Lecture

# Quantum Computing

(CS5070)

## What else?

**Professor Dr. rer. nat. habil. Sven Groppe**
https://www.ifis.uni-luebeck.de/index.php?id=groppe

# What else? - Simon's algorithm

- Simon's problem is proven to be solved exponentially faster on a quantum computer than on a classical computer
  - Simon's algorithm, served as the inspiration for Shor's algorithm
  - Simon's algorithm: $O(n)$ queries to the black box
  - Classical algorithm at least $\Omega(2^{\frac{n}{2}})$ queries
- Given a function (implemented by a black box or oracle) $f : \{0,1\}^n \rightarrow \{0,1\}^n$ with the promise that, for some unknown $s \in \{0,1\}^n$, for all $x, y \in \{0,1\}^n$, $f(x) = f(y)$ if and only if $x \oplus y \in \{0^n, s\}$, the goal is to identify $s$ by making as few queries to $f(x)$ as possible.
  - i.e., distinguishing the $s = 0^n$ case, where the function is one-to-one, from the $s \neq 0^n$ case, where the function is two-to-one and satisfies $f(x) = f(x \oplus s)$
  - little practical value

# What else? - Bernstein–Vazirani algorithm

- designed to prove an oracle separation between complexity classes BQP and BPP
  - BQP: bounded-error quantum polynomial time is the class of decision problems solvable by a quantum computer in polynomial time, with an error probability of at most $\frac{1}{3}$ for all instances
  - BPP: bounded-error probabilistic polynomial time is the class of decision problems solvable by a probabilistic Turing machine in polynomial time with an error probability bounded away from $\frac{1}{3}$ for all instances
- Given an oracle that implements a function $f: \{0,1\}^n \to \{0,1\}$ in which $f(x)$ is promised to be the dot product between $x$ and a secret string $s \in \{0,1\}^n$ modulo 2,
  $f(x) = x \cdot s = x_1 s_1 \oplus x_2 s_2 \oplus \cdots \oplus x_n s_n$, find $s$
- Generalization is the hidden linear function problem

# What else? - Quantum Fourier Transform

- used as basic routine in other quantum algorithms for
  - factoring and computing the discrete logarithm,
  - the quantum phase estimation algorithm for estimating the eigenvalues of a unitary operator, and
  - algorithms for the hidden subgroup problem

| | Classical discrete Fourier transform | Quantum Fourier transform |
|---|---|---|
| **Input** | vector $(x_0, x_1, \ldots, x_{N-1})$ | quantum state $\|x\rangle = \sum_{i=0}^{N-1} x_i \|i\rangle$ |
| **Notation** | $N = 2^n$ with $n$ #(qu)bits, $\omega_N = e^{\frac{2 \cdot \pi \cdot i}{N}}$ and $\omega_N^n$ is an $N$-th root of unity | |
| **Output** | vector $(y_0, y_1, \ldots, y_N - 1) \in \mathbb{C}^N$ according to $y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \omega_N^{-kn}$, $k = 0, 1, 2, \ldots, N-1$ | quantum state $\sum_{i=0}^{N-1} y_i \|i\rangle$ according to $y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \omega_N^{nk}$, $k = 0, 1, 2, \ldots, N-1$ |
| **Circuit Depth** | $O(n \cdot 2^n)$ gates | Simple Decomposition: $O(n^2)$ Hadamard gates and controlled phase shift gates<br>Best known one: $O(n \log n)$ gates [HH'00] |

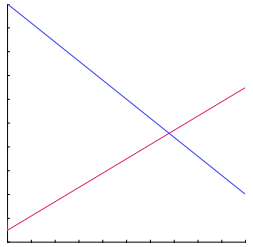
# What else? - Hidden subgroup problem

- Given a group $G$, a subgroup $H \leq G$, and a set $X$, $f : G \to X$ **hides** the subgroup $H$ if for all $g1, g2 \in G$, $f(g1) = f(g2)$ if and only if $g1H = g2H$. Equivalently, the function $f$ is constant on the cosets of $H$, while it is different between the different cosets of $H$.
- **Hidden subgroup problem (HSP):** Let $G$ be a group, $X$ a finite set, and $f : G \to X$ a function that hides a subgroup $H \leq G$. The function $f$ is given via an oracle, which uses $O(log|G| + log|X|)$ bits. Using information gained from evaluations of $f$ via its oracle, determine a generating set for $H$.
  - used within Shor's algorithm
  - an efficient quantum algorithm for the HSP for the symmetric group would give a quantum algorithm for the graph isomorphism [EH'99]
  - An efficient quantum algorithm for the HSP for the dihedral group would give a quantum algorithm for the $poly(n)$ unique shortest vector problems [R'03]

