Quantum Computing at DLR-SC

Tobias Stollenwerk

Simulation and Software Technology
High Performance Computing Department

Meeting SC-TT, Helmholtz Institute Ulm

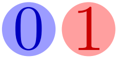
DLR

Content

- Introduction to Quantum Computers
- Quantum Computing at DLR-SC
  - Quantum Annealing
  - Gate-based Quantum Algorithms for Near-Term Devices

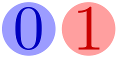
Quantum Bits

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



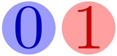
Quantum Bits

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



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

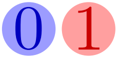
$$|\psi\rangle = a|0\rangle + b|1\rangle$$



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

Quantum Bits

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



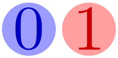
- Quantum state is superposition of "0" **and** "1"

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

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"

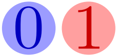
- Quantum state is superposition of "0" **and** "1"

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

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

Quantum Bits

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

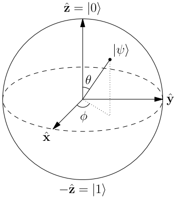


- Quantum state is superposition of "0" **and** "1"

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

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

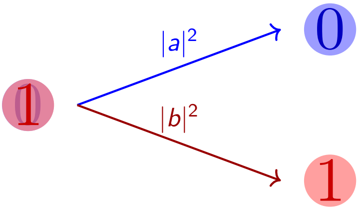
$$|\psi\rangle = a|0\rangle + b|1\rangle$$

Quantum Bits

- Measure the state

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

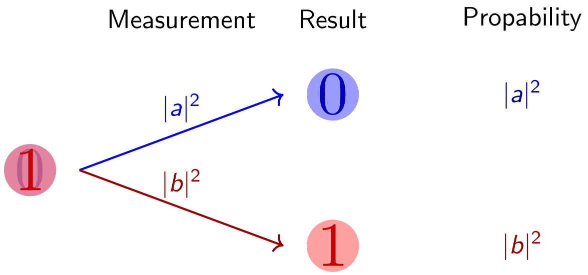
Measurement Result



Quantum Bits

- Measure the state

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

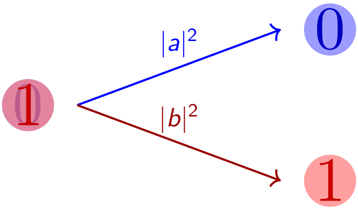


Quantum Bits

- Measure the state

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

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	$\lvert b\rvert^2$	$\lvert\psi\rangle = \lvert 1\rangle$



Quantum Bits

- Measure the state

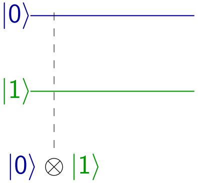
$$|\psi\rangle = a|0\rangle + b|1\rangle$$

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

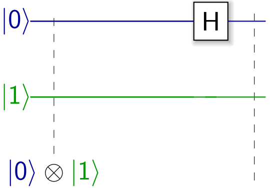
Universal Quantum Computer

Gate model: Manipulate quantum states through quantum gates



Universal Quantum Computer

Gate model: Manipulate quantum states through quantum gates



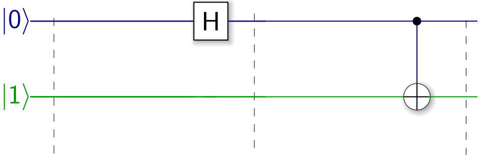
$$|0\rangle \otimes |1\rangle$$

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

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

Universal Quantum Computer

Gate model: Manipulate quantum states through quantum gates



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

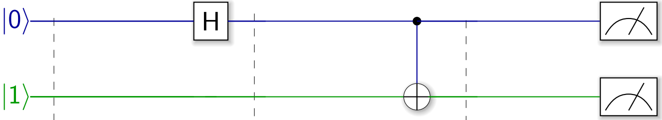
$$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \otimes |1\rangle$$

