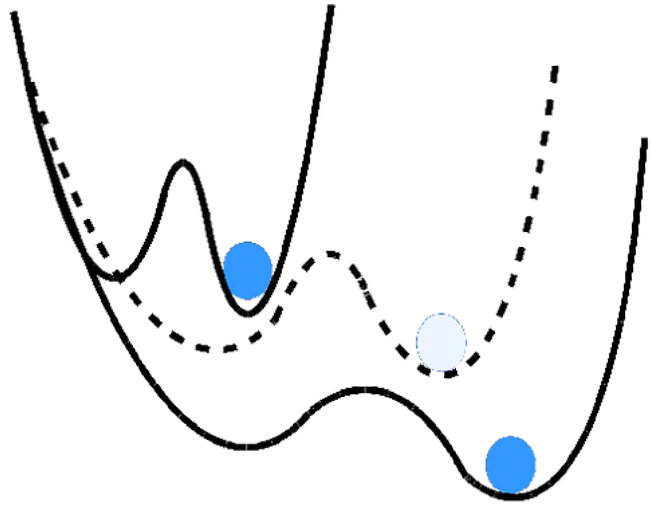
# QUANTUM ALGORITHMS 3:

## OPTIMIZATION



**Simon Apers**
(CNRS & IRIF, Paris)

McKinsey, Paris, May '23
(simonapers.github.io/mckinsey.html)

## TUTORIAL 1: BASICS (21/4)

quantum circuits

quantum Fourier transform

Grover search

## TUTORIAL 2: CHEMISTRY (28/4)

quantum problems

quantum simulation

variational quantum algorithms

## **TUTORIAL 3: OPTIMIZATION** (26/5)

optimization problems

adiabatic algorithm

theory outlook

# Optimization problems

## Adiabatic algorithm

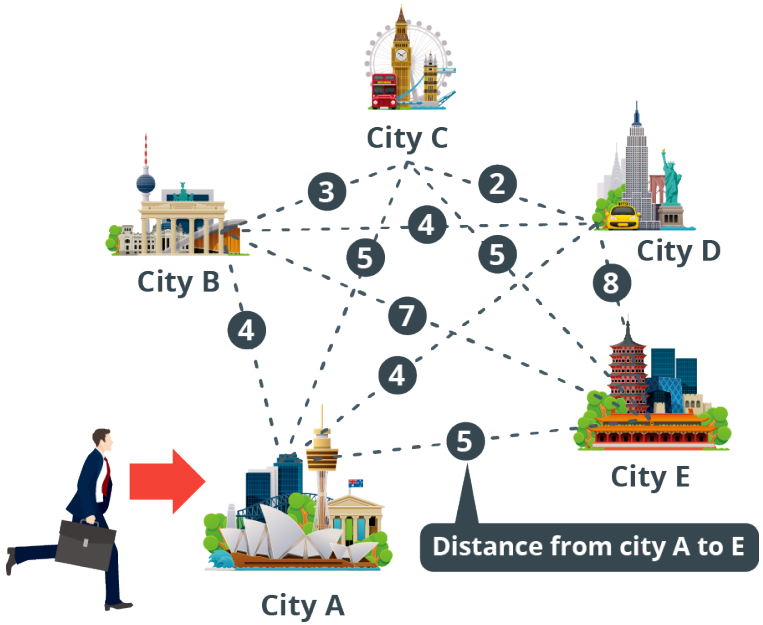
## Theory outlook

## Optimization problems

$$\min_{x \in K} f(x)$$

ubiquitous in computer science, engineering,
operations research, economics, ...

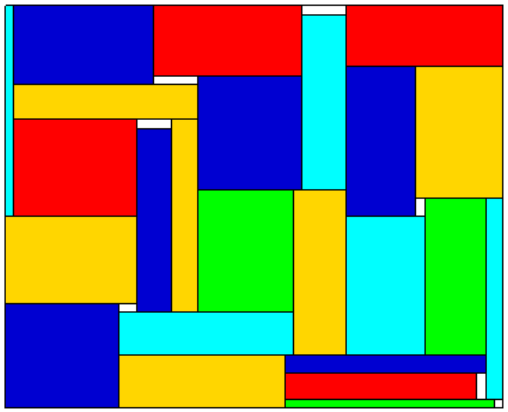
# Example 1: traveling salesperson



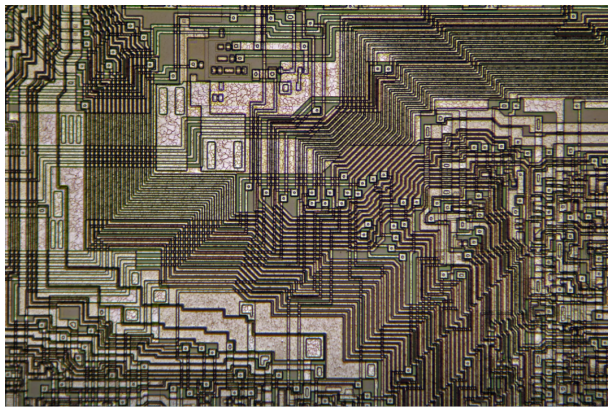

"discrete" optimization problem

solution: tour $[A, D, C, B, E]$

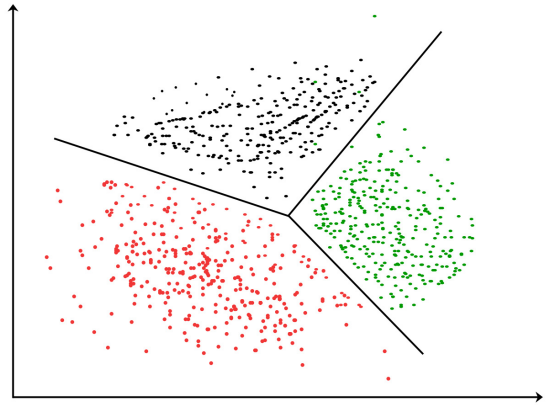
# other discrete optimization problems:



packing



chip design



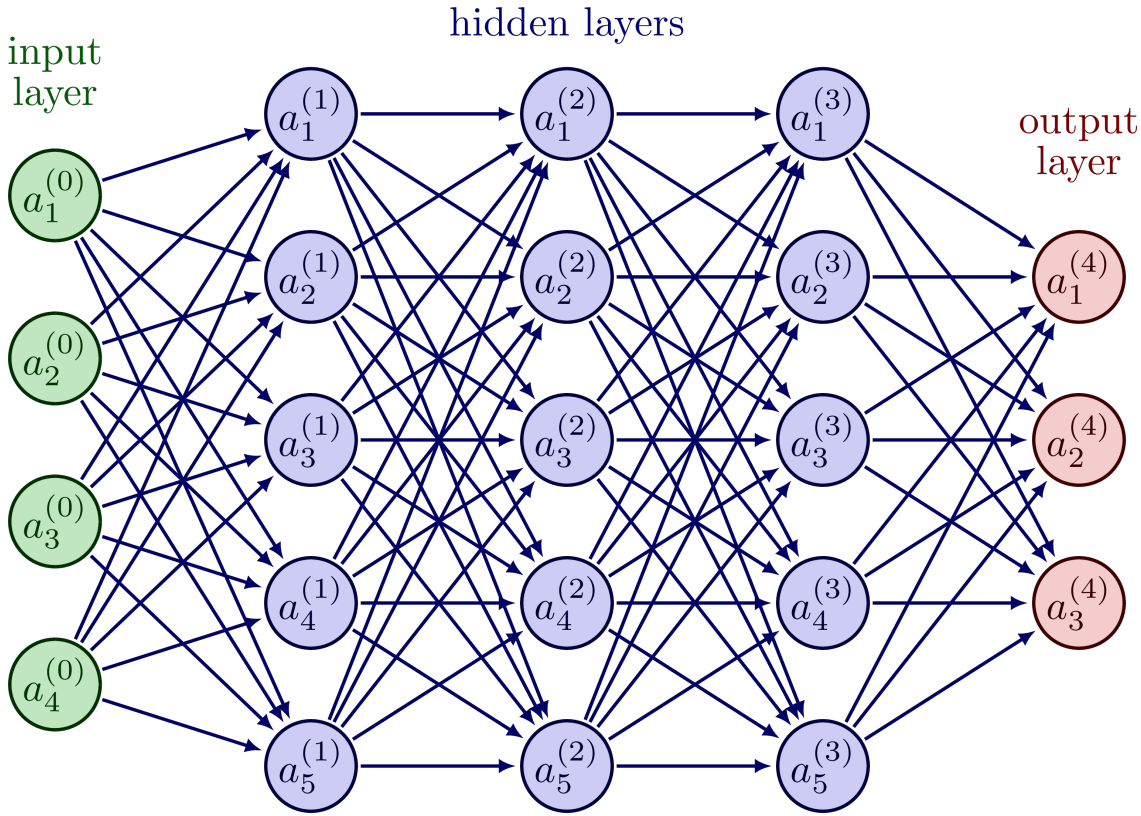
clustering

common heuristics:

exhaustive search – greedy algorithms – local search

simulated annealing – evolutionary/genetic algorithms

# Example 2: neural network training

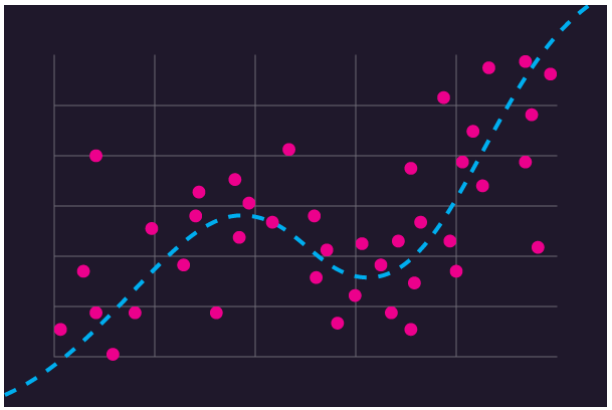



"continuous" optimization problem

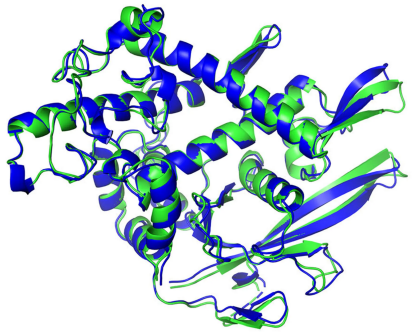
solution: weight vector $w = [0.129, 0.948, 0.474, \dots]$

other continuous optimization problems:

portfolio optimization



regression



protein folding

common algorithms:
gradient descent – interior point methods
Newton's method – local search (Nelder-Mead)

## **Optimization problems:** $\min_x f(x)$ (formalized)

discrete setting:

$$x \in \{0,1\}^n$$

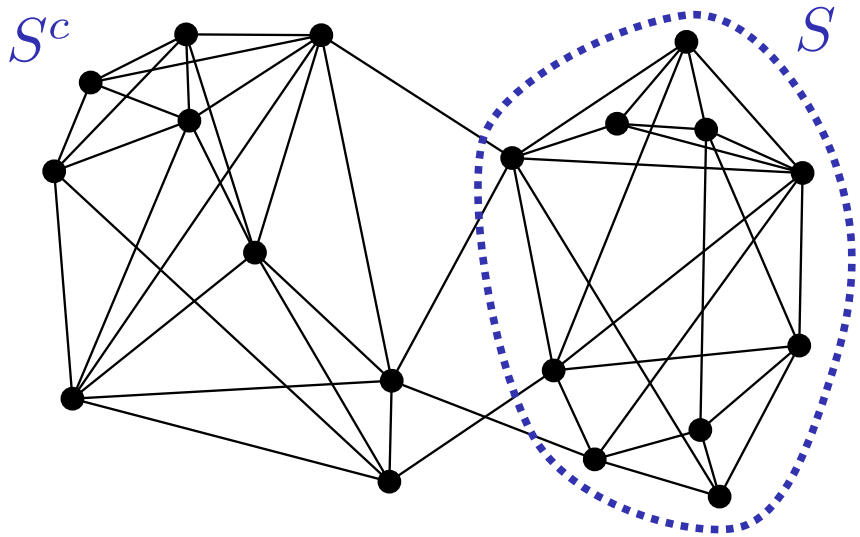
efficient algorithm:
polynomial runtime $\sim \mathrm{poly}(n)$

