Quantum Computer Systems: Research for Noisy Intermediate-Scale Quantum Computers

Chapter 3

Yongshan Ding and Frederic T. Chong.

Presenter: Yu-Ching Hu 04/06/2021

Outline

- Quantum information processing
- Quantum parallelism
- Quantum oracles
- Complexity and costs
- Quantum algorithms
- Summary

Quantum processing

- Encode information into a small number of qubits
- Build up entanglement and interference during the algorithm
- Design a final measurement

Quantum circuit

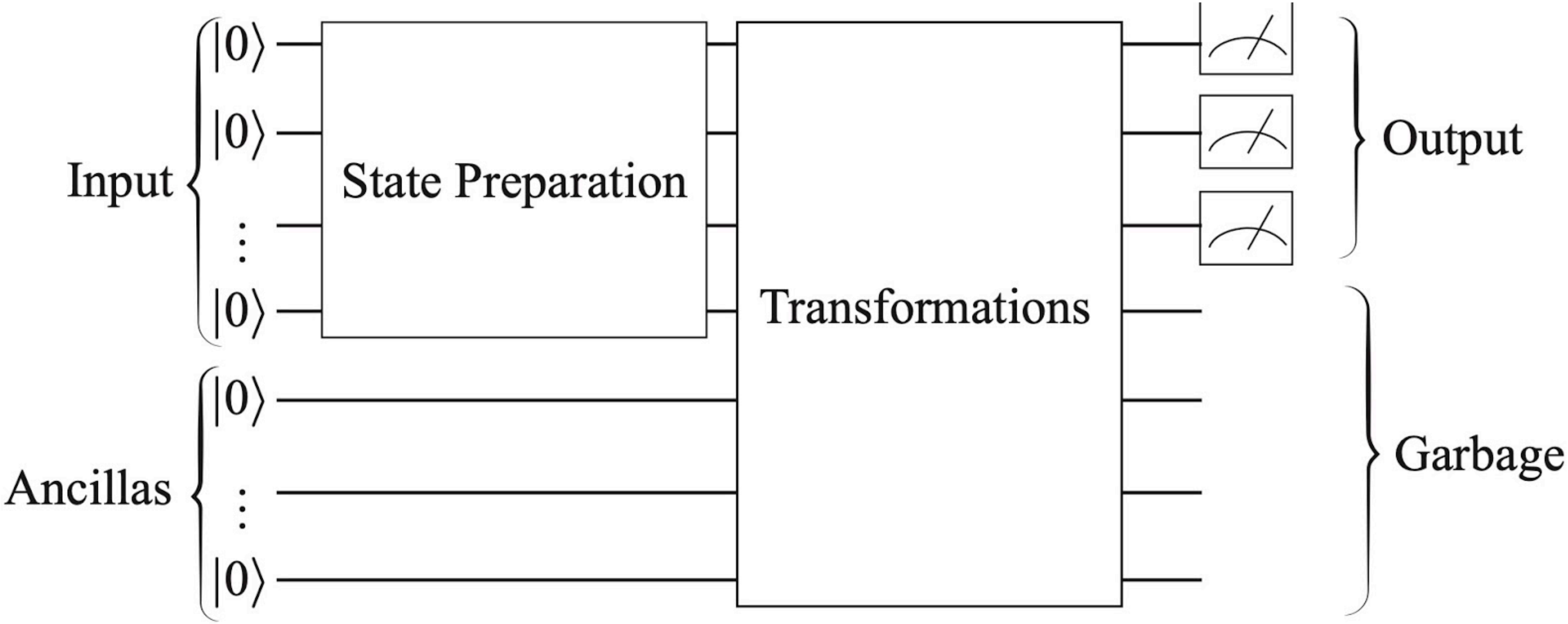

Fig. 1: A typical quantum circuit implementation of a quantum algorithm.

