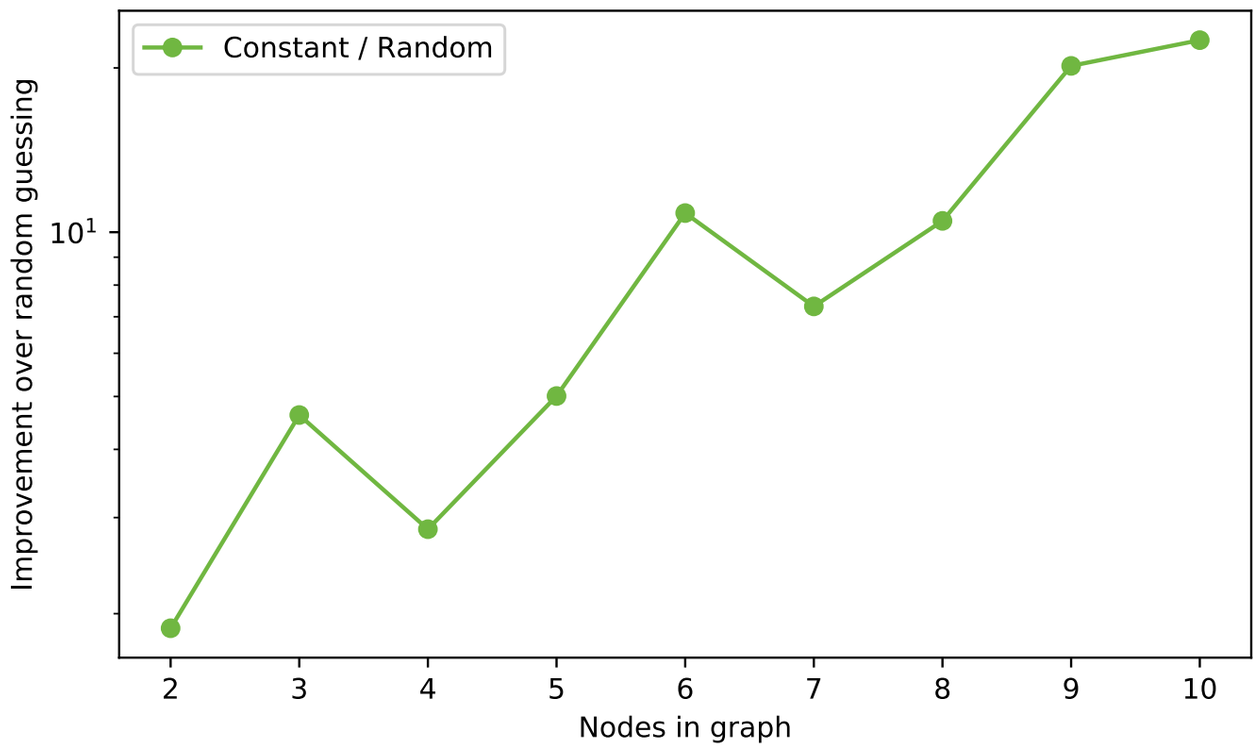


Results from DWave

Kind of like rolling a biased dice.

It is a biased in the correct way for any problem though!

Why…?



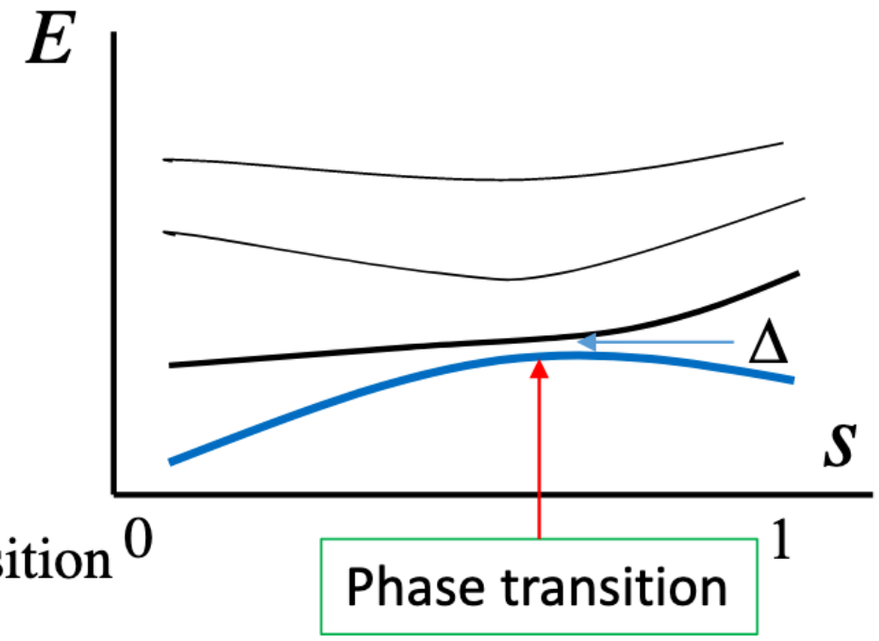
Possible explanations of poor scaling (1)