however, exhaustive search:
exponential runtime $\sim 2^n$

## MINIMUM CUT:

given graph $G = (V, E)$, find subset $S \subset V$ that minimizes cut

$$|E(S, S^c)| = |\{(i,j) \in E \mid i \in S, j \in S^c\}|$$




equivalently: find $x \in \{0,1\}^n$ that minimizes

$$f(x) = \sum_{(i,j) \in E} x_i(1 - x_j)$$

**MINIMUM CUT:**

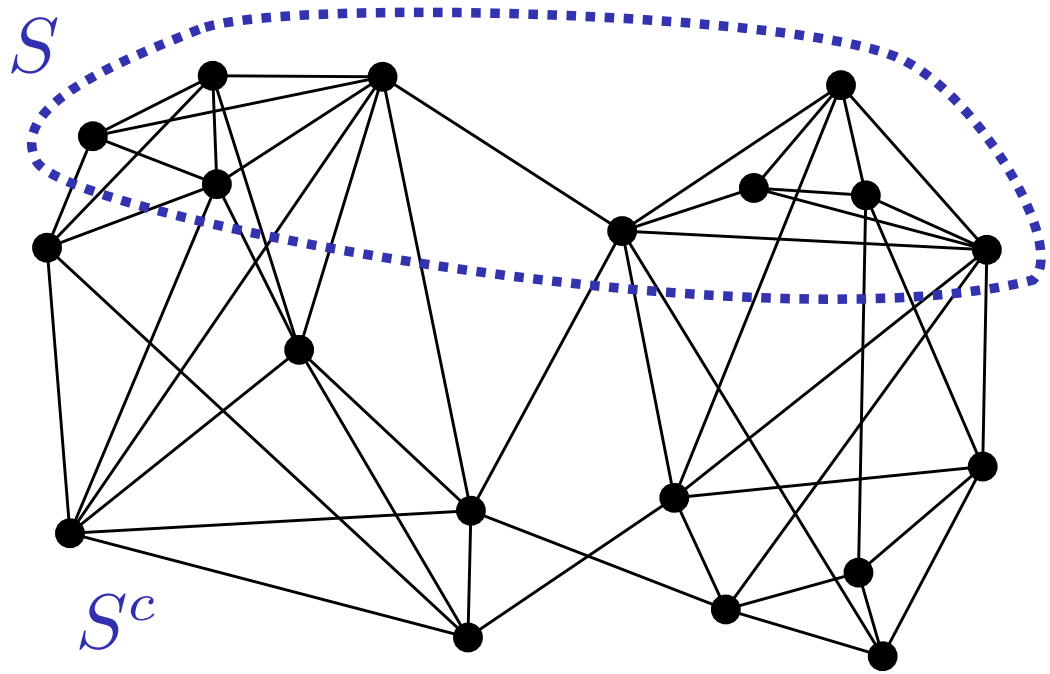
60's: can be solved in time $\mathrm{poly}(n)$!
(e.g., Ford-Fulkerson)

in class **P**
= (decision) problems that can be solved
by a classical computer in polynomial time

## MAXIMUM CUT:

given graph $G = (V, E)$, find subset $S \subset V$ that maximizes cut

$$|E(S, S^c)| = |\{(i,j) \in E \mid i \in S, j \in S^c\}|$$




equivalently: find $x \in \{0,1\}^n$ that minimizes

$$f(x) = -\sum_{(i,j)\in E} x_i(1 - x_j)$$

## MAXIMUM CUT:

70's: NP-complete!
(even to approximate within factor $16/17$)

class **NP**
= (decision) problems that can be *verified*
by a classical computer in polynomial time

problem is NP-complete
if all problems in NP can be "reduced" to it

**P** (efficiently *solvable*)

$\cap$

**NP** (efficiently *verifiable*)

? **P = NP** ?

find efficient algorithm for max cut (or TSP or . . . ),
or prove none exists

millennium prize problem = $1M

common belief: **P** $\neq$ **NP**

however, in practice we *can* solve hard problems!
(TSPs are solved, NNs are trained)

average cases are easier? reductions are “unnatural”?

two faces of optimization:
complex, theoretical algorithms solve worst case problems
$\leftrightarrow$ simple, practical heuristics solve common instances

see e.g. the “unreasonable effectiveness”
of gradient descent in training NNs

## Quantum algorithms for optimization

advantage less “native” than in chemistry

solution $x \in \{0,1\}^n$ described with $n$ bits

distinguish:

near term (non-universal, noisy)
$\leftrightarrow$ long term (universal, error-corrected)

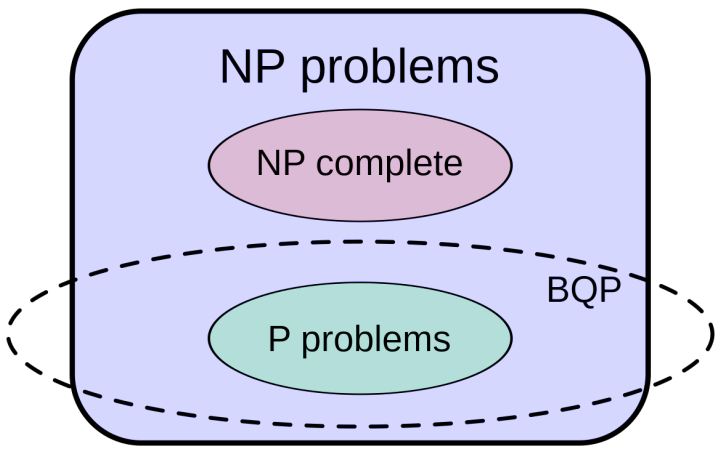
analog (e.g., adiabatic, annealing)
$\leftrightarrow$ digital (gate-based)

# Quantum algorithms for optimization: **complexity classes**

class **BQP**
= (decision) problems that can be solved
by a quantum computer in polynomial time

quantum computers generalize classical computers
$\rightarrow$ **P** $\subseteq$ **BQP**

common belief:




**!** quantum computers not expected to solve NP-complete problems
(such as TSP or training NNs)

## Quantum algorithms for optimization: **theory vs practice?**

### theory:

provable polynomial speedups,
but no “killer applications” with exponential speedups so far

### practice:

similar to classical heuristics,
quantum heuristics for hard problems might work well in practice

heuristics better fitted to near-term devices,
first demonstrations but not convincing yet

base case: **Grover's algorithm**

quadratic speedup over exhaustive search

e.g., $n$-variable SAT formula

$$f(x) = (x_1 \vee \bar{x}_4) \wedge x_3 \wedge (\bar{x}_3 \vee x_1 \vee x_7 \vee \bar{x}_5)$$

? $\exists$ x such that $f(x) = 1$ ?

