# Variational Quantum Algorithms

Adrián Pérez Salinas

[1]Lorentz Institute - Leiden University
[2]$\langle aQa \rangle^L$: Applied Quantum Algorithms

Spring school in near-term quantum computing
— Benasque 2024 —

# Contents

Section 1

# Introduction

# Introduction

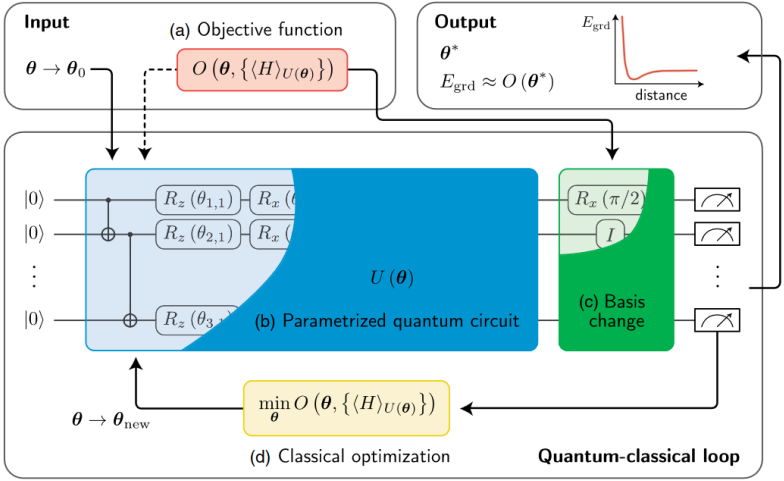
- Nowadays quantum computers have hardware limitations
  - Qubit size
  - Available depth
  - Noise
- Variational algorithms were conceived as a way to overcome such limitations by
  - Simplifying problems
  - Reaching solutions quickly
  - Mitigate errors through optimization

Notice $C(\theta, x) \geq E_0$, the ground state of $H$

**How they work**

1. We create a parameterized quantum circuit (PQC) $U(\theta)$
2. Such ansatz generate an output state $|\psi(\theta, x)\rangle = U(\theta)\,|x\rangle$.
3. We encode a problem of interest in a hamiltonian $H$
4. The cost function is defined $C(\theta, x) = \langle\psi(\theta, x)|\, H\, |\psi(\theta, x)\rangle$
5. We optimize *classically* the parameters $\theta$ to minimize $C(\theta, x)$

# Introduction



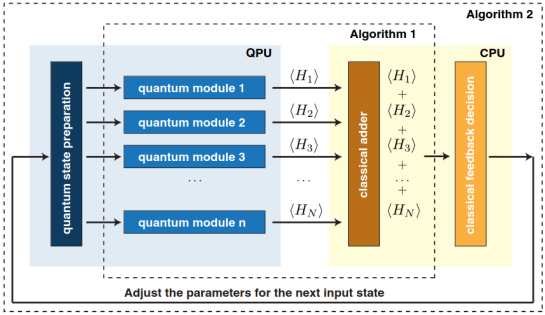

Bharti et al. 2022

# Variational Quantum Eigensolver (VQE)

- The VQE was the first work in this domain[a]
- Originally implemented in an experiment, to make it cheaper

$$H = \sum_{i,\alpha} h^i_\alpha \sigma^i_\alpha + \sum_{i,j,\alpha,\beta} h^{ij}_{\alpha\beta} \sigma^i_\alpha \sigma^j_\beta$$

[a] Peruzzo et al. 2014.




- The Hamiltonian is decomposed in measurable Pauli strings
- Only doable if the Hamiltonian has a few Pauli strings

# Quantum Approximate Optimization Algorithm (QAOA)

The QAOA is described in a simple way[a]

- Initial Hamiltonian: $H_I$
- Final Hamiltonian: $H_F$

The output state is given by

$$\left|\psi(\vec{\beta},\vec{\gamma})\right\rangle = \prod_{j=1}^{L} e^{-i\beta_j H_I} e^{-i\gamma_j H_F}$$

[a] Farhi, Goldstone, and Gutmann 2014.