Computational complexity
From the perspective of phase transitions

Adiabatic condition $\tau \propto \Delta^{-2}$

$$H = sH_0 - (1-s)\sum_{i=1}^{N}\sigma_i^x \quad (s: 0 \to 1) \qquad s = \frac{t}{\tau}$$

Adiabatic computation, Farhi et al (2001)



Gap scaling

$$\Delta \propto \begin{cases} e^{-aN} & 1^{\text{st}} \text{ order transition} \\ N^{-b} & 2^{\text{nd}} \text{ order transition} \end{cases}$$

Complexity

$$\tau \propto \begin{cases} e^{2aN} & \text{(hard)} \\ N^{2b} & \text{(easy)} \end{cases}$$

Very important to avoid 1st order transition

Analysis by equilibrium statistical mechanics (in the $N \to \infty$ limit) is useful to infer the system properties for large but finite N.

(These slides are taken directly from Nishimori's RB19 talk)

Anderson Localization (2)

Anderson Hamiltonian (1958)

How do materials undergo phase transition from conductor to insulator?
The wavefunction is localized in space!

Anderson Localization (2)

Anderson Hamiltonian (1958)

How do materials undergo phase transition from conductor to insulator?
The wavefunction is localized in space!

$$\hat{H} = W \sum_n \hat{\epsilon_n} \hat{c}_n^\dagger \hat{c}_n + t \sum_{\langle n,m \rangle} \left(\hat{c}_n^\dagger \hat{c}_m + \mathrm{h.c} \right)$$

single fermion Hamiltonian

Jordan-Wiger transformation

$$\hat{H} = W \sum_n \hat{\epsilon_n} \sigma_n^z + t \sum_{\langle n,m \rangle} \sigma_n^x \sigma_m^y \cdots$$

very similar to spin-glass

W is the strength of disorder
epsilon is a random binomial draw

Anderson Localization (2)

Anderson Hamiltonian (1958)

How do materials undergo phase transition from conductor to insulator?
The wavefunction is localized in space!

$$\hat{H} = W \sum_n \hat{\epsilon_n} \hat{c}_n^\dagger \hat{c}_n + t \sum_{\langle n,m \rangle} \left(\hat{c}_n^\dagger \hat{c}_m + \mathrm{h.c} \right)$$

single fermion Hamiltonian

↓

Jordan-Wiger transformation

$$\hat{H} = W \sum_n \hat{\epsilon_n} \sigma_n^z + t \sum_{\langle n,m \rangle} \sigma_n^x \sigma_m^y \cdots$$

very similar to spin-glass

Anderson showed that the wavefunction of this type of Hamiltonian is localized (exponentially decays) in space
—> quantum analogy of non-ergodicity

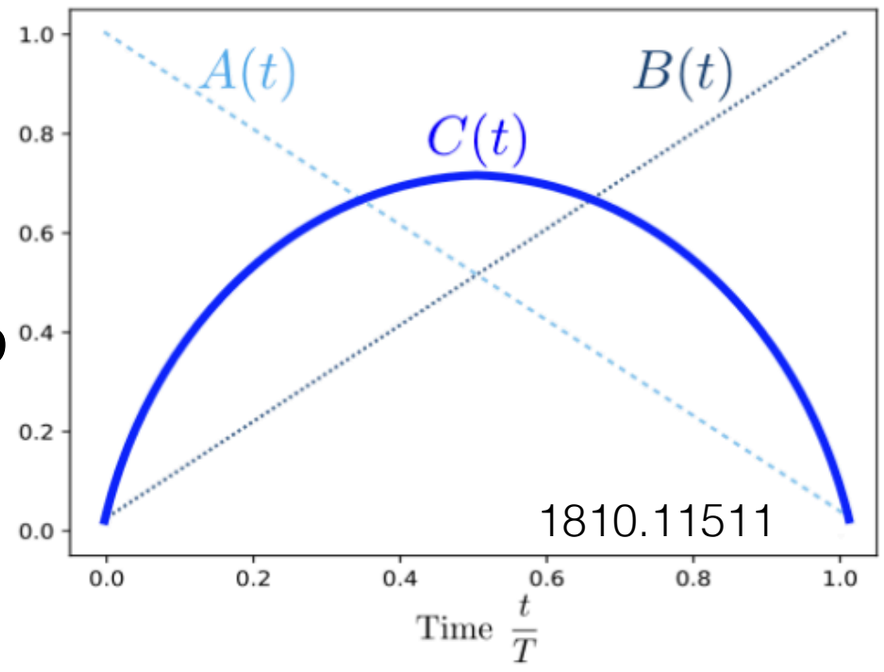
Some improvements to annealing schedule

Non-stoquastic Hamiltonians
Stoquastic = Stochastic + quantum (Current annealers are this)
Non-stoquastic is basically a universal annealer (but it is not advertised this way)
Idea: Extra phase-space may allow larger gap to solve classical problems.