= NP-complete problem

exhaustive search: time $2^n$

Grover search: time $2^{n/2}$

for general SAT:
$2^n$ classically and $2^{n/2}$ quantumly conjectured optimal
"(quantum) strong exponential time hypothesis"

for general NP problems:
often faster ($\ll 2^n$) classical algorithms
(e.g., branch-and-bound for vertex cover in $2^{0.35n}$)

$\rightarrow$ even quadratic quantum speedup not guaranteed

OPTIMIZATION PROBLEMS

**ADIABATIC ALGORITHM**

THEORY OUTLOOK

# optimization vs ground states

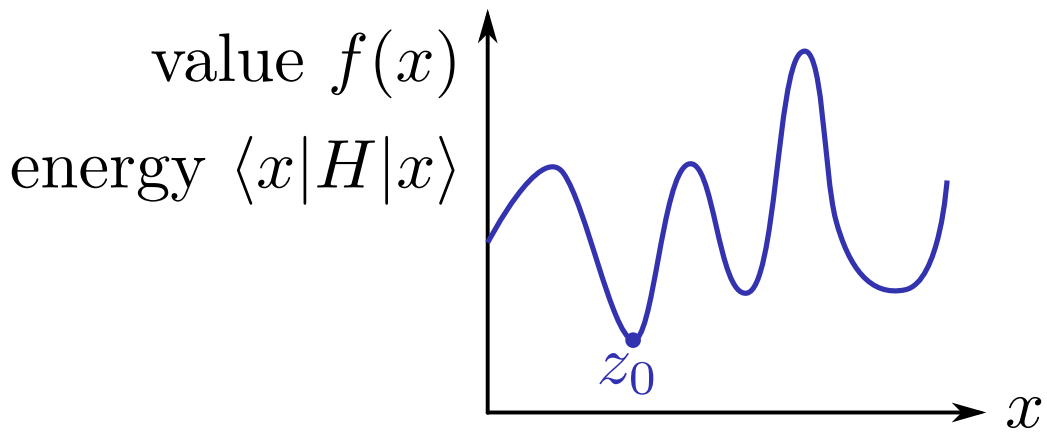
optimization problem

$$\min_{x \in \{0,1\}^n} f(x)$$

$\rightarrow$ find ground state energy of Hamiltonian

$$H = \sum_z f(z) \, |z\rangle \, \langle z|$$

such that $H \, |x\rangle = f(x) \, |x\rangle$



## optimization vs ground states: **QUBO**

quadratic unconstrained binary optimization (QUBO) problem:

$$\min_{x\in\{0,1\}^n} f(x), \qquad \text{with} \qquad f(x) = \sum_{i,j} Q_{ij} x_i x_j$$

maps to Ising Hamiltonian

$$H = \sum_z f(z) \left|z\right\rangle \left\langle z\right|$$

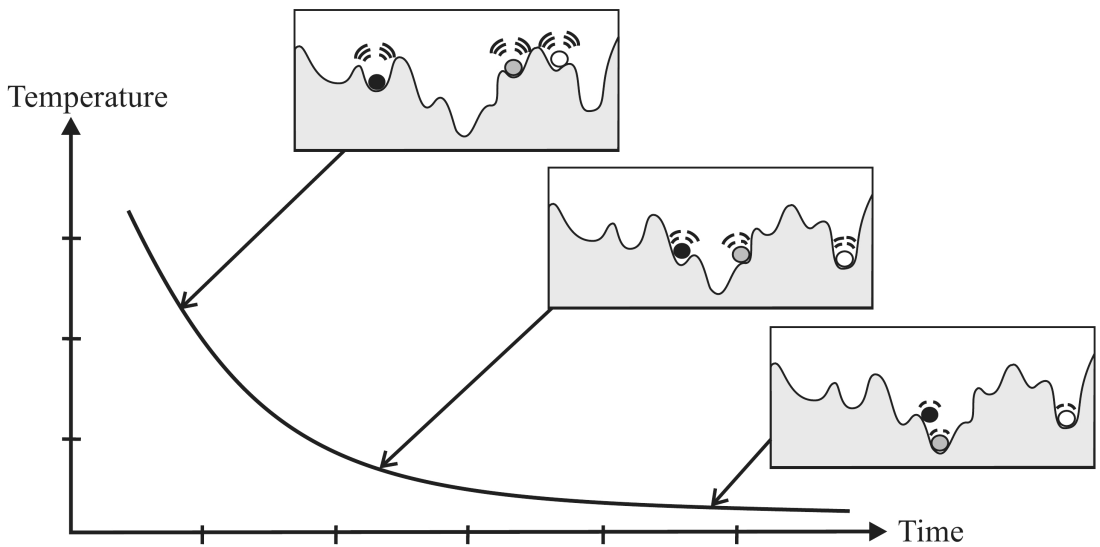
**EX:** use that

$$Z_i \left|x\right\rangle = (-1)^{x_i} \left|x\right\rangle \qquad \text{and} \qquad Z_i Z_j \left|x\right\rangle = (-1)^{x_i + x_j} \left|x\right\rangle$$

to express $H$ in terms of $I$'s, $Z_i$'s and $Z_i Z_j$'s

key heuristic for ground state problems:
adiabatic algorithm

$\sim$ quantum analogue of simulated annealing

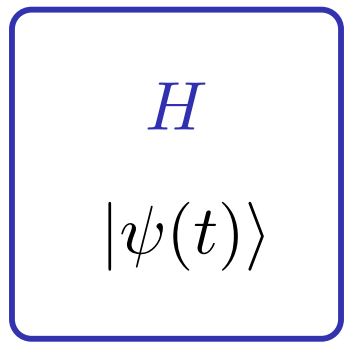


# Adiabatic theorem

system Hamiltonian $H$, initial state $|\psi(0)\rangle$

evolves according to Schrödinger's equation

$$\partial_t |\psi(t)\rangle = -iH |\psi(t)\rangle$$



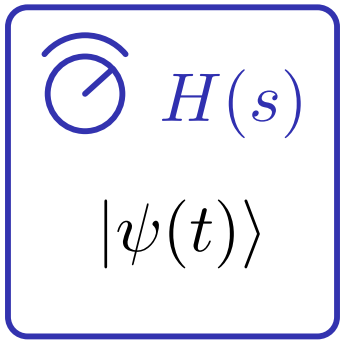

$\rightarrow$ eigenvectors do not change!

(up to global phase)

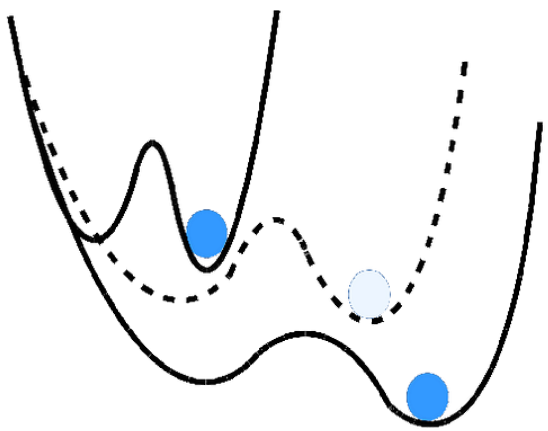
$H |\psi(0)\rangle = \lambda |\psi(0)\rangle$ then