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

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

Universal Quantum Computer

Gate model: Manipulate quantum states through quantum gates



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

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

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

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

Quantum Algorithms

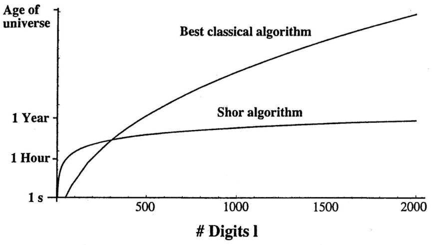
Algorithm	Runtime classical	Runtime quantum	Application
Deutsch-Josza	$2^n/2$	1	Academical
Grover's search algorithm	n	$\sqrt{n}$	Database
Shor's Factorization Algorithm	Exponential	Polynomial	Cryptography

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

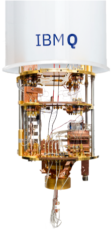
Quantum Algorithms

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



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

Rigetti Chip Architecture

Google Bristlecone Chip

Near-Term Quantum Algorithms

Question

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

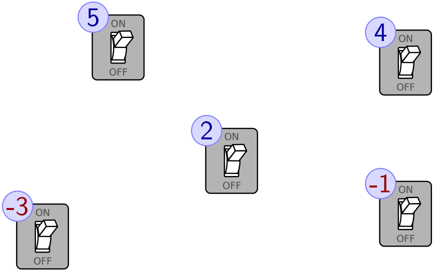
Answer

- 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 Annealer

- Optimizer for Ising problems

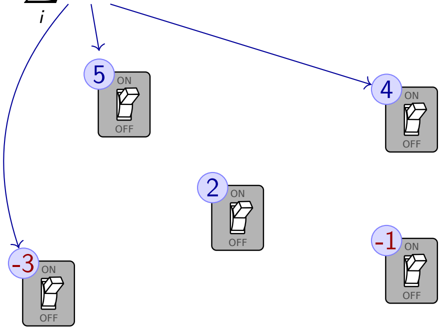
$$E = \sum_i H_i \, s_i \qquad \text{with } x_i \in \{-1, 1\}$$



Quantum Annealer

- Optimizer for Ising problems

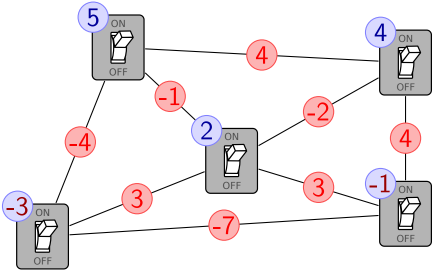
$$E = \sum_i H_i \, s_i \qquad \text{with } x_i \in \{-1, 1\}$$



Quantum Annealer

- Optimizer for Ising problems

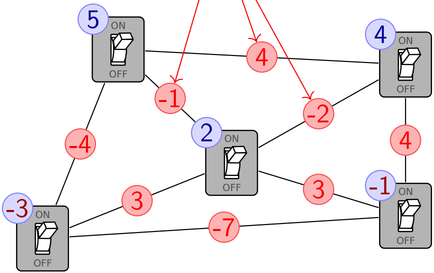
$$E = \sum_i H_i \, s_i \qquad \text{with } x_i \in \{-1, 1\}$$



Quantum Annealer

- Optimizer for Ising problems

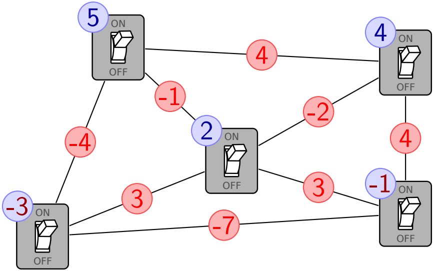
$$E = \sum_i H_i \, s_i + \sum_{i \neq j} J_{ij} \, s_i s_j \qquad \text{with } x_i \in \{-1, 1\}$$



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?

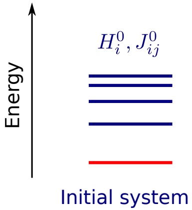
- How do we bring the system in to the final state?
- Solution: Adiabatic evolution
  1. Prepare simple initial system with know ground state
  2. Change system *slowly* towards the final system

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

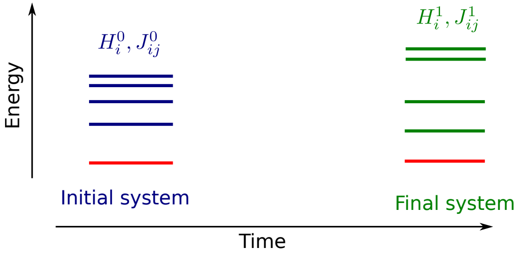
Quantum Annealing - How does it work?

