Introduction to Quantum Computing

Tobias Stollenwerk

Simulation and Software Technology
High Performance Computing Department

Seminar Materials Physics in Space

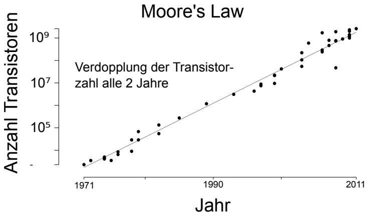
DLR

Content

- Introduction to Quantum Computers
- Near-Term Quantum Algorithms
- Quantum Annealer (D-Wave)

The End of Moore's Law?

- Quantum effects become relevant with increasing transistor density



- 'Beyond exascale' systems . . . will be based on new technologies . . .
  Paul Messina, Argonne National Laboratory

- Quantum technologies could be a solution

Quantum Bits

- Classical bit is either "0" **or** "1"

0 1

Quantum Bits

- Classical bit is either "0" **or** "1"

0 1

- Quantum state is superposition of "0" **and** "1"

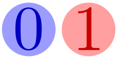
$$|\psi\rangle = a|0\rangle + b|1\rangle$$

0 1

where $|a|^2 + |b|^2 = 1$ and $a, b \in \mathbb{C}$

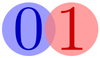
Quantum Bits

- Classical bit is either "0" **or** "1"



- Quantum state is superposition of "0" **and** "1"

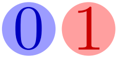
$$|\psi\rangle = a|0\rangle + b|1\rangle$$



where $|a|^2 + |b|^2 = 1$ and $a, b \in \mathbb{C}$

Quantum Bits

- Classical bit is either "0" **or** "1"



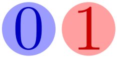
- Quantum state is superposition of "0" **and** "1"

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

where $|a|^2 + |b|^2 = 1$ and $a, b \in \mathbb{C}$

Quantum Bits

- Classical bit is either "0" **or** "1"



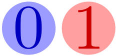
- Quantum state is superposition of "0" **and** "1"

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

where $|a|^2 + |b|^2 = 1$ and $a, b \in \mathbb{C}$

Quantum Bits

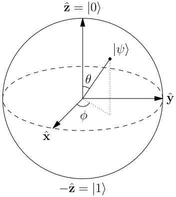
- Classical bit is either "0" **or** "1"



- Quantum state is superposition of "0" **and** "1"

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

where $|a|^2 + |b|^2 = 1$ and $a, b \in \mathbb{C}$



$$\Rightarrow |\psi\rangle = sin\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi} cos\left(\frac{\theta}{2}\right)|1\rangle$$

Quantum Bits

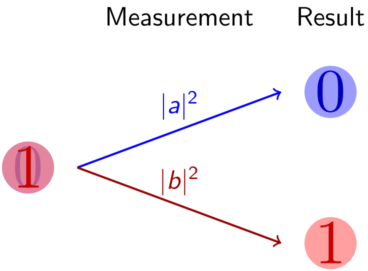
- Measure the state

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

Quantum Bits

- Measure the state

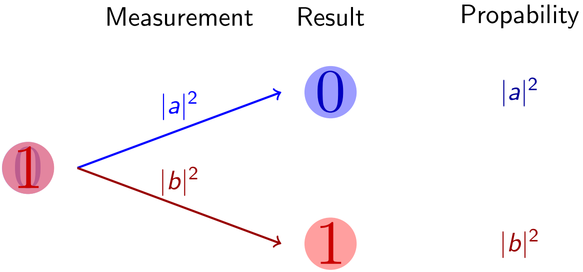
$$|\psi\rangle = a|0\rangle + b|1\rangle$$



Quantum Bits

- Measure the state

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

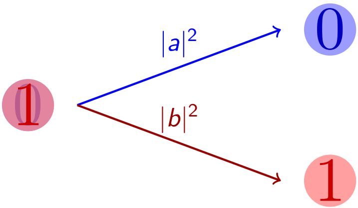


Quantum Bits

- Measure the state

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

Measurement	Result	Propability	Final state
	0	$\lvert a\rvert^2$	$\lvert\psi\rangle = \lvert 0\rangle$
	1	$\lvert b\rvert^2$	$\lvert\psi\rangle = \lvert 1\rangle$