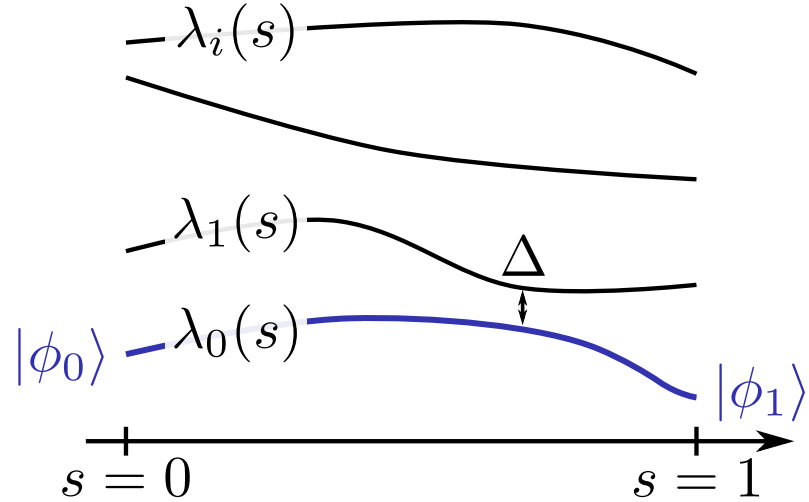
$$|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle = e^{-i\lambda t} |\psi(0)\rangle$$

# what if Hamiltonian (i.e., system) changes?




***Adiabatic theorem:*** *A physical system remains in its instantaneous groundstate if it changes slowly enough and if there is a gap between the groundstate and the rest of the Hamiltonian's spectrum.*

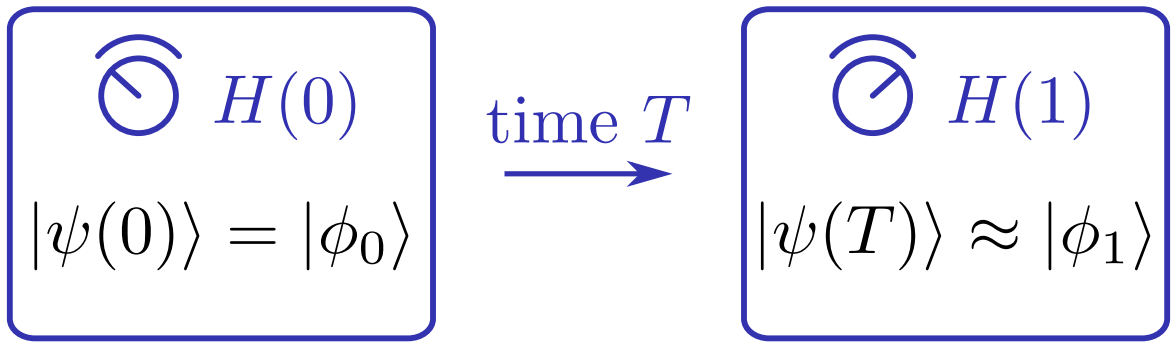




parameterized Hamiltonian $H(s)$:

e.g., Ising Hamiltonian with external field strength $s$

$$H(s) = -\sum_{\langle i,j \rangle} Z_i Z_j + s \sum_i X_i$$




**1.** start from $|\phi_0\rangle$ (g.s. $H_0$)

**2.** slowly change Hamiltonian from $H(0)$ to $H(1)$

(in time $T \gg \mathrm{poly}(1/\Delta)$)

**3.** end in $\approx |\phi_1\rangle$ (g.s. $H_1$)

(i.e., no “jumps”!)

## Adiabatic quantum computation (AQC)

set up:

• initial Hamiltonian $H_0$,
easy to prepare ground state $|\phi_0\rangle$

• final Hamiltonian $H_1$,
"target state" $|\phi_1\rangle$

computation:

evolve $|\phi_0\rangle$ with parameterized Hamiltonian

$$H(s) = (1-s)H_0 + sH_1$$

for $s : 0 \to 1$ in time $T$

## Adiabatic quantum computation (AQC)

= “analog” model of quantum computation,
contrasts with “digital” gate-based model

if all Hamiltonians allowed:
*universal* model
(adiabatic $\rightarrow$ gate: Hamiltonian simulation,
gate $\rightarrow$ adiabatic: “Feynman Hamiltonian”)

if not:
restricted model
(e.g., quantum annealers from D-Wave)

# Adiabatic optimization algorithm

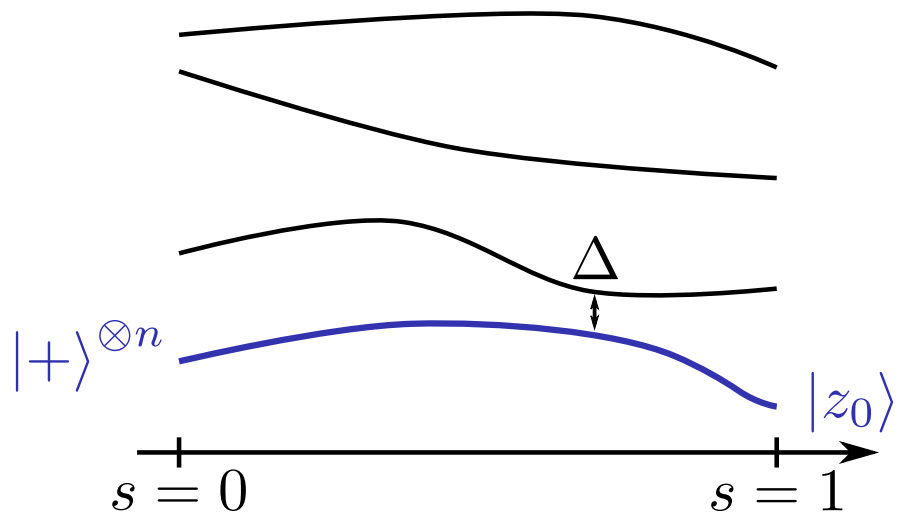
initial Hamiltonian $H_0 = -\sum_i X_i$,
easy to prepare ground state $|\phi_0\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle$

final Hamiltonian $H_1 = \sum_z f(z) |z\rangle \langle z|$,
"target state" $|\phi_1\rangle = |z_0\rangle$, with $z_0$ minimizer $f$

parameterized Hamiltonian

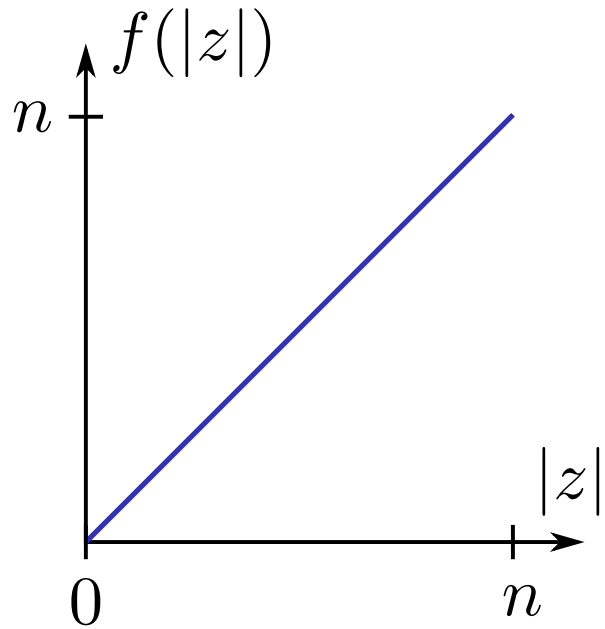
$$H(s) = s \sum_z f(z) |z\rangle \langle z| - (1-s) \sum_i X_i$$