Originally conceived as a quantum algorithm to solve combinatorial problems encoded in

$$H_F = \sum_i a_i Z_i + \sum_{i,j} a_{ij} Z_i Z_j$$

QAOA is closely related to the adiabatic evolution via Trotter simulation

$$H(t) = (1-t)H_I + tH_F$$

starting from the ground state of $H_I$.
$|\psi(t)\rangle \approx \prod_j e^{-i\Delta t(1-t_j)H_I} e^{-i\Delta t t_j H_F} |E_{I,0}\rangle$
The parameters $\vec{\beta},\vec{\gamma}$ aim to shortcut the adiabatic evolution.

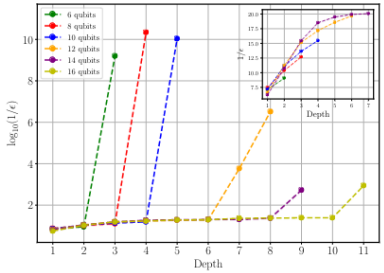
Section 2

# Ansatzes

# Ansatzes

Ansatzes are just the circuit structure, without specifying the parameters

- Ansatzes give certain properties to the quantum circuit
- Some ansatzes are capable of reaching any unitary, some others are restricted
- With sufficient depth, it is possible to reach (in principle) any state
- Ansatzes are responsible for connectivity and correlation
- Ansatzes determine whether a quantum circuit is simulable or not



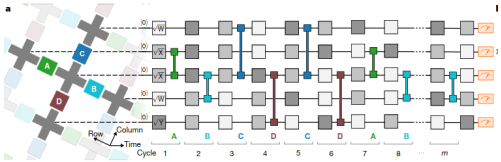

In this graph, the error in obtaining the ground energy decreases when long-range correlations are available[a]

[a] Bravo-Prieto et al. 2020.

# Hardware Efficient Ansatz (HEA)

**Hardware efficient**

- Quantum circuits must be run in some hardware
- The hardware is composed by several qubits in some grid
- Only certain connectivity is allowed




Arute et al. 2019

- This ansatz is directly mapped to quantum hardware
- Overhead when simulating physics away from the chip (e. g. long range correlations)
- If the chip is 1D, any shallow HEA is doable with tensor networks[a]
- Useful if the chip is close to the problem of interest, not for general purpose

[a] Attend Mari Carmen's talk

# Hamiltonian variational ansatz

Hamiltonian-inspired is similar to hardware inspired, but linking it to the problem

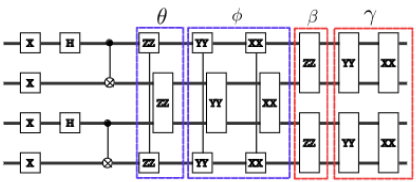
$$H = \sum_i h_i,$$

then

$$U(\theta) = \prod_{j=1}^{L} e^{-ig_j\theta_j},$$

with $g_j = h_i$, for some $(i, j)$.
Notice QAOA is a particular instance of the Hamiltonian-inspired ansatz

- This ansatz immediately captures the properties of the problem of interest
- It is an informed guess to solve the problem
- If the chip is not prepared, implementing this ansatz might bring large overhead




$$H = \sum_i X_i X_{i+1} + Y_i Y_{i+1} + \Delta Z_i Z_{i+1}$$

# Unitary coupled-cluster (UCC) ansatz

- Unitary Coupled Cluster is a family of ansatzes specifically designed for fermionic problems
- Widely used in chemistry
- UCC includes physical information into the quantum circuit: particle number

The ansatz is generated by

$$|\psi\rangle = e^S |HF\rangle ,$$

with $|HF\rangle$ being the Hartree-Fock[a] state

[a] single-occupation state with lowest energy

Hamiltonian is composed by creation/annihilation operators $a_p^\dagger, a_q$

- indices $p, q$ are occupied orbitals
- indices $i, j$ are virtual (free) orbitals
- $S = T - T^\dagger$

$T = \sum_k T_k$

$T_k(\theta) = \frac{1}{(k!)^2} \sum_{ij\ldots}^{\text{occupied}} \sum_{ab\ldots}^{\text{virtual}} t_{ij\ldots}^{ab\ldots} a_a^\dagger a_b^\dagger \ldots a_i a_j \ldots$

Thus, we utilize an Ansatz with $k$ annihilation and creation operators. The coefficients $t$ come from the problem.