Quantum Bits

- Measure the state

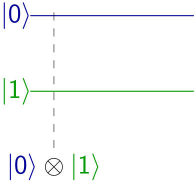
$$|\psi\rangle = a|0\rangle + b|1\rangle$$

Measurement	Result	Propability	Final state
$\lvert a\rvert^2$	0	$\lvert a\rvert^2$	$\lvert\psi\rangle = \lvert 0\rangle$
$\lvert b\rvert^2$	1	$\lvert b\rvert^2$	$\lvert\psi\rangle = \lvert 1\rangle$

- Measurement changes the state

Universal Quantum Computer

Gate model: Manipulate quantum states through quantum gates



Universal Quantum Computer

Gate model: Manipulate quantum states through quantum gates

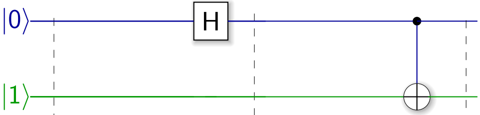
$|0\rangle \otimes |1\rangle$

$$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \otimes |1\rangle$$

$$= \frac{1}{\sqrt{2}}\left(|0\rangle|1\rangle + |1\rangle|1\rangle\right)$$

Universal Quantum Computer

Gate model: Manipulate quantum states through quantum gates



$|0\rangle \otimes |1\rangle$

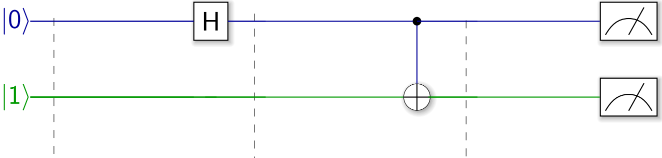
$$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \otimes |1\rangle$$

$$= \frac{1}{\sqrt{2}}\left(|0\rangle|1\rangle + |1\rangle|1\rangle\right)$$

$$\frac{1}{\sqrt{2}}\left(|0\rangle|1\rangle + |1\rangle|0\rangle\right)$$

Universal Quantum Computer

Gate model: Manipulate quantum states through quantum gates



$$|0\rangle \otimes |1\rangle$$

$$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \otimes |1\rangle$$

$$= \frac{1}{\sqrt{2}}\left(|0\rangle|1\rangle + |1\rangle|1\rangle\right)$$

$$\frac{1}{\sqrt{2}}\left(|0\rangle|1\rangle + |1\rangle|0\rangle\right)$$

DLR

Quantum Algorithms

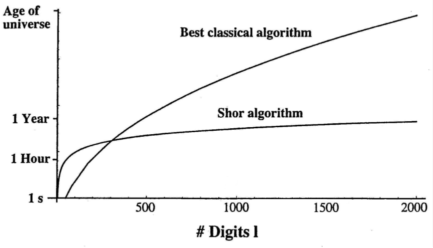
Algorithm	Runtime classical	Runtime quantum	Application
Deutsch-Josza	$2^n/2$	1	Academical
Grover's search algorithm	n	$\sqrt{n}$	Database
Shor's Factorization Algorithm	Exponential	Polynomial	Cryptography

Quantum Algorithms

Algorithm	Runtime classical	Runtime quantum	Application
Deutsch-Josza	$2^n/2$	1	Academical
Grover's search algorithm	n	$\sqrt{n}$	Database
Shor's Factorization Algorithm	Exponential	Polynomial	Cryptography

Quantum Algorithms

Algorithm	Runtime classical	Runtime quantum	Application
Deutsch-Josza	$2^n/2$	1	Academical
Grover's search algorithm	n	$\sqrt{n}$	Database
Shor's Factorization Algorithm	Exponential	Polynomial	Cryptography