Quantum parallelism

- **Query model**
  - Black-box function (oracle)
- **Query complexity**
  - Def — # of queries required by the algorithm

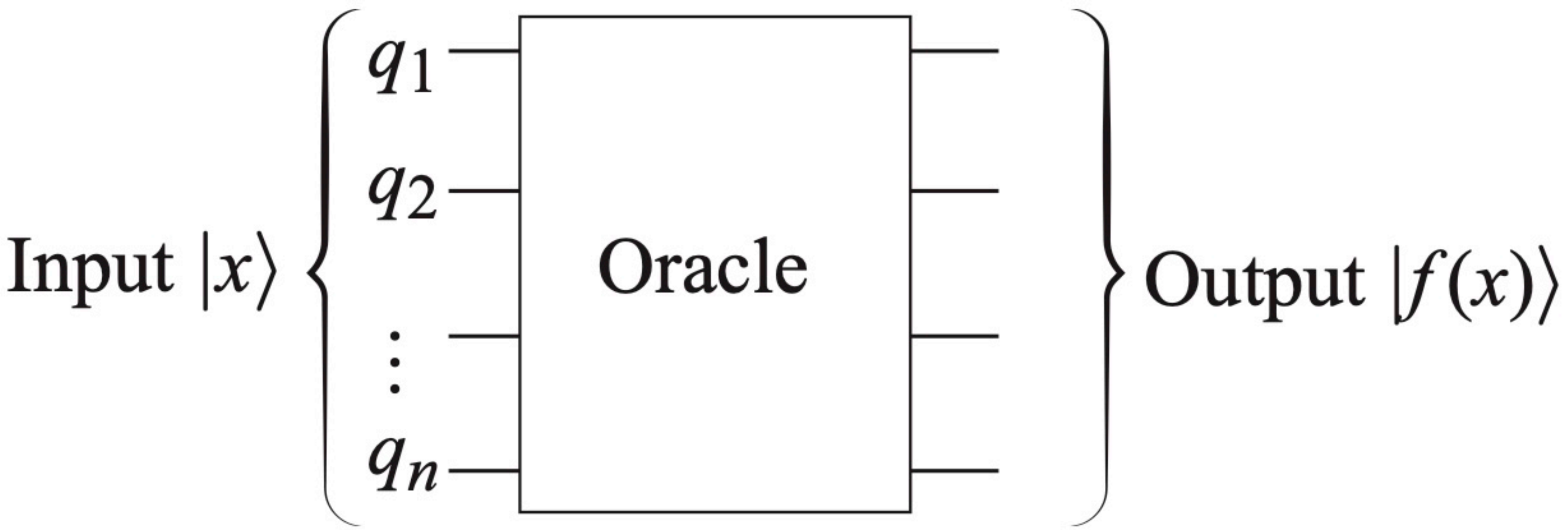

Fig. 2: An oracle that computes f(x).

Practical examples

- **Searching problem**
  - Find x such that $f(x) = 1$
- **Period-finding problem**
  - Find p such that $f(x) = f(x + p)$ for all x (from 0…0 to 1…1)
- **Collision problem**
  - Find x, y such that $f(x) = f(y)$

Quantum query algorithm v.s. Classical query algorithm

- **Quantum query algorithm**
  - Can pass several inputs at once
  - $$\sum_i |x_i\rangle \longrightarrow \sum_i (-1)^{f(x_i)} |x\rangle$$