Inhomogeneous driving fields
Instead of evolving Hamiltonian uniformly, try locally evolving Hamiltonian.
Idea: Phase-transition is group behavior or spins are frozen due to Anderson localization.

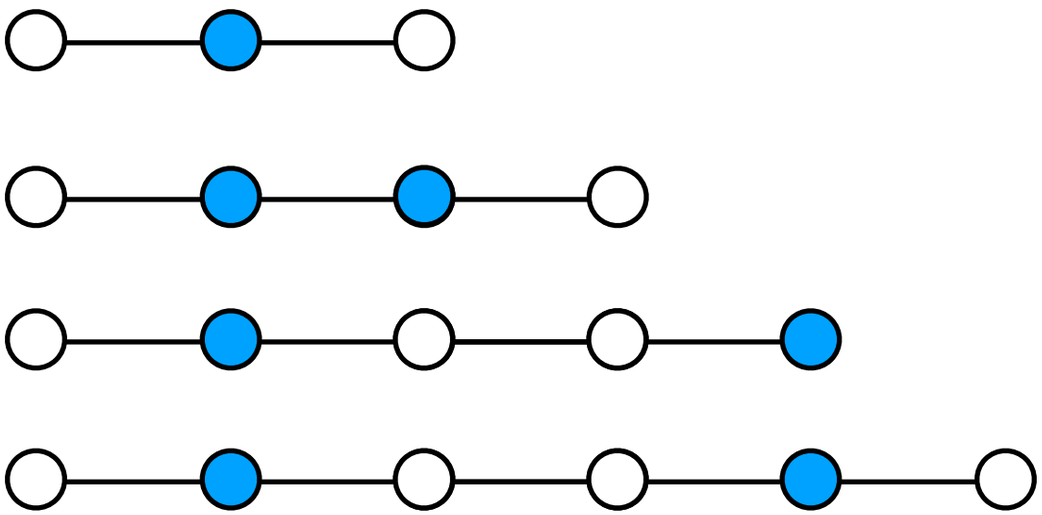
Reverse annealing
Analogous to classically reheating then cooling down the system.
Idea: Anneal much faster than adiabatic limit, but do so repeatedly to hopefully converge to the ground-state.

Navigator Hamiltonians [Vanqver algorithm from 1QBit]
$H(s) = A(s)H^{\text{init}} + B(s)H^{\text{fin}} + C(s)H^{\text{nav.}}(\theta)$
Idea: Parameterize all 3 ideas, and optimize navigator to speed up the annealing process.