- 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



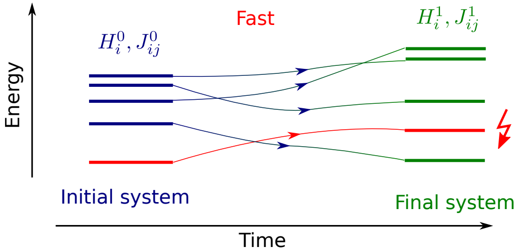
Quantum Annealing - How does it work?

- 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



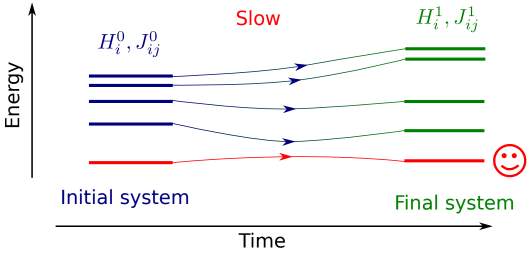
Quantum Annealing - How does it work?

- 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



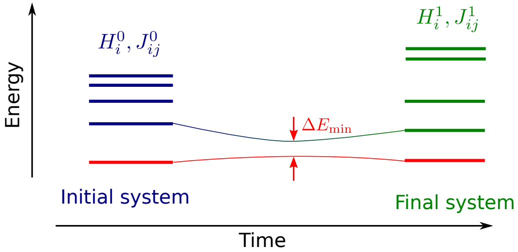
Quantum Annealing - How does it work?

- 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

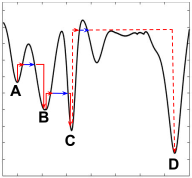


- Runtime $\propto \frac{1}{\Delta E_{\min}}$

Runtime of Quantum Annealers

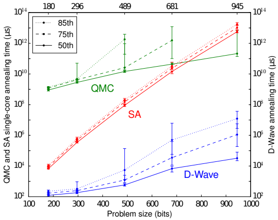
There are indications for a supremacy over classical methods

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



Open questions:

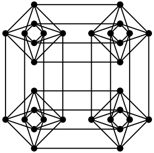
- Is there quantum supremacy for real-world problems?
- What about scaling?

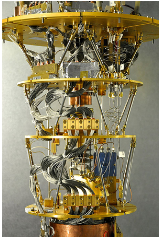


Denchev et. al., Google, arXiv:1512.02206

Quantum Annealer - Status of the Technology

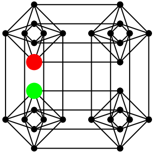
- Commercially available devices from D-Wave Systems
- Customers: Google/NASA, Lockhead Martin/USC, Los Alamos National Laboratory
- USA: Efforts to build own quantum computers by Google, Lincoln Labs, etc. (IARPA QEO)
- D-Wave Hardware Architecture: Limited Connections

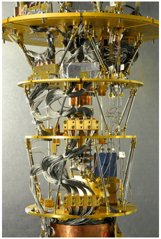




Quantum Annealer - Status of the Technology

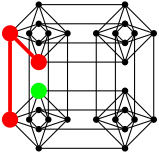
- Commercially available devices from D-Wave Systems
- Customers: Google/NASA, Lockhead Martin/USC, Los Alamos National Laboratory
- USA: Efforts to build own quantum computers by Google, Lincoln Labs, etc. (IARPA QEO)
- D-Wave Hardware Architecture: Limited Connections

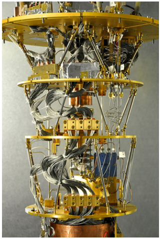




Quantum Annealer - Status of the Technology

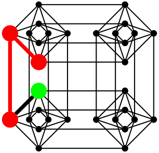
- Commercially available devices from D-Wave Systems
- Customers: Google/NASA, Lockhead Martin/USC, Los Alamos National Laboratory
- USA: Efforts to build own quantum computers by Google, Lincoln Labs, etc. (IARPA QEO)
- D-Wave Hardware Architecture: Limited Connections