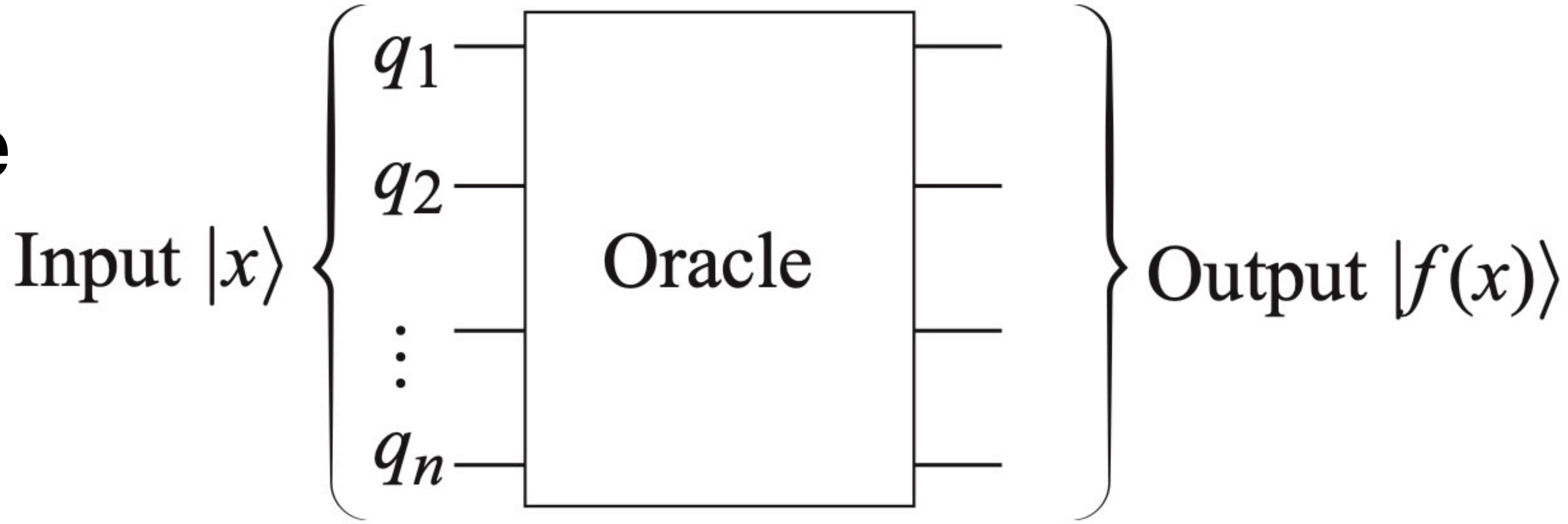

Fig. 2: An oracle that computes f(x).

- **Classical query algorithm**
  - Single input at a time

Quantum oracles

- $f: \{0,1\}^n \rightarrow \{0,1\}^m$

$|x\rangle$ input qubits (input registers)

$|y\rangle$ output qubits (output registers or ancilla)

- **XOR oracle (O_f)**
  - transform a quantum state from $|x\rangle \oplus |y\rangle$ to $|x\rangle \oplus |y \oplus f(x)\rangle$
- **Phase oracle ($O_f^{\pm}$)**
  - transform a quantum state from $|x\rangle \oplus |y\rangle$ to $|x\rangle \oplus (-1)^{f(x)\cdot y}|y\rangle$, where

$f(x) \cdot y = \sum_i f(x)_i y_i$ mod 2 is the inner product of the two bit-strings

Complexity (Costs)

1. **Device-independent computational complexity**

   ① ***Time complexity*** — unitary transformation U is related to the number of gates of the smallest circuit that implements U.

   ② ***Query complexity*** — number of times an algorithm needs to query a given black-box function (oracle) to solve a problem.

Def of P: all decision problems solvable by a polynomial-size uniform circuit family deterministically.

Def of BPP: all decision problems solvable by a polynomial-size uniform ***random*** circuit family with high probability.

P: polynomial time

BPP: bounded-error probabilistic polynomial time

Computational complexity

Def of BQP: all decision problems solvable by a polynomial-size uniform ***quantum*** circuit family with high probability.

BQP: bounded-error quantum polynomial time

Implementation cost

2. Device-dependent implementation cost

① Precision

$$D(U_{noisy}\rho_{in}U^{\dagger}_{noisy}, U_{ideal}\rho_{in}U^{\dagger}_{ideal}) \text{ ----- (1)}$$

Worst-case success rate of a quantum circuit under qubit decoherence and gate noise:

$$P_{success} = \prod_{g\in G}(1-\epsilon_g)\cdot\prod_{q\in Q}(1-\epsilon_q) \text{ ----- (2)}$$

ϵ_g is avg. gate error rate

ϵ_q is captured by modeling T_1, T_2 during idle or gate time

(* in Ch 2.3.3)

② Resource cost

- Qubit count (circuit width)
- Gate count
- Circuit depth
- Communication cost
- Spacetime volume $(\#qubits \times time)$

Gate-based quantum algorithms

- *Deutsch-Josza algorithm*
- *Bernstein-Vazirani algorithm*
- ***Some caveats***:
  1. *No existing applications for these problems.*
  2. *No fair comparison between classical functions and quantum algorithms*
  3. *Many examples can be made more efficient with randomized algorithms*

Deutsch-Josza Algorithm

- **Problem statement**: Given an oracle implementing a function $f : \{0,1\}^n \longrightarrow \{0,1\}$ which is promised to be either constant or balanced. We want to determine whether it is constant or balanced.
- **Classical solution**: The worst-case requires to query $2^{n-1} + 1$ times, covering inputs for more than half of the domain $\{0,1\}^n$ in order to determine whether f is constant or balanced.