HHL Algorithm for Radar Cross Section

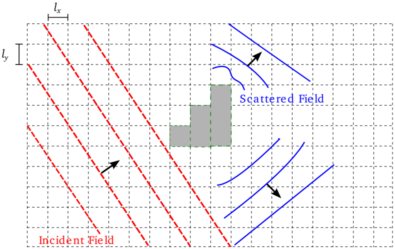
HHL Algorithm

- Harrow, Hassidim, Lloyd (2008, arXiv:0811.3171)
- Solves $A\mathbf{x} = \mathbf{b}$ in $\mathcal{O}(\log n)$ instead of $\mathcal{O}(n^2)$

Fine Print

- A must be sparse
- A must be well conditioned
- Solution $\mathbf{x}$ is encoded in state $|x\rangle = \sum_i x_i |i\rangle$
- Needs quantum error correction

HHL Algorithm for Radar Cross Section



Clader et.al. arXiv:1301.2340

- FEM calculation for stationary scattering problem
- Scattering cross section is of the form

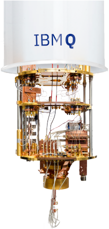
$$S \sim |\mathbf{R} \cdot \mathbf{x}|^2 = |\langle \mathbf{R} | \mathbf{x} \rangle|^2$$

Research Questions

- How would real world problems perform
- How do increasingly complex scattering geometries influence
  - The condition number
  - The number of gates

Near-Term Quantum Computers

- Recent hardware devolopment up to 72 qubits
- Hardware restrictions (fidelity, noise, feasible gates, etc.)
- "Noisy Intermediate-Scale Quantum Computers"
- Compare to early supercomputers. How to employ their power for something useful?



IBM Q

Google Bristlecone Chip

Rigetti Chip Architecture

DLR

Near-Term Quantum Algorithms

Question

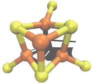
What can we do with “Noisy Intermediate-Scale Quantum Computers“ in the near future?

Answer

- Use Algorithms with no proven speedup
- Use Algorithms which do not require quantum error correction
- Note: Most of the current codes run on a supercomputer have not theoretically proven speedup

Quantum Chemistry - Variational Quantum Eigensolver

- Calculate ground state of molecules



fermionic problem ➡

classical cost function ➡

qubit Hamiltonian

$$H_q = \sum_\alpha h_\alpha P_\alpha = \sum_\alpha h_\alpha \bigotimes_{j=1}^{N} \sigma_j^{\alpha_j}.$$

classical

calculate energy

$$E = \sum_\alpha h_\alpha \langle \Psi(\theta) | P_\alpha | \Psi(\theta) \rangle \geq E_{\text{exact}}$$

adjust parameters

θ

optimize

quantum

prepare trial state

$|\Psi(\theta)\rangle$

measure expectation values

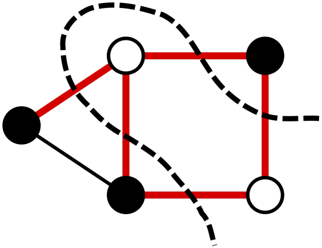
$$\langle \Psi(\theta) | \bigotimes_{j=1}^{N} \sigma_j^{\alpha_j} | \Psi(\theta) \rangle$$

solution θ^*

;

Quantum Approximate Optimization Algorithm

- Approximate solution of combinatorial optimization problems
- For MAXCUT there was a speed-up until better classical algorithm was developed

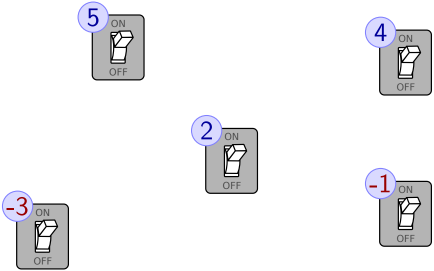


;

Quantum Annealer

- Optimizer for Ising problems

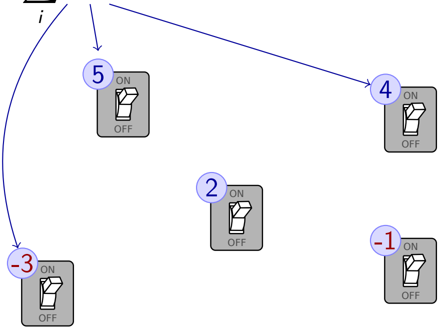
$$E = \sum_i H_i \, s_i \qquad \text{with } x_i \in \{-1, 1\}$$