Result of inhomogeneous driving field

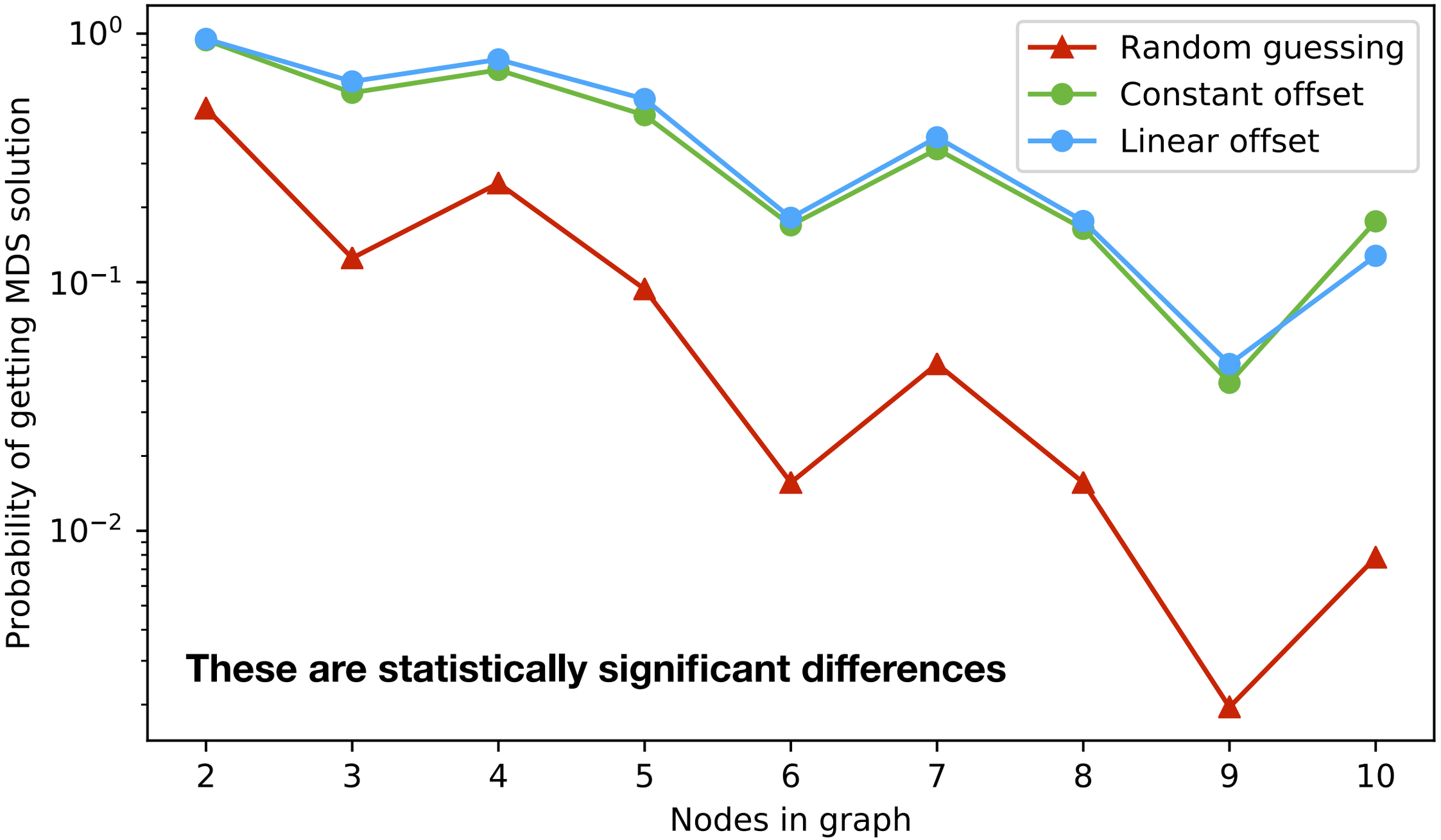
Study the simplest graph possible



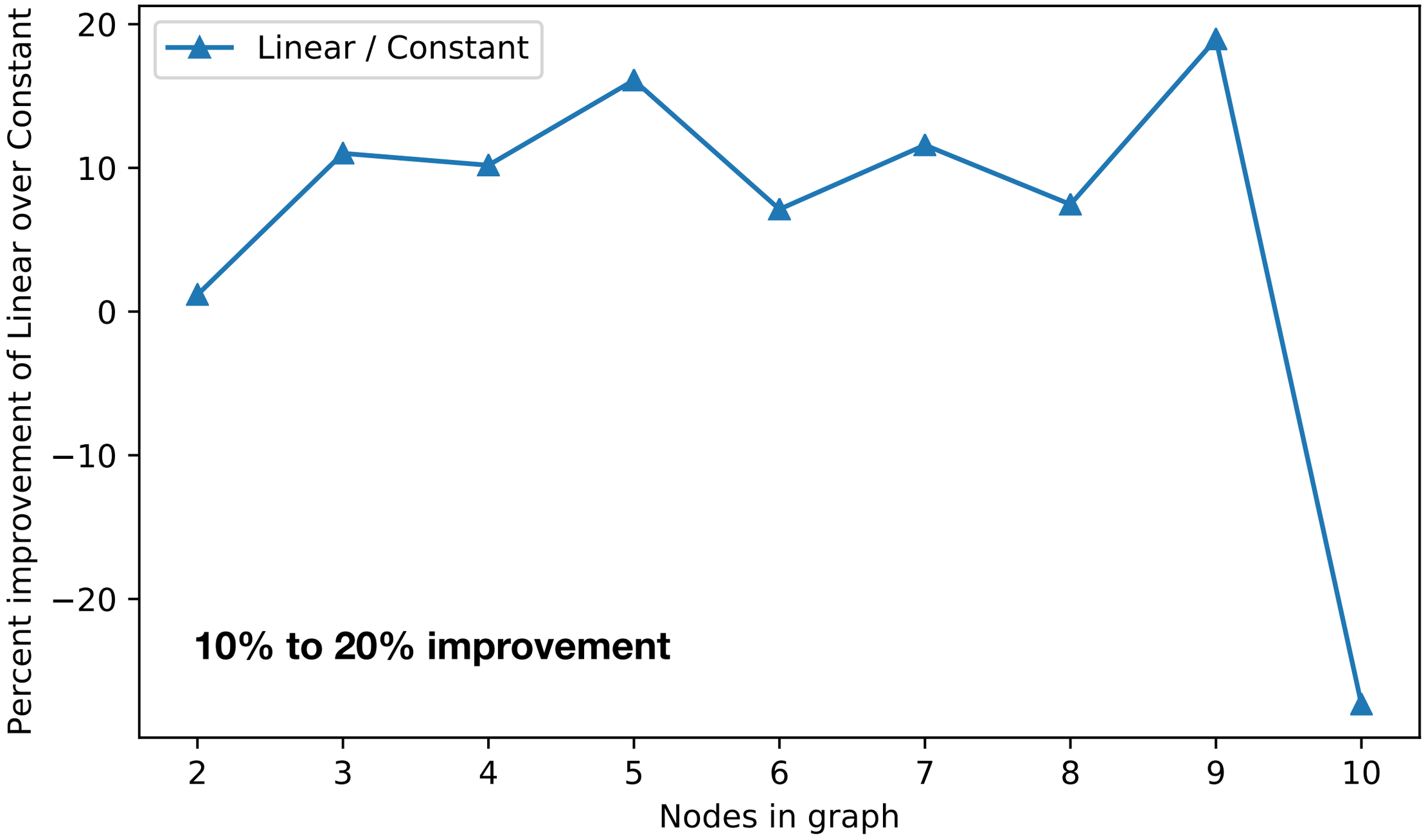
I will show DWave results where we linearly delay qubits with larger magnitude of external magnetic field