Deutsch-Josza Algorithm

- **Quantum solution**:

① $|0\rangle^{\oplus n}$

② $|+\rangle^{\oplus n} = \dfrac{|0\rangle + |1\rangle}{\sqrt{2}} \cdots \dfrac{|0\rangle + |1\rangle}{\sqrt{2}} = \dfrac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$

③ $\dfrac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle$

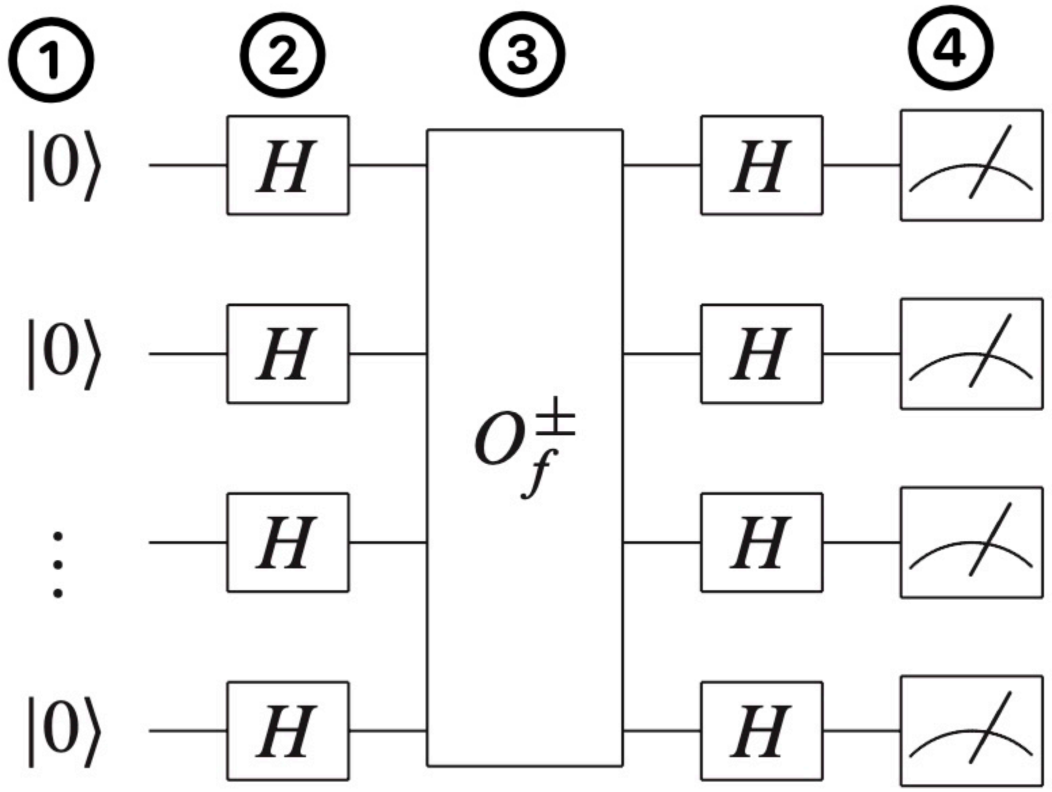

Fig. 3: Quantum circuit for Deutsch-Josza algorithm.