Quantum Annealer

- Optimizer for Ising problems

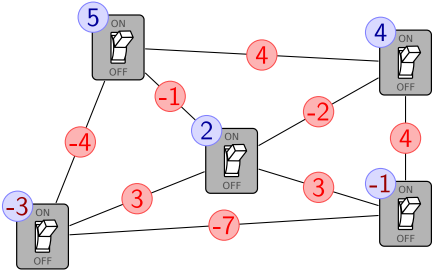
$$E = \sum_i H_i \; s_i \qquad \text{with } x_i \in \{-1, 1\}$$



Quantum Annealer

- Optimizer for Ising problems

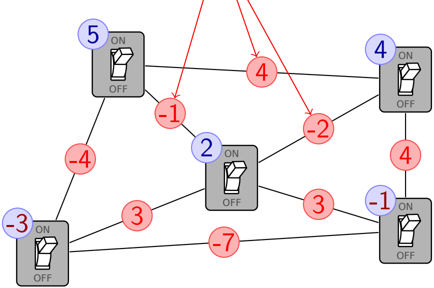
$$E = \sum_i H_i \, s_i \qquad \text{with } x_i \in \{-1, 1\}$$



Quantum Annealer

- Optimizer for Ising problems

$$E = \sum_i H_i \, s_i + \sum_{i \neq j} J_{ij} \, s_i s_j \qquad \text{with } x_i \in \{-1, 1\}$$

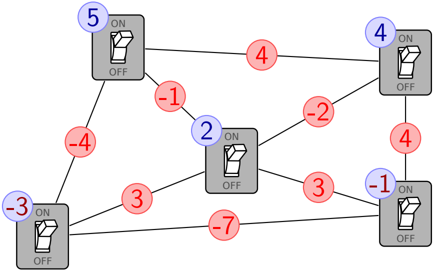


DLR

Quantum Annealer

- Optimizer for Ising problems

$$E = \sum_{i} H_i \; s_i + \sum_{i \neq j} J_{ij} \; s_i s_j \qquad \text{with } x_i \in \{-1, 1\}$$



Quantum Annealing - How does it work?

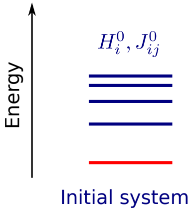
- How do we bring the system in to the final state?
- Solution: Adiabatic evolution
  1. Prepare simple initial system with know ground state
  2. Change system *slowly* towards the final system

Quantum Annealing - How does it work?

- How do we bring the system in to the final state?
- Solution: Adiabatic evolution
  1. Prepare simple initial system with know ground state
  2. Change system *slowly* towards the final system

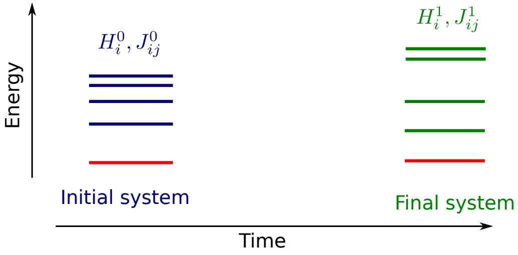
Quantum Annealing - How does it work?

- How do we bring the system in to the final state?
- Solution: Adiabatic evolution
  1. Prepare simple initial system with know ground state
  2. Change system *slowly* towards the final system



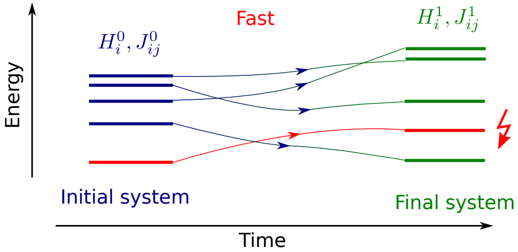
Quantum Annealing - How does it work?