Anand et al. 2022

# Unitary Coupled-Cluster (UCC) ansatz

We implement the UCC ansatz in a quantum computer via Trotterization as

$$e^{T-T^\dagger} = e^{\sum_k \theta_k (T_k - T_k^\dagger)} \approx \left( \prod_k e^{\frac{\theta_k}{t}(T_k - T_k^\dagger)} \right)^t$$

Since this is hard to implement, we can first-order approximate it as

$$|\psi_{UCC}(\theta)\rangle \approx \prod_{\boldsymbol{p},\boldsymbol{q}} e^{\theta_{\boldsymbol{pq}} A_{\boldsymbol{pq}}} |HF\rangle$$

with $A_{\boldsymbol{pq}}$ being a excitation operator $A_{\boldsymbol{pq}} = \prod_{k=1}^{n} a_{p_k}^\dagger a_{q_k}^\dagger - h.c.$

**Fermion-to-qubit mapping**

Quantum computers cannot natively implement fermionic operations, thus we need to apply a fermion-to-qubit mapping (Jordan-Wigner, Bravyi-Kitaev)

**Jordan-Wigner**

$$a_k^\dagger = I^{\otimes k-1} \otimes \sigma_+^k \otimes \sigma_z^{\otimes n-k}$$
$$a_k = I^{\otimes k-1} \otimes \sigma_-^k \otimes \sigma_z^{\otimes n-k}$$
$$2\sigma_\pm^k = \sigma_x \pm i\sigma_y$$

- Fermion-to-qubit $\rightarrow$non-local operations
- Hard to implement in a quantum computer

# Layered ansatzes

Layered ansatzes are of the form

$$U(\theta) = \prod_{l=1}^{L} \prod_{j=1}^{G} e^{-i\theta_{l,j} g_j}$$

- There are $L \times G$ different gates
- The structure is repeated in groups of size $G$
- The parameters $\theta$ are (in principle) chosen independently
- The generators $g$ are traceless Hermitian matrices, and they define the ansatz

**Examples for the generators**

- Pauli strings: $g = \bigotimes_{k=1}^{n} \sigma_k$
- Sum of Pauli strings: $g = \sum_{k=1}^{n} Z_k$

- Layered ansatzes allow for systematic studies of the resulting quantum circuits
- Once compiled, layered ansatzes allow for increasing the depth of the quantum circuit

Section 3

# Optimization methods

# Optimization

A story of getting lost in the mountains...

# Optimization

A story of getting lost in the mountains...

**In a quantum mountain**

the ground is unstable (due to errors and shot noise)

# Optimization

- Optimization is a crucial step in variational algorithms
- Optimization is a very hard problem, except in some trivial cases
- Optimization solves two problems
  - Find the minimum closest to an initial point
  - Find the best minimum of all minima
- Optimization relies only on *local* information
- Optimization iteratively suggests new candidates for the minimum

**Solving an optimization problem**

The only way to solve it with guarantees is through exhaustive search

- Such cost increases exponentially with the number of parameters $m$
- However, for the optimization to be efficient we need cost at most polynomial in $m$
- Thus we need clever ways to explore the optimization landscape

# Optimization

## How costly is to optimize?

- Number of function evaluations
- Cost of choosing the next step? Depends on method
  - Constructing auxiliary quantities
  - Extra evaluations
- How many iterations are needed to convergence? Depends on method
- How many measurements are needed per iteration? Depends on method
- When are we satisfied with a candidate? Termination condition

**Methods**

There exist many different methods to address optimization problem, even in a classical setup