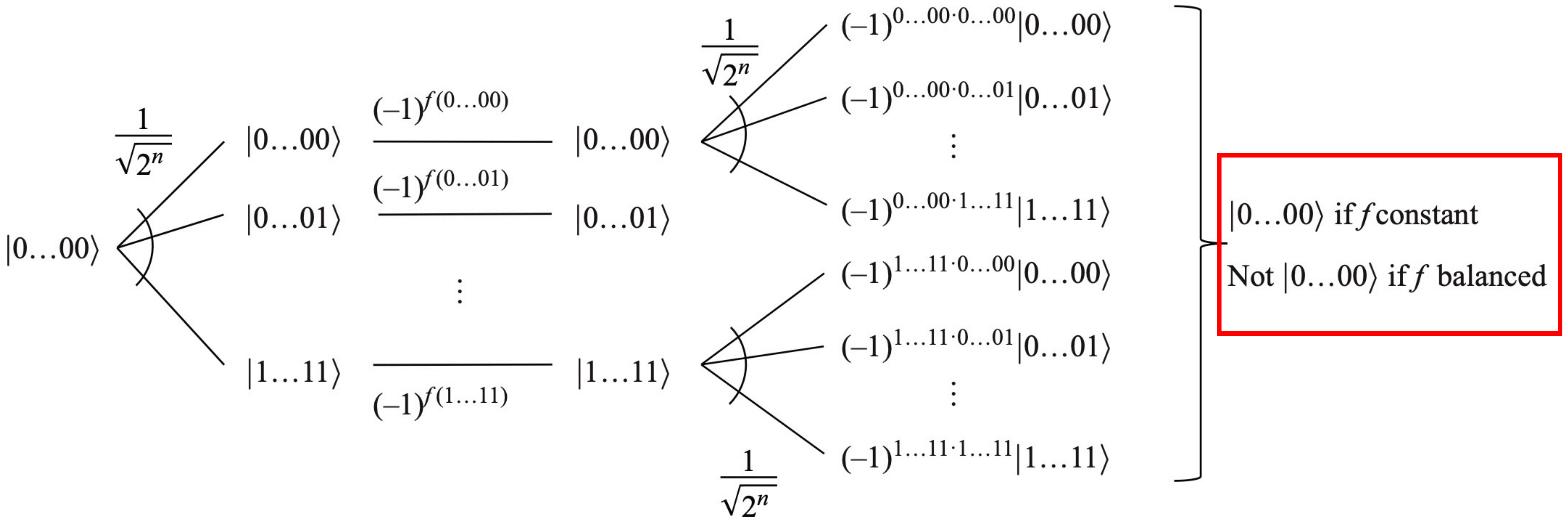

Fig. 4: Feynman path diagram for Deutsch–Josza algorithm.

$$= \begin{cases} (-1)^{f(x)} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle = (-1)^{f(x)}|+\rangle^{\otimes n} & \text{if } f(x) \text{ constant} \\ \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}|x\rangle & \text{if } f(x) \text{ balanced} \end{cases}$$

④ $$\begin{cases} (-1)^{f(x)}|0\rangle^{\otimes n} & \text{if } f(x) \text{ constant} \\ \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} H^{\otimes n}|x\rangle & \text{if } f(x) \text{ balanced} \end{cases}$$

$$\sum_{x\in\{0,1\}^n} (-1)^{f(x)} \text{ cancels out}$$

$$= \begin{cases} (-1)^{f(x)}|0\rangle^{\otimes n} & \text{if } f(x) \text{ constant} \\ \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)} \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{xy}|y\rangle & \text{if } f(x) \text{ balanced} \end{cases}$$

$$= \begin{cases} (-1)^{f(x)}|0\rangle^{\otimes n} & \text{if } f(x) \text{ constant} \\ \frac{1}{2^n} \sum_{x\in\{0,1\}^n} \sum_{y\in\{0,1\}^n} (-1)^{xy+f(x)}|y\rangle & \text{if } f(x) \text{ balanced} \end{cases}$$

Speedup over classical algorithm by $(2^{n-1}+1)$

Bernstein-Vazirani Algorithm

- **Problem statement**: Given an oracle access to $f : \{0,1\}^n \longrightarrow \{0,1\}$ and a promise that the function $f(x) = s \cdot x = \sum_{i=1}^{n} s_i \cdot x_i \bmod 2$, where s is a secret string that the algorithm is trying to learn.
- **Classical solution**: Brute-force to find out the answer by giving n inputs.

$$
\begin{aligned}
f(100\ldots0) &= s_1 \\
f(010\ldots0) &= s_2 \\
&\vdots \\
f(000\ldots1) &= s_n
\end{aligned}
$$

- **Quantum solution**: Can do this in just ***one*** query.