- How do we bring the system in to the final state?
- Solution: Adiabatic evolution
    1. Prepare simple initial system with know ground state
    2. Change system *slowly* towards the final system



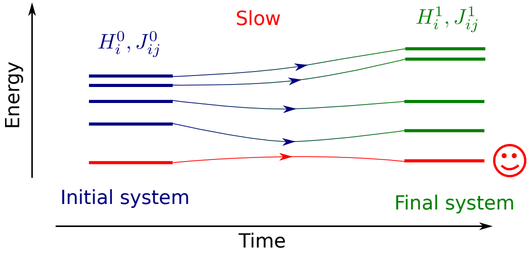
Quantum Annealing - How does it work?

- How do we bring the system in to the final state?
- Solution: Adiabatic evolution
  1. Prepare simple initial system with know ground state
  2. Change system *slowly* towards the final system



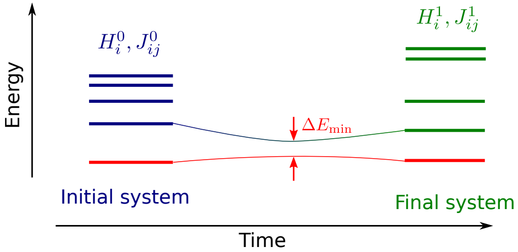
Quantum Annealing - How does it work?

- How do we bring the system in to the final state?
- Solution: Adiabatic evolution
  1. Prepare simple initial system with know ground state
  2. Change system *slowly* towards the final system



Quantum Annealing - How does it work?

- How do we bring the system in to the final state?
- Solution: Adiabatic evolution
  1. Prepare simple initial system with know ground state
  2. Change system *slowly* towards the final system



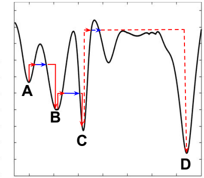
- Runtime $\propto \frac{1}{\Delta E_{\min}}$

DLR

Runtime of Quantum Annealers

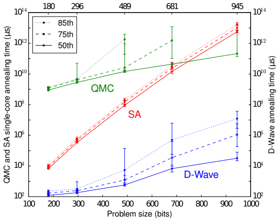
There are indications for a supremacy over classical methods

- Problems with **tall** and **narrow** barriers
- Quantum tunneling



Open questions:

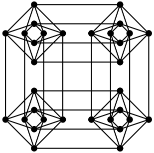
- Is there quantum supremacy for real-world problems?
- What about scaling?

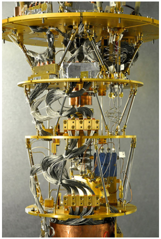


Denchev et. al., Google, arXiv:1512.02206

Quantum Annealer - Status of the Technology

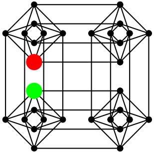
- Commercially available devices from D-Wave Systems
- Customers: Google/NASA, Lockhead Martin/USC, Los Alamos National Laboratory
- USA: Efforts to build own quantum computers by Google, Lincoln Labs, etc. (IARPA QEO)
- D-Wave Hardware Architecture: Limited Connections

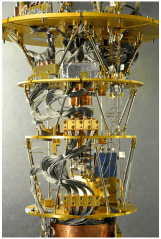




Quantum Annealer - Status of the Technology

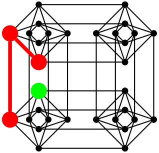
- Commercially available devices from D-Wave Systems
- Customers: Google/NASA, Lockhead Martin/USC, Los Alamos National Laboratory
- USA: Efforts to build own quantum computers by Google, Lincoln Labs, etc. (IARPA QEO)
- D-Wave Hardware Architecture: Limited Connections

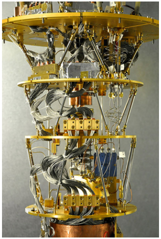




Quantum Annealer - Status of the Technology

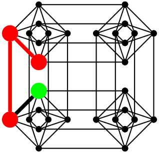
- Commercially available devices from D-Wave Systems
- Customers: Google/NASA, Lockhead Martin/USC, Los Alamos National Laboratory
- USA: Efforts to build own quantum computers by Google, Lincoln Labs, etc. (IARPA QEO)
- D-Wave Hardware Architecture: Limited Connections