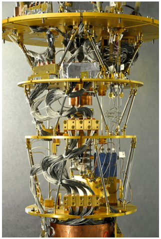




Quantum Annealer - Status of the Technology

- 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



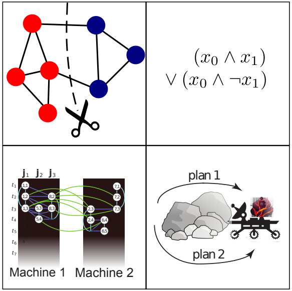


Applications

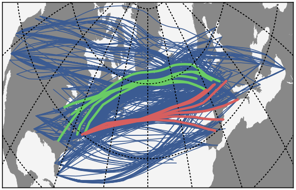
Which problems can be mapped to QUBO?

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

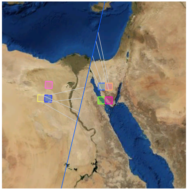
- All NP-Complete Problems. E.g.
  - Graph Partitioning
  - Satisfiablity Problems
- Planning
  - Job-Shop Scheduling
  - Mars-Lander Operations
- Machine Learning



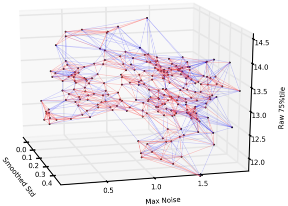
Application for Aerospace Research



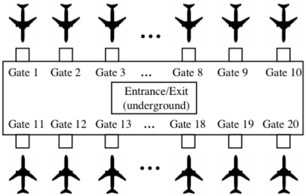
Air traffic management



Satellite Mission Planning

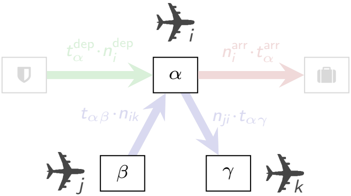


Telemetry Verification



Flight Gate Assignment

Flight Gate Assignment - Cost Function

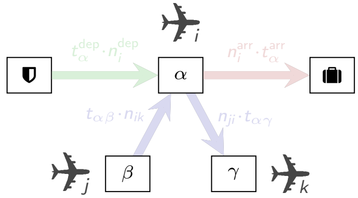


Minimizing the total transfer time with the cost function

$$C = \sum_{i\alpha} n_i^{arr} t_\alpha^{arr} x_{i\alpha} + \sum_{i\alpha} n_i^{dep} t_\alpha^{dep} x_{i\alpha} + \sum_{ij\alpha\beta} n_{ij} t_{\alpha\beta} x_{i\alpha} x_{j\beta}$$

→ Quadratic Assignment problem

Flight Gate Assignment - Cost Function

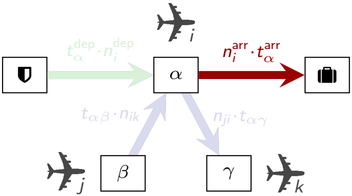


Minimizing the total transfer time with the cost function

$$C = \sum_{i\alpha} n_i^{\text{arr}} t_\alpha^{\text{arr}} x_{i\alpha} + \sum_{i\alpha} n_i^{\text{dep}} t_\alpha^{\text{dep}} x_{i\alpha} + \sum_{ij\alpha\beta} n_{ij} t_{\alpha\beta} x_{i\alpha} x_{j\beta}$$

$\rightarrow$ Quadratic Assignment problem

Flight Gate Assignment - Cost Function

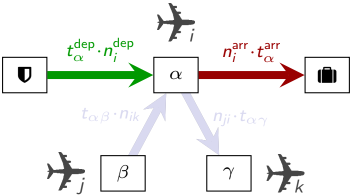


Minimizing the total transfer time with the cost function

$$C = \sum_{i\alpha} n_i^{\text{arr}} t_\alpha^{\text{arr}} x_{i\alpha} + \sum_{i\alpha} n_i^{\text{dep}} t_\alpha^{\text{dep}} x_{i\alpha} + \sum_{ij\alpha\beta} n_{ij} t_{\alpha\beta} x_{i\alpha} x_{j\beta}$$