Bernstein-Vazirani Algorithm

① $|0\rangle^{\otimes n}$

② $|+\rangle^{\otimes n} = \frac{1}{\sqrt{2}} \sum_{x \in \{0,1\}^n} |x\rangle$

③ $\frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{s \cdot x} |x\rangle$

$= \frac{1}{2}(|0\rangle + (-1)^{s_1} |1\rangle) \otimes \frac{1}{2}(|0\rangle + (-1)^{s_2} |1\rangle) \otimes \cdots \otimes \frac{1}{2}(|0\rangle + (-1)^{s_n} |1\rangle).$

④ 0 if $s_i = 0$ and 1 if $s_i = 1$.

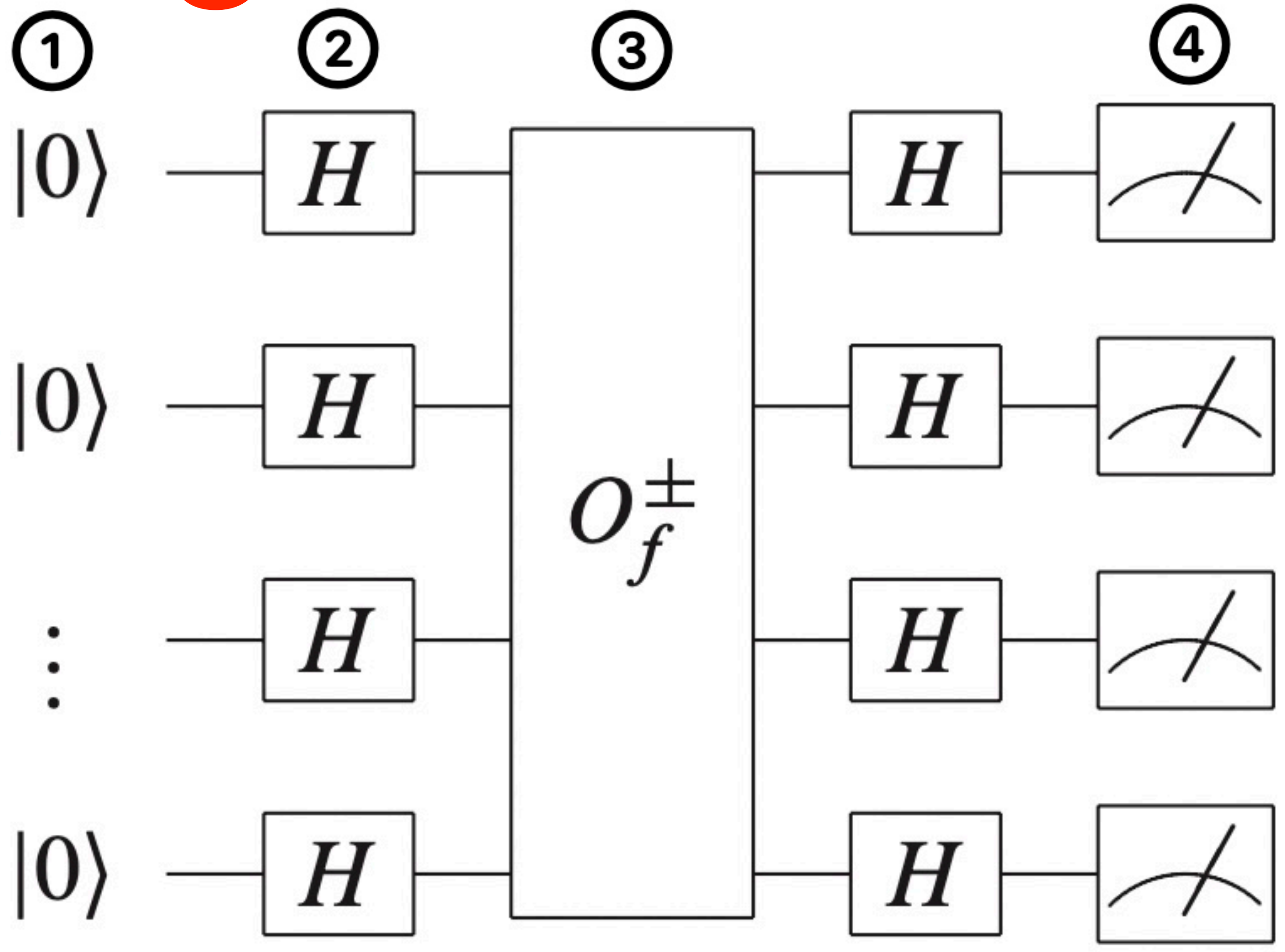

Fig. 5: Quantum circuit for Bernstein–Vazirani algorithm.