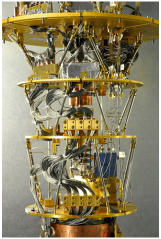




Quantum Annealer - Status of the Technology

- Commercially available devices from D-Wave Systems
- Customers: Google/NASA, Lockhead Martin/USC, Los Alamos National Laboratory
- USA: Efforts to build own quantum computers by Google, Lincoln Labs, etc. (IARPA QEO)
- D-Wave Hardware Architecture: Limited Connections



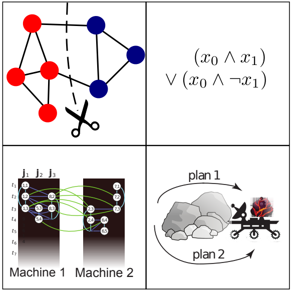


Applications

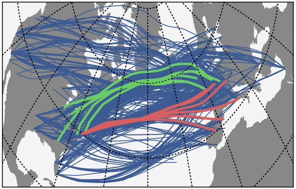
Which problems can be mapped to QUBO?

$$E = \sum_i H_i x_i + \sum_{i \neq j} J_{ij} x_i x_j \qquad \text{with } x_i \in \{0, 1\}$$

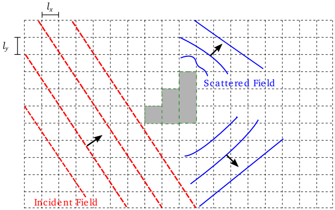
- All NP-Complete Problems. E.g.
  - Graph Partitioning
  - Satisfiablity Problems
- Planning
  - Job-Shop Scheduling
  - Mars-Lander Operations
- Machine Learning



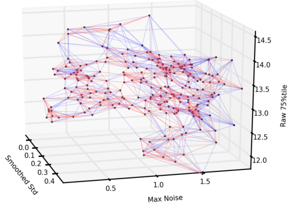
Application for Aerospace Research



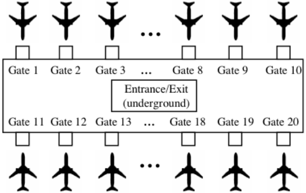
Air traffic management



Radar Cross Section



Telemetry Verification

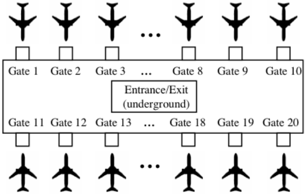


Flight Gate Assignment

Flight Gate Assignment

Flight i

Gate α



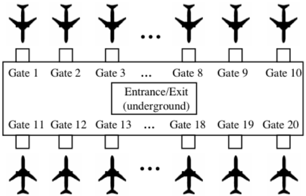
$x_{i\alpha}$: Flight i is assigned to gate α

$$Q = \sum_{i\alpha} \left(n_i^d \; t_\alpha^d + \; n_i^a \; t_\alpha^a \right) x_{i\alpha} + \sum_{ij\alpha\beta} n_{ij} \; t_{\alpha\beta} \, x_{i\alpha} \, x_{j\beta}$$

Flight Gate Assignment

Flight i

Gate α



$x_{i\alpha}$: Flight i is assigned to gate α

\# arriving passengers from flight i

$$Q = \sum_{i\alpha} \left(n_i^d \; t_\alpha^d + n_i^a \; t_\alpha^a \right) x_{i\alpha} + \sum_{ij\alpha\beta} n_{ij} \; t_{\alpha\beta} \, x_{i\alpha} \, x_{j\beta}$$

\# lay over passengers from flight i to j

\# departing passengers for flight i

Flight Gate Assignment Constraints

- One flight per gate: $\forall i: \sum_\alpha x_{i\alpha} = 1$:

$$Q_C = \lambda_C \sum_i \left(\sum_\alpha x_{i\alpha} - 1 \right)^2$$

- No arrival before departure at the same gate

$$x_{i\alpha} \cdot x_{j\alpha} = 0 \quad \forall (i,j) \in C, \forall \alpha$$

with $C = \left\{ (i,j) \mid (t_i^{in} - t_j^{out} < t^{\text{buf}}) \wedge (t_j^{in} - t_i^{out} < t^{\text{buf}}) \right\}$

$$Q_T = \lambda_T \sum_\alpha \sum_{(i,j) \in C} x_{i\alpha} x_{j\alpha}$$

Thank You