adiabatically "turns on" magnetic interactions



## **Toy example:** Hamming weight

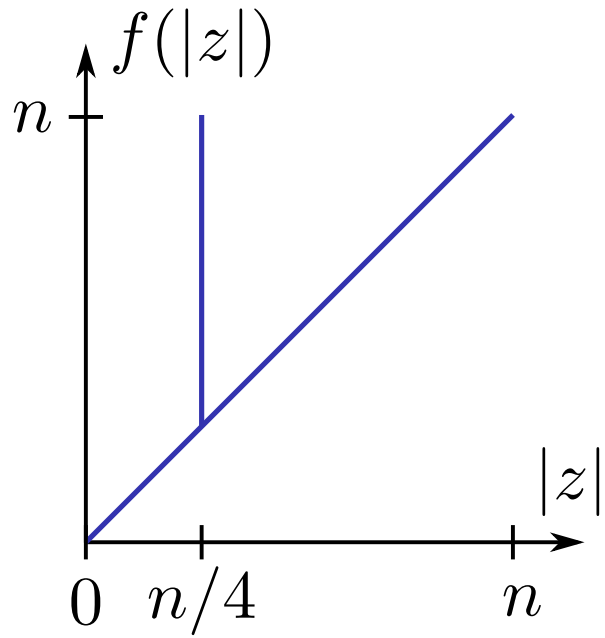
$$f(z) = |z|$$




quantum adiabatic algorithm finds $z_0$ in $\mathrm{poly}(n)$ time
classical greedy/annealing algorithm finds $z_0$ in $\mathrm{poly}(n)$ time

## **Toy example:** Hamming weight with a spike

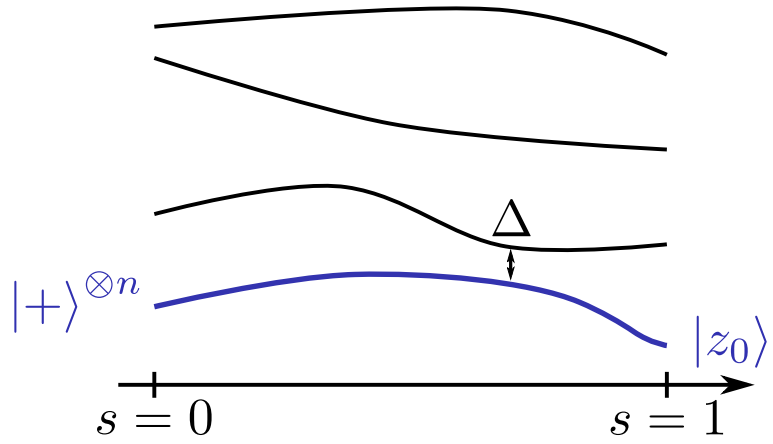
$$f(z) = \begin{cases} |z| & |z| \neq n/4 \\ n & |z| = n/4 \end{cases}$$




quantum adiabatic algorithm finds $z_0$ in $\mathrm{poly}(n)$ time
classical greedy/annealing algorithm needs $\exp(n)$ time

"quantum tunneling" from local minimum

# Adiabatic optimization algorithm




? generic scaling $\Delta$ ?

(recall, runtime $\sim \mathrm{poly}(1/\Delta)$)

random instances of NP-complete problems,
variants of Hamming weight with spike:

$$\Delta \sim 1/2^n$$

## **QAOA:** quantum approximate optimization algorithm

inspired by circuit model implementation
of time-$T$ adiabatic algorithm:

$$\partial_t |\psi(t)\rangle = -iH(t/T) |\psi(t)\rangle$$

approximation 1:
for $T \gg 1$ we get

$$|\psi(t+1)\rangle \approx e^{-iH(t/T)} |\psi(t)\rangle$$

and so

$$|\psi(T)\rangle \approx e^{-iH(1)} e^{-iH(1-1/T)} \ldots e^{-iH(1/T)} e^{-iH(0)} |\psi(0)\rangle$$

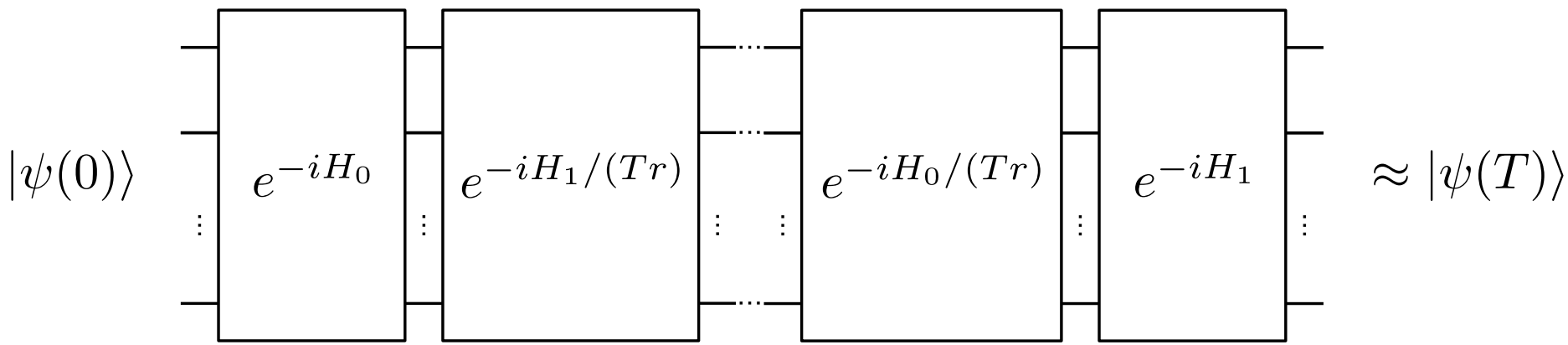
## approximation 2:

by Lie-Trotter formula on $H(s) = (1-s)H_0 + sH_1$, for $r \gg 1$:

$$e^{-iH(s)} \approx \left(e^{-i(1-s)H_0/r} e^{-isH_1/r}\right)^r$$

## combined:

$$|\psi(T)\rangle \approx e^{-iH_1} e^{-iH_0/(Tr)} e^{-iH_1(1-1/T)/r} \ldots e^{-iH_0(1-1/T)/r} e^{-iH_1/(Tr)} e^{-iH_0} |\psi(0)\rangle$$




= product of $e^{-iH_1\delta}$'s and $e^{-iH_0\delta}$'s !

