Project 4: Quantum Error mitigation

Anthony BENOIS

SCIPER 368368
École Polytechnique Fédérale de Lausanne (EPFL)

22/01/2024

Introduction

Quantum error mitigation

- NISQ era: necessity to deal with noise (**fault-tolerant** quantum computing is not ready yet)
- **Sources of errors**:
  - decoherence (coupling with environment)
  - coherent errors: inaccurate manipulation of qubits
  - measurements
- Quantum error mitigation:
  - do not explicitly correct errors
  - **reduce** the effect of errors

$\rightarrow$ one strategy: **the zero-noise extrapolation**[1] **(ZNE)**.

[1] Giurgica-Tiron et al., "Digital zero noise extrapolation for quantum error mitigation".

Outlines

Zero-noise extrapolation (ZNE): general idea

General idea:

- **Parameterisation of noise** by a scalar λ
- **Goal**: infer $E^* \equiv E(0)$ from $E_{\text{meas}} = E(\lambda = 1)$
- **Solution**: Measure $E(\lambda)$ for **various** $\lambda \geq 1$, and extrapolate the results ($=$ artificially increase the noise)

Methods:

- Circuit folding
- Layer folding
- parameter noise scaling: mitigation informed by the physical nature of the gates

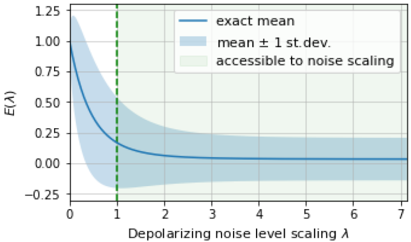


Figure 1: Illustration ZNE

Zero-noise extrapolation (ZNE): general idea

General idea:

- **Parameterisation of noise** by a scalar λ
- **Goal**: infer $E^* \equiv E(0)$ from $E_{\text{meas}} = E(\lambda = 1)$
- **Solution**: Measure $E(\lambda)$ for **various** $\lambda \geq 1$, and extrapolate the results (= artificially increase the noise)

Methods:

- Circuit folding
- Layer folding
- parameter noise scaling: mitigation informed by the physical nature of the gates

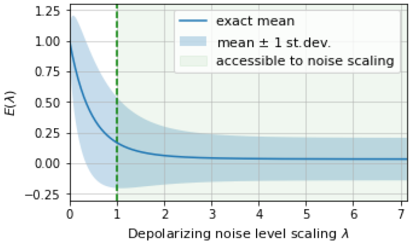


Figure 1: Illustration ZNE

Circuit folding

Original circuit (U is made of d layers): $U = L_1 \ldots L_d$

General idea:

- define the **noise scaling**: $\lambda \equiv d'/d$
- **circuit folding replacement rule**[2]: $U \to U\,(U^\dagger U)^n\, L_d^\dagger L_{d-1}^\dagger \ldots L_{d-s-1}^\dagger L_{d-s-1} \ldots L_d$

$$\lambda = \frac{d(2n+1)+2s}{d} = 1 + \frac{2k}{d}$$

[2] here: **right** folding

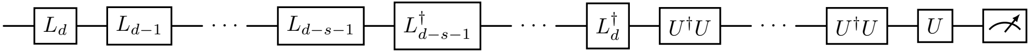
Circuit folding

Original circuit (U is made of d layers): $U = L_1 \dots L_d$

General idea:

- define the **noise scaling**: $\lambda \equiv d'/d$
- **circuit folding replacement rule**[2]: $U \to U\,(U^\dagger U)^n\, L_d^\dagger L_{d-1}^\dagger \dots L_{d-s-1}^\dagger L_{d-s-1} \dots L_d$

$$\lambda = \frac{d(2n+1)+2s}{d} = 1 + \frac{2k}{d}$$



[2] here: **right** folding

Layer folding

Original circuit (U is made of d layers): $U = L_1 \dots L_d$ General idea:

- define the **subset S of indices to fold** (from right, from left, or at random)
- **layer folding replacement rule** :

$$\forall j \in \{1, 2, \dots, d\}, \quad L_j \to \tilde{L}_j = \begin{cases} L_j \, (L_j^\dagger L_j)^n & \text{if } j \notin S, \\ L_j \, (L_j^\dagger L_j)^{n+1} & \text{if } j \in S. \end{cases}$$

$$\lambda \equiv \frac{d'}{d} = \frac{d(2n+1) + 2s}{d} = 1 + \frac{2k}{d}$$

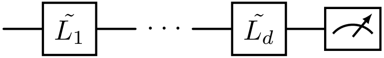
Layer folding

Original circuit (U is made of d layers): $U = L_1 \ldots L_d$ General idea:

- define the **subset** S **of indices to fold** (from right, from left, or at random)
- **layer folding replacement rule** :

$$\forall j \in \{1, 2, \ldots, d\}, \quad L_j \to \tilde{L_j} = \begin{cases} L_j \, (L_j^\dagger L_j)^n & \text{if } j \notin S, \\ L_j \, (L_j^\dagger L_j)^{n+1} & \text{if } j \in S. \end{cases}$$

$$\lambda \equiv \frac{d'}{d} = \frac{d(2n+1) + 2s}{d} = 1 + \frac{2k}{d}$$



Parameter noise scaling

A quantum gate is parameterised by I **classical** control parameters $\theta = (\theta_1, \theta_1, ..., \theta_I)$:

$$G(\theta) = \exp\left(-i\sum_{j=1}^{I} \theta_j H_j\right)$$

Suppose that the parameters are subjected to some uncontrolled classical noise[3](**= stochastic calibration error**):

$$\theta'_j = \theta_j + \varepsilon_j, \qquad \varepsilon_j \sim \mathcal{N}(0, \sigma_j^2), \qquad (\sigma_j^2 \text{ known})$$

Parameter noise scaling $\rightarrow$ arbitrarily add classical noise $\delta_j \sim \mathcal{N}(0, (\lambda - 1)\sigma_j^2)$, so that:

$$\theta'_j = \theta_j + \sqrt{\lambda}\varepsilon_j$$

[3] Gaussian assumption

Parameter noise scaling

A quantum gate is parameterised by I **classical** control parameters $\theta = (\theta_1, \theta_1, ..., \theta_I)$:

$$G(\theta) = \exp\left(-i\sum_{j=1}^{I} \theta_j H_j\right)$$

Suppose that the parameters are subjected to some uncontrolled classical noise[3](**= stochastic calibration error**):

$$\theta_j' = \theta_j + \varepsilon_j, \qquad \varepsilon_j \sim \mathcal{N}(0, \sigma_j^2), \qquad (\sigma_j^2 \text{ known})$$

Parameter noise scaling $\rightarrow$ arbitrarily add classical noise $\delta_j \sim \mathcal{N}(0, (\lambda - 1)\sigma_j^2)$, so that:

$$\theta_j' = \theta_j + \sqrt{\lambda}\varepsilon_j$$

[3] Gaussian assumption

Remarks

- ZNE = running the **same quantum circuit** at **various levels of noise**
  + extrapolation of the measurements
- The noise can only be **increased** ($\lambda \geq 1$)
- Expected to work because an **underlying law** governing expectation values as a function of noise is assumed

Case of depolarizing noise :

$$\rho_j \xrightarrow{\text{noisy } L_j} \rho_{j+1} \xrightarrow{\text{noisy } L_{j+1}} \rho_{j+2} \quad \Longleftrightarrow \quad \rho_j \xrightarrow{\text{noisy } L_j L_{j+1}} \rho_{j+2}$$

$$\rho \xrightarrow{\text{noisy circuit + folding}} p^\lambda U \rho U^\dagger + (1 - p^\lambda)\mathbb{I}/D \quad \Longrightarrow \quad E(\lambda) = a + bp^\lambda$$

Remarks

- ZNE = running the **same quantum circuit** at **various levels of noise** + extrapolation of the measurements
- The noise can only be **increased** ($\lambda \geq 1$)
- Expected to work because an **underlying law** governing expectation values as a function of noise is assumed

Case of depolarizing noise :

$$\rho_j \xrightarrow{\text{noisy } L_j} \rho_{j+1} \xrightarrow{\text{noisy } L_{j+1}} \rho_{j+2} \iff \rho_j \xrightarrow{\text{noisy } L_j L_{j+1}} \rho_{j+2}$$

$$\rho \xrightarrow{\text{noisy circuit + folding}} p^\lambda U \rho U^\dagger + (1 - p^\lambda)\mathbb{I}/D \implies E(\lambda) = a + bp^\lambda$$