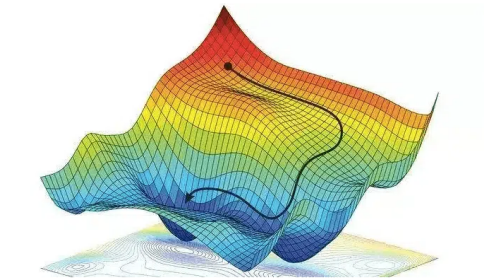
And no method is capable of solving all problems

# Gradient-based optimizers

These optimizers fall under the general rule

$$\theta_{i+1} = \theta_i - \eta_i g(\theta_i); \qquad g(\theta) \approx \frac{\partial C(\theta)}{\partial \theta}$$

The parameters $\theta$ are iteratively updated



All freedom is encoded in the quantities $\eta, g(\theta)$

- $\eta$ can be adaptive in the iteration number (faster convergence)
- $g(\theta)$ is close to the gradient, but may store past information or *momentum* (avoid sticking to local minima or saddle points)
- There exist second-order methods that include the information of the Hessian (choosing better directions)

# Gradient calculation - in quantum

**Numerical estimation**
Most gradient-calculators rely on numerical estimations of the kind

$$\partial_i C(\theta) \approx \frac{C(\theta + \hat{e}_i\epsilon) - C(\theta - \hat{e}_i\epsilon)}{2\epsilon}$$

for $\hat{e}_i$ being the unitary vector in the $i$-th direction. $\epsilon$ is usually very small

Due to shot noise $\tilde{C}(\theta) = C(\theta) + \mathcal{O}(M^{-1/2})$, and thus the derivative is computed with large error $\mathcal{O}(\epsilon^{-1}M^{-1/2})$

Reliable estimations require large $\epsilon$, which is not common

**Exact calculation** We exploit the unitary structure to compute exact gradients

**Parameter shift rule**

Derivative with respect to $\theta$ of $C(\theta) = \langle\psi| e^{ig\theta} H e^{-ig\theta} |\psi\rangle$, where $g$ has eigenvalues $e_0, e_1$, and $r = (e_1 - e_0)/2$. Then

$$\partial_\theta C(\theta) = r\left(C\left(\theta + \frac{\pi}{4r}\right) - C\left(\theta - \frac{\pi}{4r}\right)\right)$$

This strategy can be further generalized to arbitrary generators[a]

[a] Crooks 2019; Wierichs et al. 2022.

# Gradient calculation - in quantum

**Quantum natural gradient**[a]

- The natural gradient aims to perform gradient descent over the information landscape
- The optimization happens in a more sensitive geometric space

$$G(\theta)_{ij} = \langle \partial_i \psi(\theta) | \partial_j \psi(\theta) \rangle - \langle \psi(\theta) | \partial_j \psi(\theta) \rangle \langle \partial_i \psi(\theta) | \psi(\theta) \rangle$$

and we update parameters as

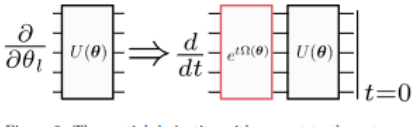
$$\theta_{i+1} = \theta_i - \eta G(\theta)^+ \nabla C(\theta)$$

[a] Stokes et al. 2020.

**Parametrization effect**[a] An arbitrary gates $U \in \mathcal{SU}(N)$ is parameterized as

$$U(\theta) = e^{i \sum_m \theta_m g_m},$$

with $g_m$ being the generators of the $\mathcal{SU}(N)$ group. These generators can be chosen in many different ways. The derivative can be constructed through the adjoint representation of the Lie algebra $\Omega_l(\theta)$




[a] Wiersema et al. 2023.