An improvement will suggest maybe **we are seeing physics**!

Result of inhomogeneous driving field

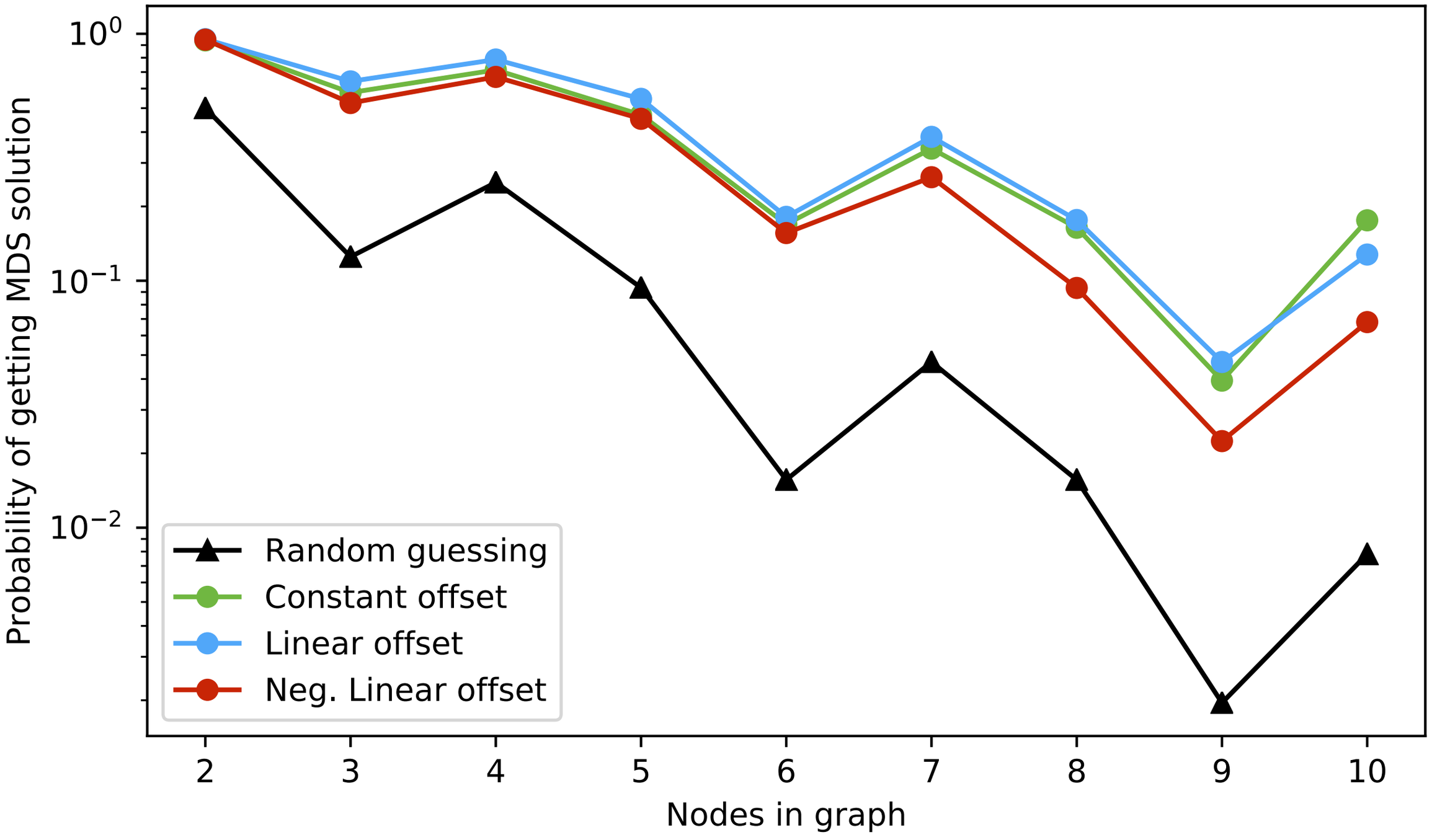


Result of inhomogeneous driving field



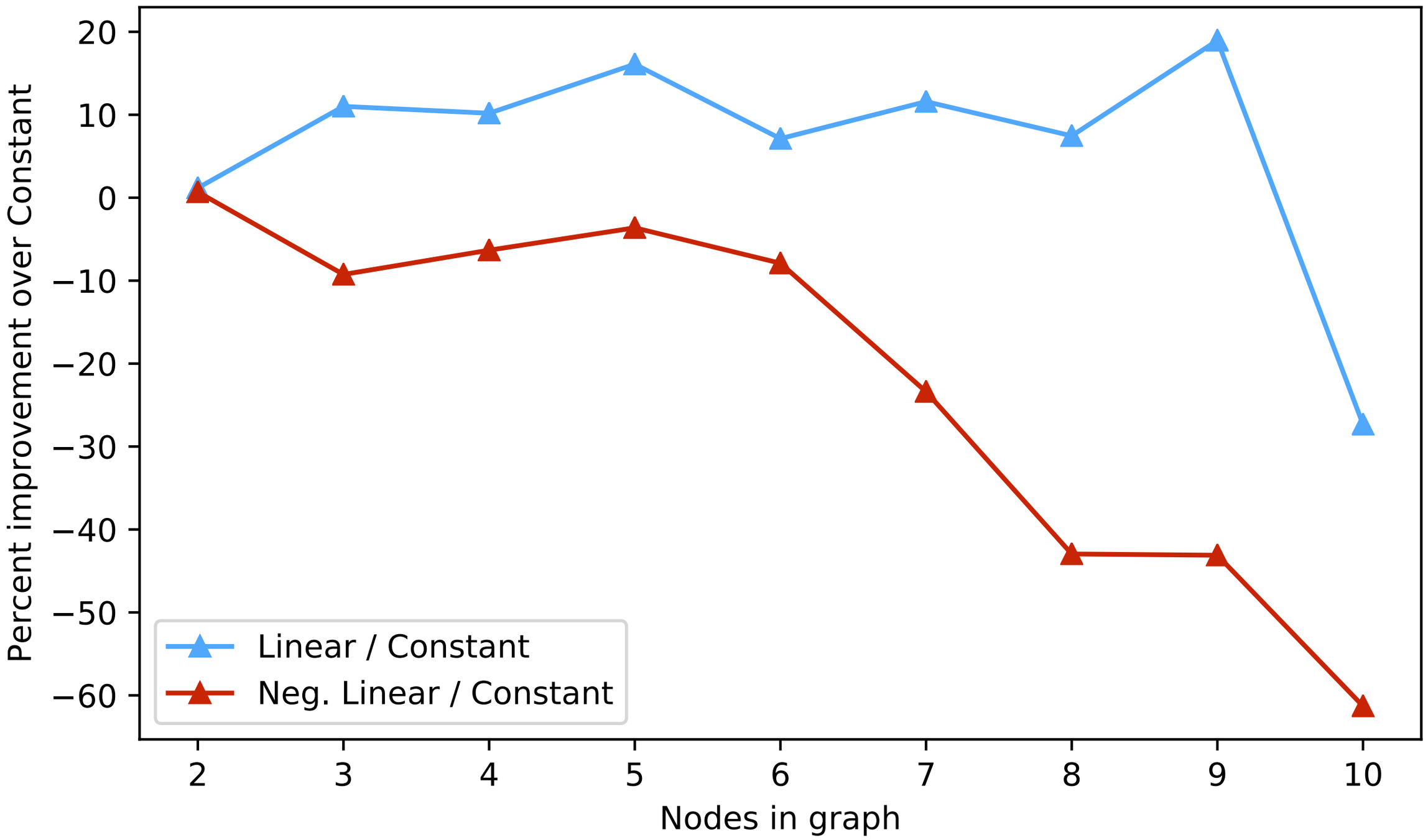
What if we do a negative linear offset?

That is, we delay the anneal opposite of what Anderson Localization suggests



These are statistically significant differences

What if we do a negative linear offset?



Solving a programmable spin-glass model

$$H^{\text{init}} = \sum_i \sigma_i^x \qquad H^{\text{fin}} = \sum_{ij} J_{ij}\sigma_i^z\sigma_j^z + \sum_i h_i\sigma_i^z$$

The purpose is to solve the final Hamiltonian
One can map ALL classical NP-hard optimization problems to this model
(this is why it is interesting)

If we drop the tran

- if J is a constant
  - beyond 1-din ions
- If J is random, we get the **spin-glass model**

Can we solve this with classical methods?