- The state of i-th qubit depends on s_i:

if $s_i = 0$ then qubit i is $| + \rangle$

if $s_i = 1$ then qubit i is $| - \rangle$

NISQ quantum algorithms

1. Variational Quantum Eigensolver (**VQE**)
2. Quantum Approximate Optimization Algorithm (**QAOA**)

Variational Quantum Eigensolver

- $\langle \psi | H | \psi \rangle \geq E_0$
- Find the lowest eigenvalue
- Guess & check

H: hermitian matrix

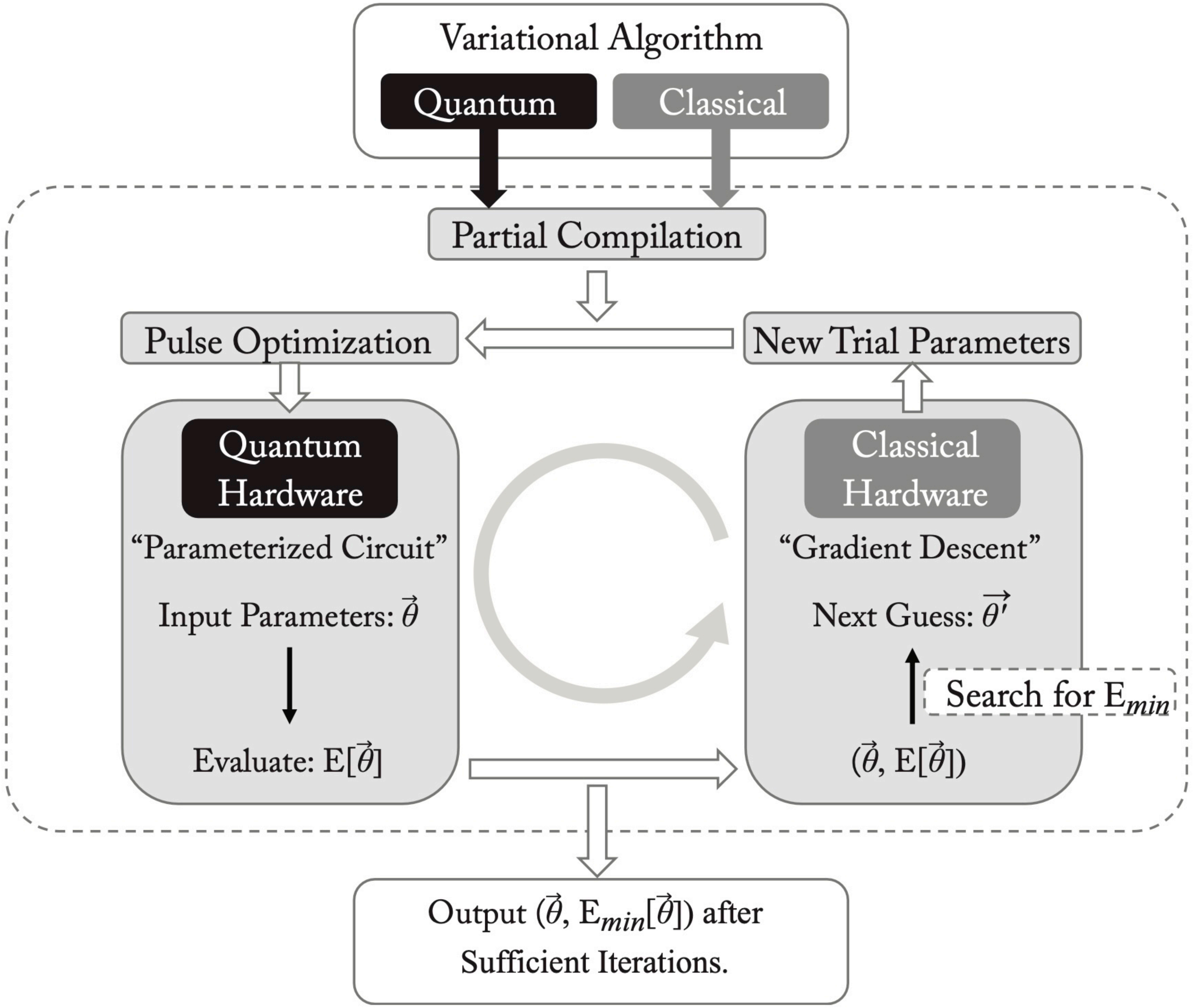

Fig. 6: Illustration of a variational quantum.