# What else? - Phase Estimation

- **Eigenvector/-value:** If $T$ is a linear transformation from a vector space $V$ over a field $F$ into itself and $v$ is a nonzero vector in $V$, then $v$ is an eigenvector of $T$ if $T(v)$ is a scalar multiple of $v$, i.e., $T(v) = \lambda v$ , where $\lambda$ is a scalar in $F$ (called eigenvalue)
- Let $U$ be a unitary operator that operates on $m$ qubits with an eigenvector $|\psi\rangle$, such that $U|\psi\rangle = e^{2 \cdot \pi \cdot i \cdot \theta} |\psi\rangle , 0 \leq \theta < 1$.
  **Phase estimation** finds the eigenvalue $e^{2 \cdot \pi \cdot i \cdot \theta}$ of $|\psi\rangle$, which in this case is equivalent to estimating the phase $\theta$, to a finite level of precision $\varepsilon$.
  - $O\left(\log\left(\frac{1}{\varepsilon}\right)\right)$ qubits (without counting the ones used to encode the eigenvector state) and $O\left(\frac{1}{\varepsilon}\right)$ controlled-$U$ operations
- used as a subroutine in other quantum algorithms, such as
  - Shor's algorithm and
  - HHL
- Other algos for eigenvalues: variational quantum eigensolver (VQE)

# What else? - Quantum Linear Systems Problem (QLSP)



Find $\vec{x}$, such that $A \cdot \vec{x} = \vec{b}$, where $\vec{x}$, $\vec{b}$ are $N$-dimensional vectors, $A$ is $N \times N$ matrix

| Reference | Complexity | Subroutines H | P | L | A | V | Note |
|---|---|---|---|---|---|---|---|
| Harrow, Hassidim, Lloyed 2008 arXiv | $O\left(\frac{\kappa^2 \cdot \log(N)}{\epsilon}\right)$ | ✓ | ✓ | | ✓ | | known as HHL algorithm, quadratic scaling in condition number |
| Ambainis 2012 | $O\left(\frac{\kappa \cdot \log(N)}{\epsilon^3}\right)$ | ✓ | ✓ | | | ✓ | better scaling in condition number & worse scaling in precision |
| Childs, Kothari, Somma 2017 arXiv | $O\left(\kappa \cdot \log(N) \cdot \text{poly}\log(\frac{1}{\epsilon})\right)$ | ✓ | | ✓ | | ✓ | Exp. improvement in precision & still linear in condition number |
| Subaşı, Somma, Orsucci 2018 arXiv | $O\left(\frac{\kappa \cdot \log(N)}{\epsilon}\right)$ | ✓ | | | | | simple "adiabatic-inspired" randomized algorithm |
| An, Li 2019 | $O\left(\frac{\kappa \cdot \log(N)}{\epsilon}\right)$ | ✓ | | | | | adiabatic algorithm |

**Legend:** **H:** Hamiltonian Simulation **P:** Phase Estimation **A:** Amplitude Amplification
**V:** Variable-Time Amplitude Amplification **L:** Linear Combination of Unitaries

# What else? - Linear Regression


- Problem to be solved: $min_{w \in R^{d+1}} E(w)$
  - $E(w) = ||X \cdot w - Y||^2$ Euclidean error function
  - $X \in \mathbb{R}^{N \times (d+1)}$, i.e.
    - $X$ contains $N$ data points ($N \in \mathbb{N}$) along its rows, and
    - each data point is a $d$ dimensional row vector ($d \in \mathbb{N}$, #features), augmented by unity, having a total length of $d+1$
  - $Y$: Regression labels ($Y \in \mathbb{R}^N$), i.e. the dependant variable in linear regression
  - $w$: Regression weights to be learned, $w \in \mathbb{R}^{d+1}$
- Best classical algorithm $O(N \cdot d^{1.37})$ time, where $N$ is the size of the training data (using a fast matrix multiplication algorithm, such as Coppersmith–Winograd)
- Quantum linear regression algorithm [W'17]: $O\left(poly\left(log(N), d, \kappa, \frac{1}{\epsilon}\right)\right)$
- Linear regression can be formulated as QUBO problem [DP'21] (to be run on quantum annealer)