# Quantum approximate optimization algorithm **(QAOA)**

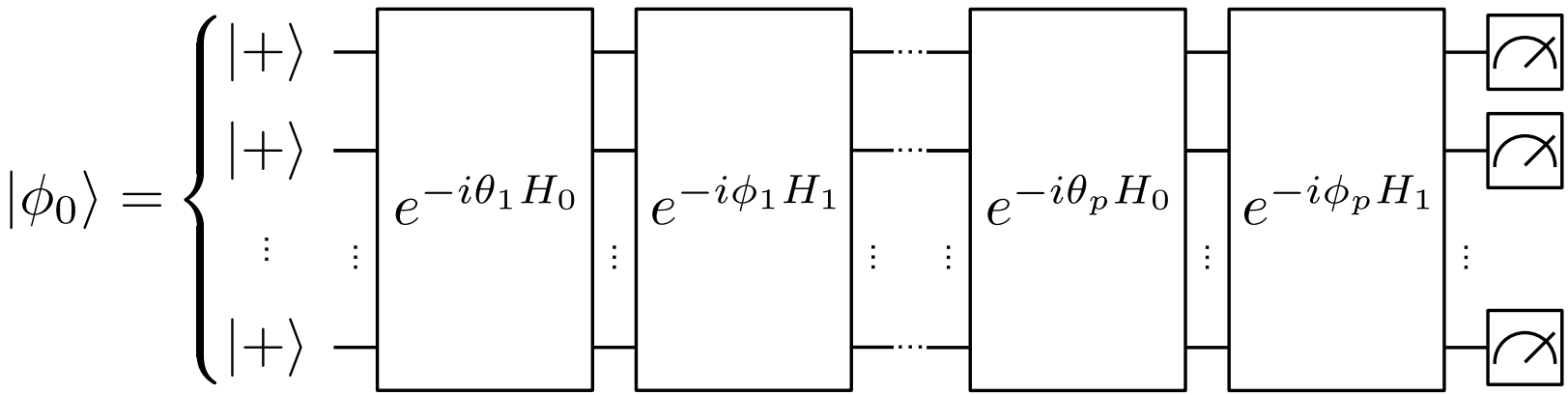
variational circuit based on mixer Hamiltonian

$$e^{-i\phi H_1} = e^{-i\phi \sum_i X_i}$$

and cost Hamiltonian

$$e^{-i\theta H_0} = e^{-i\theta \sum f(z)|z\rangle\langle z|}$$

$\rightarrow$ depth-$p$ QAOA circuit with parameters $\{\theta_1, \phi_1, \ldots, \theta_p, \phi_p\}$:




correctness adiabatic algorithm $\Rightarrow$ correctness QAOA (for $p \rightarrow \infty$)

# Optimization problems

# Adiabatic algorithm

# Theory outlook

## Theory outlook: **Grover**

base case: quadratic speedup over exhaustive search

e.g., $n$-variable SAT formula in time $2^{n/2}$
(versus time $2^n$ classically)

conjectured best possible for *general* SAT

## Theory outlook: **Grover**

3-SAT ($\leq 3$ variables per clause)

$$(x_1 \vee \bar{x}_4) \wedge x_3 \wedge (\bar{x}_3 \vee x_1 \vee x_7)$$

solved by Schöning's algorithm in time $2^{0.415n} \ll 2^{n/2}$!

uses local search subroutine

$\rightarrow$ local search + Grover = time $\sqrt{2^{0.415n}} = 2^{0.207n}$

similar situation for **TSP** (but more work and smaller speedup):

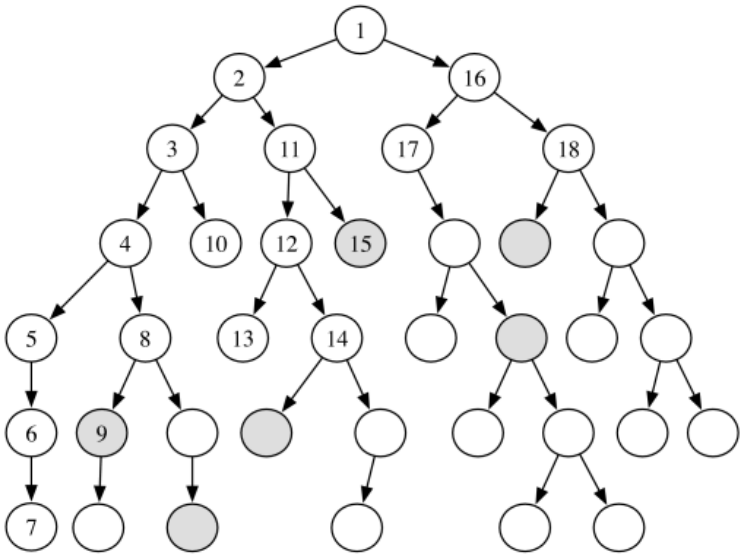
– exhaustive search: $n^n$

– dynamic programming: $2^n$

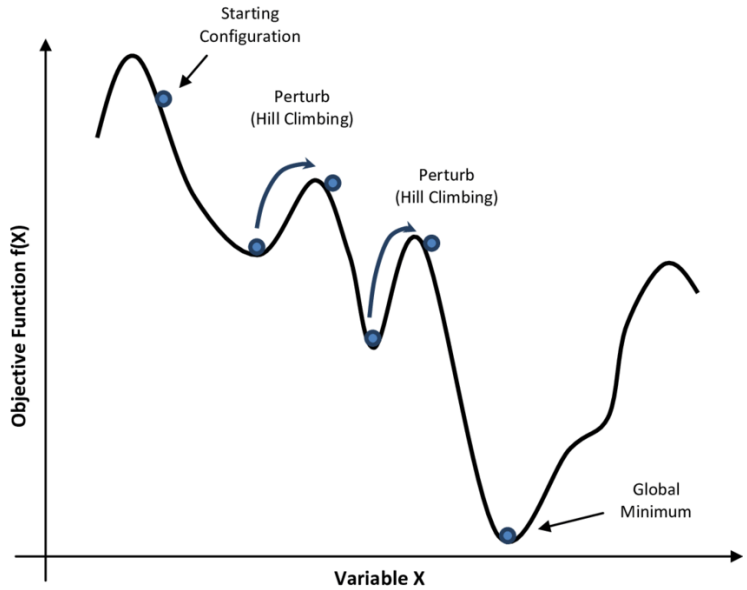
– dynamic programming + Grover: $2^{0.79n}$