- **If J and h is programmable, we have a computer**

These systems are solvable with **thermal** (classical) phase transitions
OR
at zero temperature **quantum** phase transitions

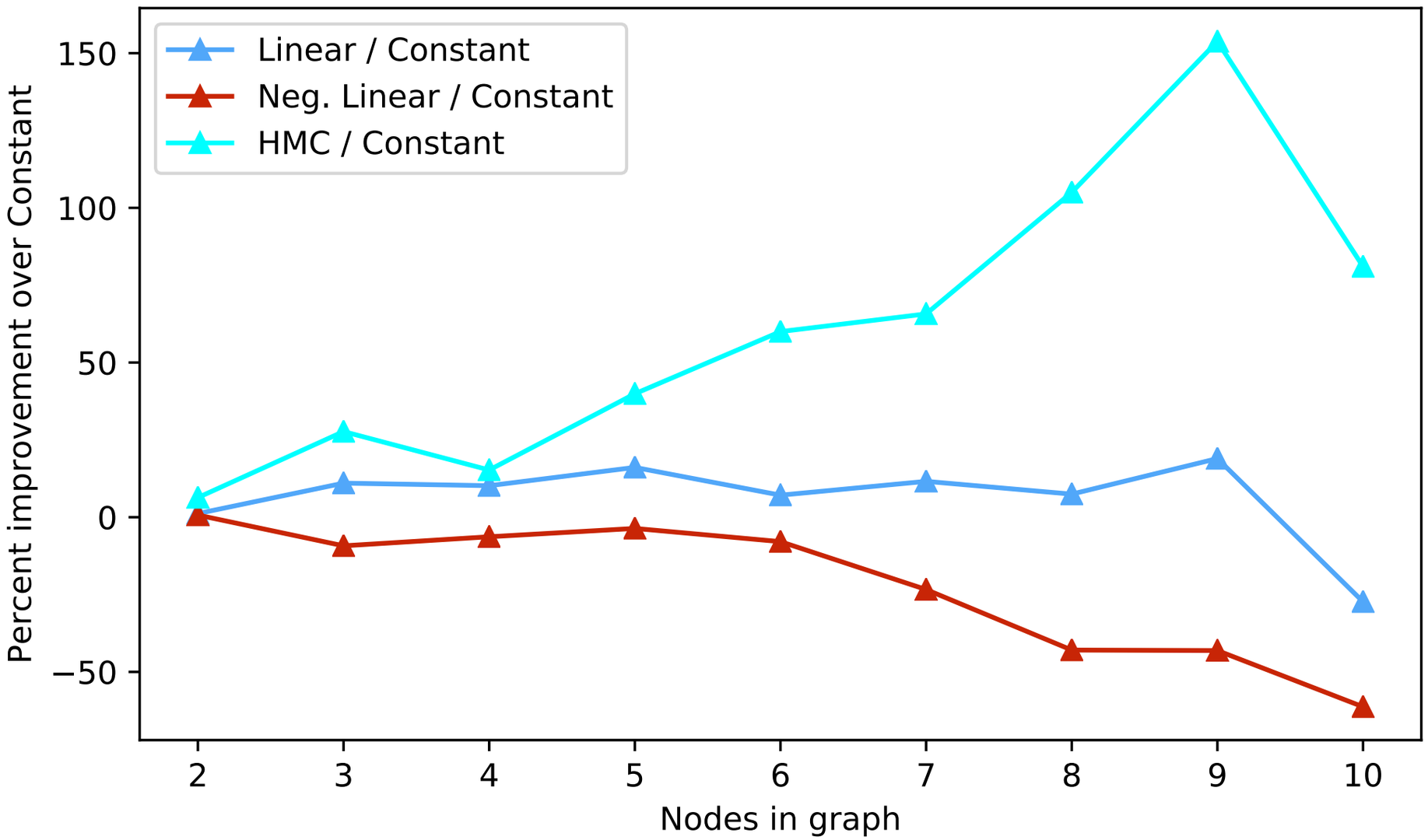
This is why quantum annealing is (and is not) a quantum computer

Hybrid Monte Carlo

This is on-going work with Tom Luu *et al.*

In Lattice QCD we solve very large spin systems… maybe we can repurpose this technology to solve the spin-glass model?

The details are for another talk….