# Gradient-free optimizers

- Sample in different points and construct new candidates
- Explore larger regions of the parameter space than gradient-based
- Less prone to getting stuck in local minima
- Slower convergence
- More evaluations

## Simplicial methods

Each iteration constructs new samples deterministically from previous step; Nelder and Mead 1965

## Evolutionary algorithms

Each iteration samples from a probability distribution; Hansen 2006

## Coordinate-based algorithms

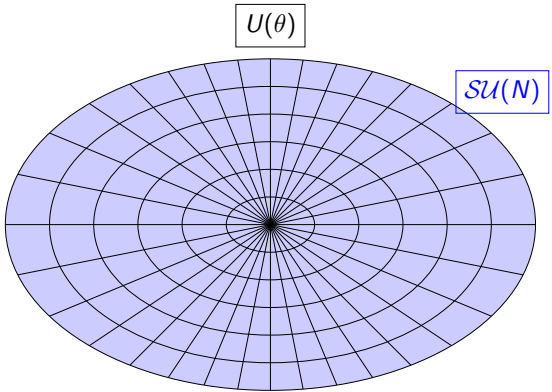
Solves a 1D optimization problem per coordinate, iteratively; Powell 1964

Section 4

# Problems of variational algorithms

# Expressivity

- It is the capability of a parameterized model to approximate a family of solutions
- PQCs are a subset of unitary operations
- does there exist a parameter configuration approximating *any* unitary operation?
  - How accurate is the approximation?
  - How many gates are needed?
  - Can we exploit some structure?



# Expressivity

**Haar measure**

- The Haar measure is a useful tool for expressivity
- Haar measure is just uniform distribution of unitaries over the unitary space

Formally

$$\mu(U) = \mu(VU), \forall V \rightarrow$$
$$\int_{\mathcal{SU}(N)} f(U) d\mu(U) = \int_{\mathcal{SU}(N)} f(VU) d\mu(U)$$

What? This is weird...

**$t$-designs**

- $t$-designs are the usual way to study expressivity
- A $t$ design over a probability distribution matches the $t$-th statistical moment
  - 1-design: mean
  - 2-design: variance

In the unitary group

$$\int_{t-\text{design}} dU \; U^{\otimes t} \otimes (U^\dagger)^{\otimes t} = \int_{\text{Haar}} dU \; U^{\otimes t} \otimes (U^\dagger)^{\otimes t}$$

# Expressivity

**Haar measure**

- The Haar measure is a useful tool for expressivity
- Haar measure is just uniform distribution of unitaries over the unitary space

An easy example, when we integrate over a sphere, we do

$$\int_0^{2\pi} d\phi \int_0^{\pi} \sin\theta d\theta \int_0^{R} r^2 dr \; f(r,\theta,\phi)$$

**$t$-designs**

- $t$-designs are the usual way to study expressivity
- A $t$ design over a probability distribution matches the $t$-th statistical moment
  - 1-design: mean
  - 2-design: variance

In the unitary group

$$\int_{t-\text{design}} dU \; U^{\otimes t} \otimes (U^\dagger)^{\otimes t} = \int_{\text{Haar}} dU \; U^{\otimes t} \otimes (U^\dagger)^{\otimes t}$$

# Expressivity

Usual measures for expressivity

**States**

$$A_\psi(O) = \int_{\boldsymbol{\theta}} d\theta \, \mathrm{Tr}\left(O \left(|\psi(\theta)\rangle \langle\psi(\theta)|\right)^{\otimes t}\right) - \int_{\mathrm{Haar}} d\psi \, \mathrm{Tr}\left(O \left(|\psi\rangle \langle\psi|\right)^{\otimes t}\right)$$

Sim, Johnson, and Aspuru-Guzik 2019

**Unitaries**

$$A_U(O) = \int_{\boldsymbol{\theta}} d\theta U(\theta)^{\otimes t} O (U(\theta)^\dagger)^{\otimes t} - \int_{\mathrm{Haar}} dU \; U^{\otimes t} O (U^\dagger)^{\otimes t}$$

Holmes et al. 2022

**Reduce expressivity**

We can always define Haar measures and $t$-designs with respect to different spaces to artificially increase or decrease the expressivity

# Trainability

- It is the capability of a parameterized model to find the optimal solution to a given problem
- Trainability maps to the hardness of optimizing a variational landscape
- How efficiently can I explore the parameter space?
  - Are gradients well defined and large?
  - Are there local minima?
  - Can we exploit some structure?