→ Quadratic Assignment problem

Flight Gate Assignment - Cost Function

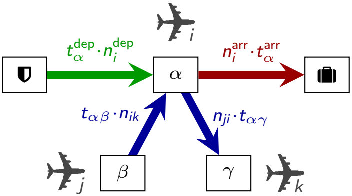


Minimizing the total transfer time with the cost function

$$C = \sum_{i\alpha} n_i^{arr} t_\alpha^{arr} x_{i\alpha} + \sum_{i\alpha} n_i^{dep} t_\alpha^{dep} x_{i\alpha} + \sum_{ij\alpha\beta} n_{ij} t_{\alpha\beta} x_{i\alpha} x_{j\beta}$$

→ Quadratic Assignment problem

Flight Gate Assignment - Cost Function

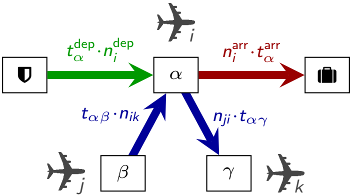


Minimizing the total transfer time with the cost function

$$C = \sum_{i\alpha} n_i^{\text{arr}} t_\alpha^{\text{arr}} x_{i\alpha} + \sum_{i\alpha} n_i^{\text{dep}} t_\alpha^{\text{dep}} x_{i\alpha} + \sum_{ij\alpha\beta} n_{ij} t_{\alpha\beta} x_{i\alpha} x_{j\beta}$$

→ Quadratic Assignment problem

Flight Gate Assignment - Cost Function



Minimizing the total transfer time with the cost function

$$C = \sum_{i\alpha} n_i^{\text{arr}} t_\alpha^{\text{arr}} x_{i\alpha} + \sum_{i\alpha} n_i^{\text{dep}} t_\alpha^{\text{dep}} x_{i\alpha} + \sum_{ij\alpha\beta} n_{ij} t_{\alpha\beta} x_{i\alpha} x_{j\beta}$$

→ Quadratic Assignment problem

Flight Gate Assignment - Constraints

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

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

- No arrival before departure at the same gate

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

with F : set of forbidden flight pairs

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

Flight Gate Assignment - Constraints

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

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

- No arrival before departure at the same gate

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

with F : set of forbidden flight pairs

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

Flight Gate Assignment - Constraints

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

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

- No arrival before departure at the same gate

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

with F : set of forbidden flight pairs

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

Flight Gate Assignment - Constraints

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

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

- No arrival before departure at the same gate

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

with F : set of forbidden flight pairs

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

Flight Gate Assignment - Constraints

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

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

- No arrival before departure at the same gate

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

with F : set of forbidden flight pairs

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

Hybrid Quantum-Classical Algorithms

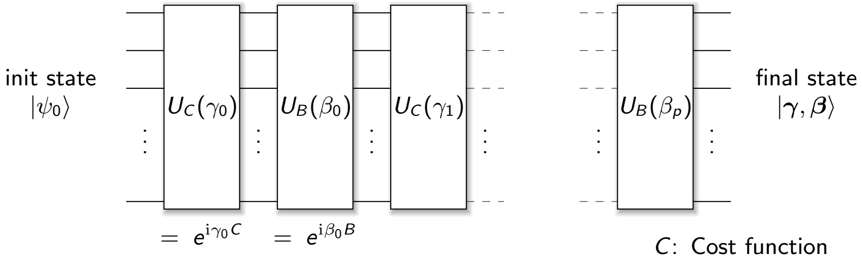
Problem

$$\min_{\gamma} \langle \gamma | C | \gamma \rangle$$

Algorithm

- Prepare parametrized state $|\gamma\rangle$ on gate based quantum computer
- Calculate expectation value $\langle \gamma | C | \gamma \rangle$
- Optimize parameters γ with classical optimization