Quantum Approximate Optimization Algorithm

- Can be applied to MaxCut/Clustering problem
- $H = \dfrac{1}{2} \sum_{edges\ i,j} (I - Z_i Z_j)$, where Z_i, Z_j are the Pauli Z matrix for i-th and j-th vertex.

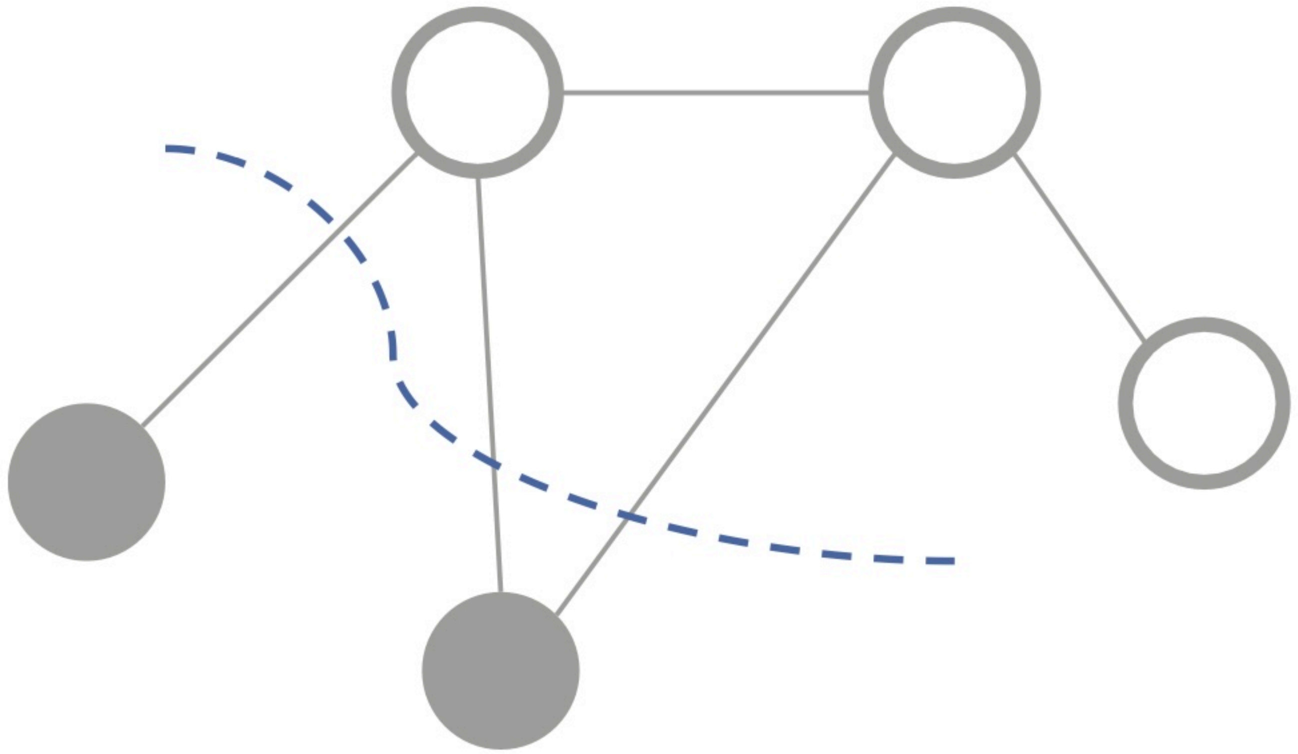

Fig. 7: A MaxCut of a graph.

Summary

- Quantum computing provides an entirely new way of solving computational problems efficiently (**exponential speedup**).
- Large gate-based algorithms have shown practical potential of quantum computers.
  - Shor's algorithm
  - Grover's algorithm
- In the short-term, quantum devices are still limited in fidelity and size and are not fault tolerant.
- One of the biggest challenges of the NISQ era is to develop algorithms that run on NISQ computers.

Q&A