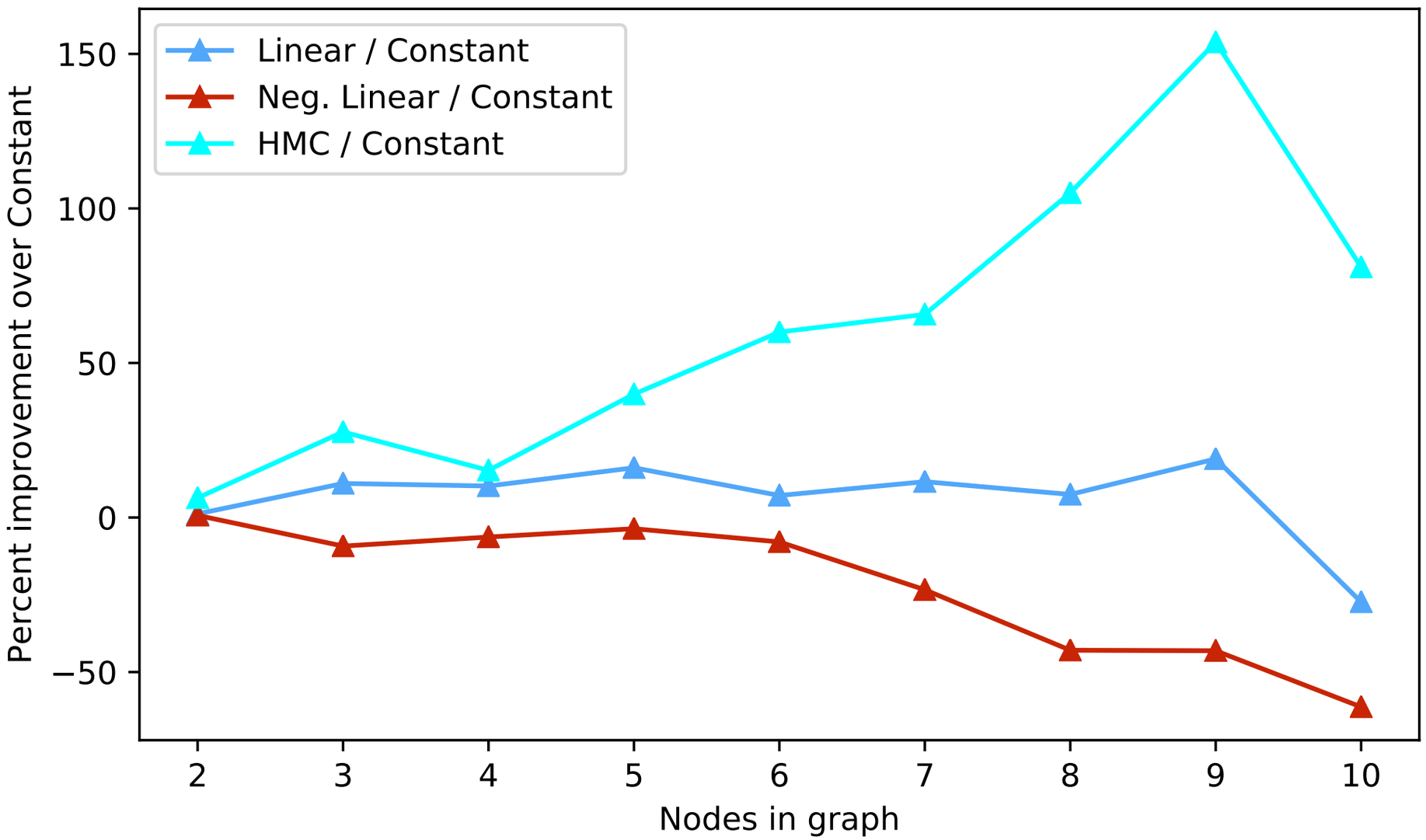


Hybrid Monte Carlo

We have code

It works

Next: dynamically change the external field for HMC to see if we can improve this.



Key take aways

We can solve all classical NP-hard optimization if we can efficiently solve the spin-glass ground state.

We map a computer science problem to a physics problem.

We can hope to use understanding of physics to solve computational problems!
(physics help quantum computation)

We can hope to gain new insight from QC to help classical computing!
(quantum computing inspire progress in classical computing)

We can hope to use better classical computing to help physics now!
(the cycle of life and death continues…)

Key take aways

We can solve all classical NP-hard optimization if we can efficiently solve the spin-glass ground state.

We map a computer science problem to a physics problem.

We can hope to use understanding of physics to solve computational problems!
(physics help quantum computation)

We can hope to gain new insight from QC to help classical computing!
(quantum computing inspire progress in classical computing)

We can hope to use better classical computing to help physics now!
(the cycle of life and death continues…)

I think this is very cool…

Collaborators

Arjun Gambhir (LQCD guy)
Travis Humble (Quantum Computing (ORNL))
Shigetoshi Sota (Condensed Matter (RIKEN))

Chris Koerber (Nuclear)
Chih-Chieh Chen (Condensed Matter (industry))
Jim Ostrowski (Industrial & Systems Engineer (UTennessee))

Thomas Luu (LQCD (Julich))
Johann Ostmeyer (Tom’s postdoc)

Roger Ruo-Qian Wang (Civil and Env. Engineer (Rutgers))

Naturally leads to a broad set of collaborators!