Quantum Approximate Optimization Algorithm



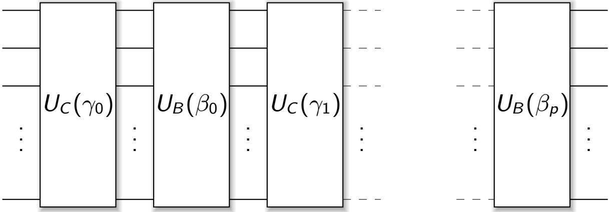
C: Cost function
B: Mixer

Hadfield et. al., arXiv:1709.03489

DLR

Quantum Approximate Optimization Algorithm

init state
$|\psi_0\rangle$



final state
$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$

$= \; e^{\mathrm{i}\gamma_0 C}$ $\quad = \; e^{\mathrm{i}\beta_0 B}$

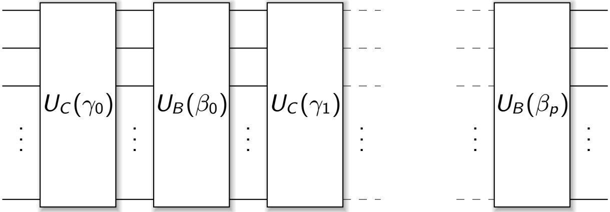
C: Cost function
B: Mixer

Get expectation value $\langle \boldsymbol{\gamma}, \boldsymbol{\beta} | C | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle$ through multiple measurements

Hadfield et. al., arXiv:1709.03489

Quantum Approximate Optimization Algorithm

init state
$|\psi_0\rangle$



final state
$|\boldsymbol{\gamma}, \boldsymbol{\beta}\rangle$

$= e^{\mathrm{i}\gamma_0 C}$ $= e^{\mathrm{i}\beta_0 B}$

C: Cost function
B: Mixer

Get expectation value $\langle \boldsymbol{\gamma}, \boldsymbol{\beta} | C | \boldsymbol{\gamma}, \boldsymbol{\beta} \rangle$ through multiple measurements

Optimize classically

Hadfield et. al., arXiv:1709.03489

QAOA for Constraint Optimization

Find suitable mixer B

- That keeps valid states valid
- That explores the whole space

Example:

$$\sum_{i\alpha} x_{i\alpha} = 1$$

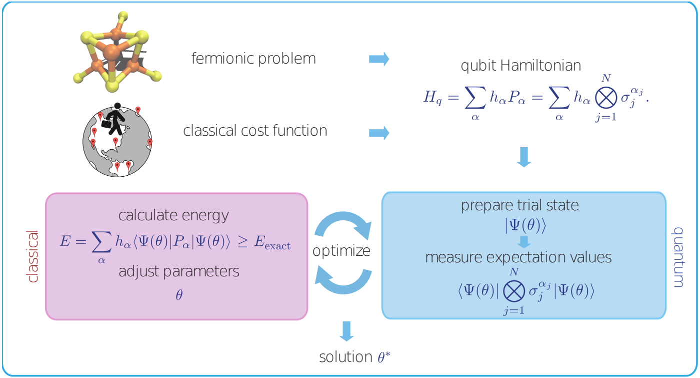
Use SWAP mixer:

$$\begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{matrix} \Rightarrow \begin{matrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{matrix}$$

Hadfield et. al., arXiv:1709.03489

Quantum Chemistry - Variational Quantum Eigensolver

- Calculate ground state of molecules



Moll et.al. arXiv:1710.01022

HHL Algorithm for Radar Cross Section

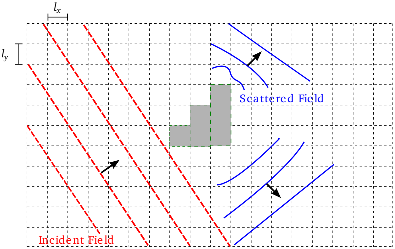
HHL Algorithm

- 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

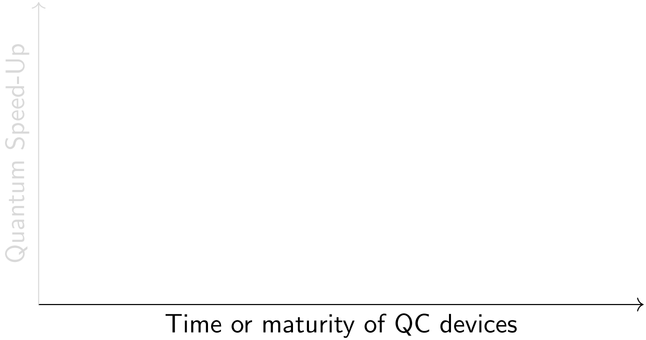
- FEM calculation for stationary scattering problem
- Scattering cross section is of the form