Grover might not always give speedup!

e.g., unclear how to combine with other heuristics for fast SAT solving:



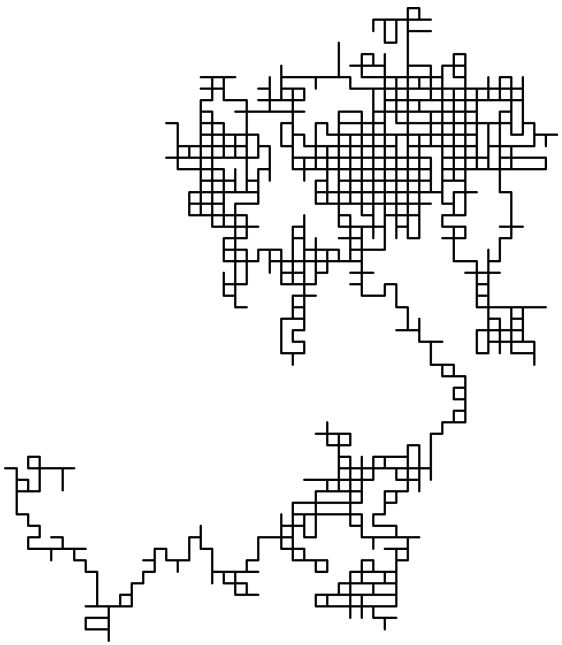

backtracking, branch-and-bound



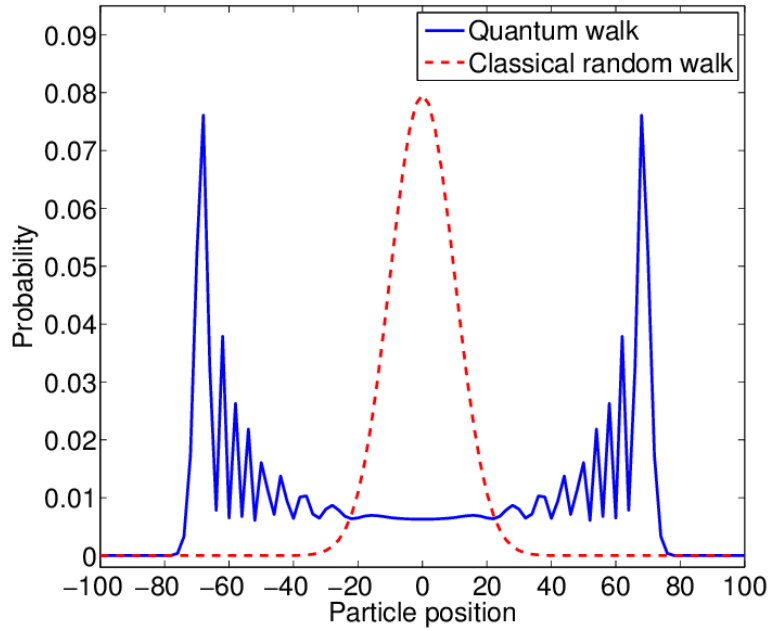

simulated annealing

# Theory outlook: **quantum walks**

= quantum version of random walks



RW on grid




RW vs QW on line

local exploration of state space / graphs,
can give quadratically faster “hitting time”

polynomial quantum walk speedups for algorithms based on backtracking, branch-and-bound, simulated annealing

# Theory outlook: **HHL**

linear equation:

$$ax = b$$

$\rightarrow$ solution $x = b/a$

$N$-dimensional linear system:

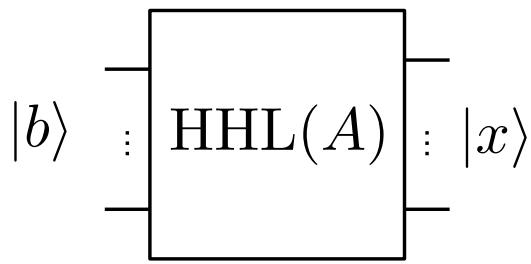
$$\begin{bmatrix} & A & \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}$$

$\rightarrow$ solution $x = A^{-1}b$

computing $x$ is bottleneck in
engineering, machine learning, economics, computer graphics, ...

## Theory outlook: **HHL**

Harrow-Hassidim-Lloyd '08:
quantum algorithm for linear system solving,
returns *quantum solution* $|x\rangle = \sum_{i=1}^{N} x_i |i\rangle$




complexity: $\mathrm{poly}(\mathrm{cond}(A), \log n)$
vs. $\mathrm{poly}(n)$ of naïve classical algorithms

# Theory outlook: **HHL**

(rough) idea:

interpret $A$ as Hamiltonian,
use Hamiltonian simulation $e^{-iAt}$ to map

$$|b\rangle \to e^{-iAt}|b\rangle = \sum_{k=0}^{\infty} \frac{1}{k!}(-iAt)^k |b\rangle$$

$$\to \dots$$

$$\to \sum_{k=0}^{\infty} (I-A)^k |b\rangle = A^{-1}|b\rangle = |x\rangle$$

caveats:

- output is *quantum state*
  - QRAM issues
  - dequantization

## Theory outlook: **other**

from HHL and Hamiltonian simulation:
quantum algorithms for LPs and SDPs, interior point methods,
"quantum linear algebra" for machine learning

from quantum query complexity:
quantum oracle speedups for gradient estimation,
convex optimization

for more:
see quantumalgorithmzoo.org

# SUMMARY:

## OPTIMIZATION PROBLEMS

quantum (probably) cannot solve NP-complete problems
Grover: quadratic baseline

## ADIABATIC ALGORITHM

main heuristic, quantum version of simulated annealing
viable in near-term
inspiration for QAOA

## THEORY OUTLOOK

exist provable quantum speedups
often polynomial
many caveats, may be impractical

## Figure references

traveling salesperson: https://annealing-cloud.com/en/knowledge/1.html

chip: https://www.wired.com/story/fit-billions-transistors-chip-let-ai-do/

clustering: https://www.geeksforgeeks.org/clustering-in-machine-learning/

neural network: https://tikz.net/neural_networks/

annealing 1: http://fri.oden.utexas.edu/fri/Labs_2019/lab5/part2.php

adiabatic: https://medium.com/@quantum_wa/quantum-annealing-cdb129e96601

backtracking: https://www.javatpoint.com/backtracking-introduction

branch-and-bound: https://artint.info/2e/html/ArtInt2e.Ch3.S8.SS1.html

annealing 2: https://medium.com/analytics-vidhya/simulated-annealing-869e171e763c

quantum walk: https://www.researchgate.net/publication/45898194_Discrete-Time_Quantum_Walk_-_Dynamics_and_Applications