# What else? - Quantum Counting

- quantum algorithm for efficiently counting the number of solutions for a given search problem
  - Consider $f : \{0,1\}^n \rightarrow \{0,1\}$ to be the oracle function returning $1$ if the input is a solution
  - Calculate the number of solutions $M = \left|f^{-1}(1)\right|$
- based on
  - the quantum phase estimation algorithm,
  - quantum Fourier transform and
  - on Grover's search algorithm
- **Runtim Quantum Solution:** $O\left(\sqrt{\frac{N}{M}}\right)$, where $N = 2^n$
- **Runtime Classical Solution:** $\Omega(N)$
  - without any prior knowledge of the structure of the function $f$

# What else? - Quantum Random Walk

- (Classical) random walk [P1905] is a random process that describes a path that consists of a succession of random steps on some mathematical space
  - Example: random walk on the integer number line $\mathbb{Z}$ which starts at $0$, and at each step moves $\pm 1$ with equal probability
- Quantum walks: random process is achieved by
  1. quantum superposition of states,
  2. non-random, reversible unitary evolution and
  3. state measurements
- Speedups of quantum walks over any classical algorithm
  - Exponential for specific problems [C+'03] [C+'07]
  - Polynomial for many practical problems like
    - the element distinctness problem [A'07],
    - the triangle finding problem [MSS'05],
    - and evaluating NAND trees [FGG'07].
    - Grover's search can also be regarded as some kind of quantum walk algorithm

# What else? - Quantum Artificial Life

- Algorithm for simulating life processes according to Darwinian Evolution including
  - self replications of individuals
  - interactions between individuals
  - mutations and
  - death of individuals
- Realization [R+'16] [R+'18]:
  - Realization of individuals by two qubits for their genotype and phenotype
    - Setup: the state of the genotype is instantiated by some rotation of an ancillary state $|0\rangle\langle 0|$
  - Copying the genotype for transmission of genetic information through generations
  - Phenotype is dependent on the genotype as well as the individual's interactions with their environment
  - Individuals move throughout a two-dimensional spatial grid and occupy cells randomly
    - two or more individuals occupy the same cell $\Rightarrow$ they interact with each other

# What else? - Quantum approximate optimization algorithm (QAOA)

- **To be maximized:** $C(z) = \sum_{\alpha=1}^{m} C_\alpha(z)$
  - $z = z_1 z_2 \ldots z_n$ is the bit string and
  - $C_\alpha(z) = 1$ if $z$ satisfies clause $\alpha$ and $0$ otherwise
    - Typically: $C_\alpha$ depends on only a few of the $n$ bits
    - **Satisfiability:** Is there a string that satisfies every clause?
    - **MaxSat:** Determine string that maximizes $C(z)$
    - **Approximate optimization:** Determine string $z$ for which $C(z)$ is close to the maximum of $C$
- Mapping from quadratic binary optimization (QUBO) - problems to QAOA
  - $\Rightarrow$ In practice universal quantum computers can directly process problems proposed for quantum annealing

# What else?

- Clustering
  - Quantum variant of k-means: q-means [K+'18]
  - Support Vector Clustering (SVC) [B-H+'02], Quantum Clustering (QC) [HG'01], Approximate Quantum Clustering (AQC) [S'13], Dynamic Quantum Clustering (DQC) [WH'09]
- Nearest Neighbor Search (e.g. [WKS'14], [L+'21])
- ...

# Summary & Conclusions

- Simon's algorithm
- Bernstein–Vazirani algorithm
- Quantum Fourier Transform
- Hidden subgroup problem
- Phase Estimation
- Quantum Linear Systems Problem (QLSP)
- Quantum Linear Regression
- Quantum Counting
- Quantum Random Walk
- Quantum Artificial Life
- Quantum approximate optimization algorithm (QAOA)
- Clustering
- Nearest Neighbor Search