# Barren plateaus

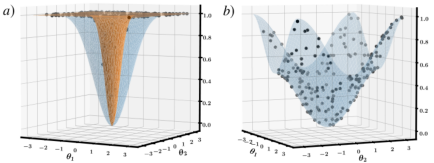
Barren plateaus are known as the problem of exponentially vanishing gradients

$$\mathrm{Var}_{\boldsymbol{\theta}}\, C(\theta) \in \mathcal{O}(e^{-n})$$

They appear if the ansatz forms a 2-design.

**Intuition**

Focus on the quantum state $|\psi\rangle(\theta)$. It lives in a $(2^n)$-dimensional space. The overlap between two random states is in average $2^{-n}$. Therefore $\mathrm{Var}_{\boldsymbol{\theta}}(\langle\psi(\theta)|\, H\, |\psi(\theta)\rangle) = \mathrm{Tr}(H^2)2^{-n}$




The problem is finished **for deep layered ansatzes**! Ragone et al. 2023

- BPs increase with depth; Cerezo, Sone, et al. 2021
- BPs increase with closeness to 2-designs; Holmes et al. 2022
- BPs increase with noise; Wang et al. 2021
- BPs increase with size of the search space: Larocca et al. 2022

# Non-convexity

- Number of local minima
  - If there are many, we would need a method to explore them efficiently
- Goodness of sub-optimal local minima
  - If most minima are close to the global one, we might still be satisfied

**Intuition**

All elements in the ansatz lead to sinusoidal functions $\rightarrow$Plenty of local minima...

Some known result[a]

- If circuits are shallow ($\log(n)$) there exist plenty of local minima
- In addition, a subexponential number of them are close to the global minimum

We need a way to circumvent the non-convexity problem, since it seems to hard to be solved

[a] Anschuetz and Kiani 2022.

# Trainability

## What can we do?

- Use ansatzes with little expressivity
- Use warm starts
- Use physically motivated ansatzes

# Simulability

The purpose of the VQAs is to perform computations not accessible with classical computers

**Tensor-Networks simulation**

- TNs, and in particular Matrix Product States are good at simulating circuits with 1D topology
- They are guaranteed to work with $\log(n)$-depth

**Lie-algebraic simulation**

- From BPs there are classical techniques to simulate VQAs
- These methods are efficient if the resulting circuit has no BPs
  - ... up to a possible initial phase to acquire data with a quantum computer

Goh et al. 2023

## Applicability

TNs are applicable for shallow circuits, which are not suffering from BPs.
Is this the only regime where VQAs are useful?

## Applicability

Does there exist any regime in which VQAs are not classically simulable?[a]

[a] Cerezo, Larocca, et al. 2023.

# Noise robustness

- Noise adds complexity to the quantum state
- However, noise has the common effect of mixing states
- In the end, all noises tend to the same state, and each individual noise contributes only slightly
- Therefore, noise detriments the VQAs, by making the cost functions flatter

**Noise simulation**

There exists clever ways to simulate noise in quantum systems with low effort[ab]
Even a small amount of noise is enough to trigger classical simulation

[a] Fontana et al. 2023.
[b] Rudolph et al. 2023.

Section 5

# Conclusions

# Conclusions

- Variational quantum algorithms gave us strong tools to study quantum systems
- VQAs are challenges for classical optimizers
- There is a strong trade-off between expressivity and trainability
- We have unveiled some cases in which VQAs are not giving quantum advantages
- The performance of VQAs strongly depends on the matches between problem to solve and hardware to run

# Variational Quantum Algorithms

Adrián Pérez Salinas

[1]Lorentz Institute - Leiden University
[2]$\langle aQa \rangle^L$: Applied Quantum Algorithms

Spring school in near-term quantum computing
— Benasque 2024 —