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

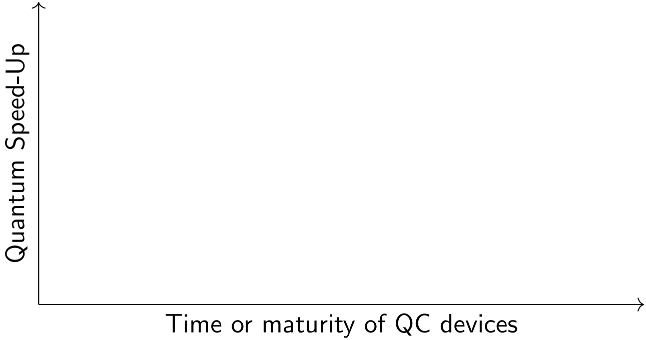
Resource Analysis

- Matrix oracle on QC
- Implement matrix oracale on classical reversible circuits
- Estimate resources on QC

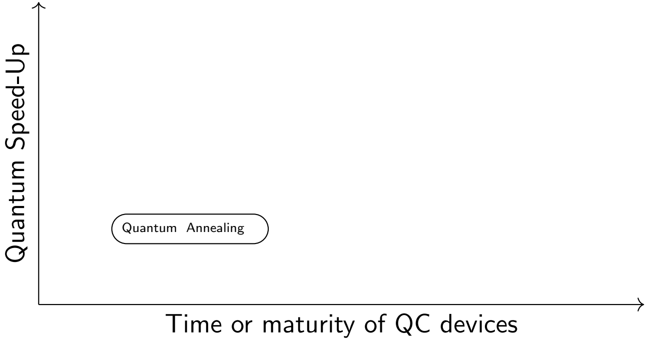
Personal View on Speed-Up and Timescales



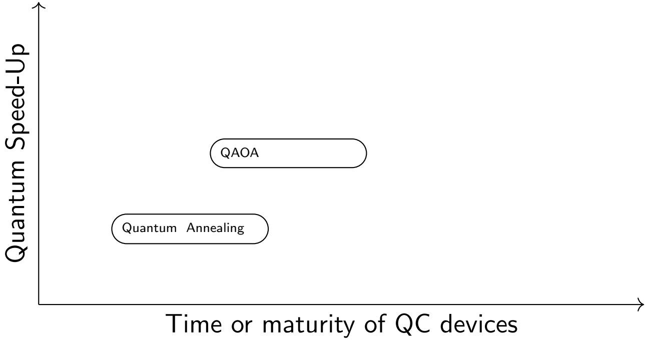
Personal View on Speed-Up and Timescales



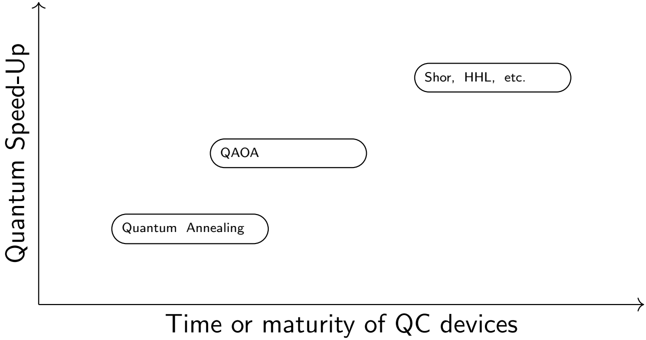
Personal View on Speed-Up and Timescales



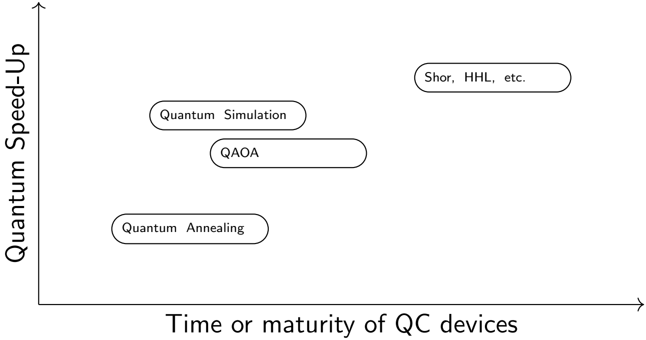
Personal View on Speed-Up and Timescales



Personal View on Speed-Up and Timescales



Personal View on Speed-Up and Timescales



Thank You