Extrapolation

Polynomial extrapolation:

- Linear regression
- Quadratic regression
- Richardson interpolation

Poly-exponential extrapolation:

- Poly-exponential regression
- Exponential regression:

$$E_{exp}(\lambda) = a \pm \mathrm{e}^{z_0 + z_1 \lambda}$$

- Adaptive exponential regression (cf. appendix)

ZNE with randomized circuits

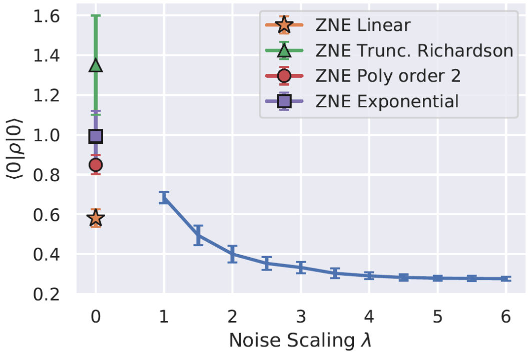


Figure 2: Comparison of extrapolation methods (50 two-qubit randomized circuits). The true zero-noise value is 1.

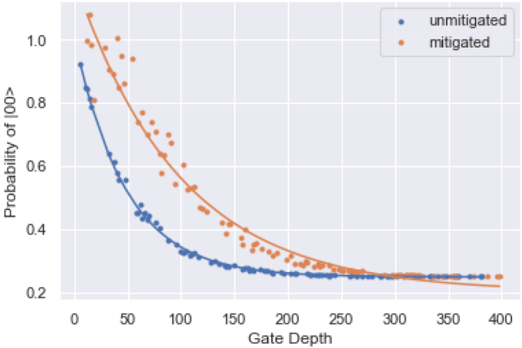


Figure 3: Comparison of two qubit randomized circuits with and without error-mitigation. The decay goes from 97.9% (U) to 99% (M).

Error distributions

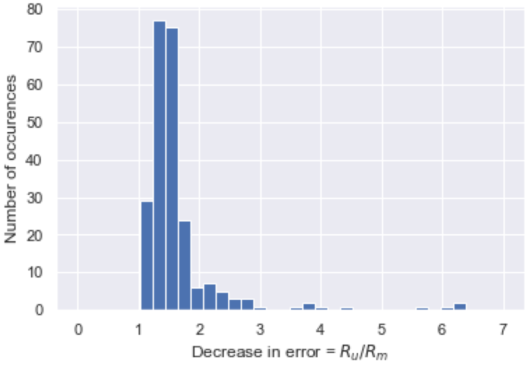


Figure 4: Comparison of improvements from ZNE (using quadratic extrapolation with folding from left) and 250 6-qubits random circuits.

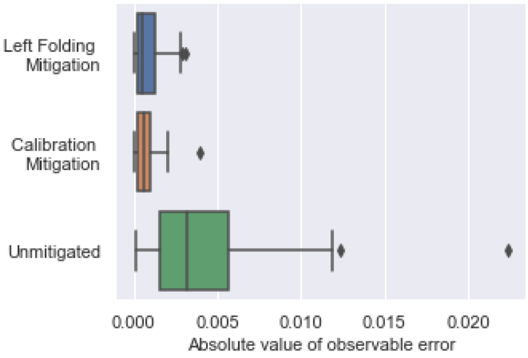


Figure 5: Errors without mitigation, with parameter noise mitigation, and with unitary folding mitigation (50 random six-qubit).

Application to other Algorithms

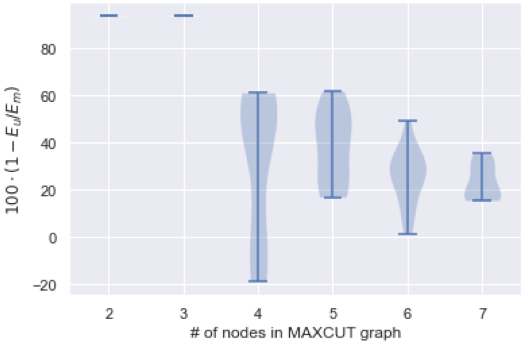


Figure 6: Percent closer to optimal on random MAXCUT executions. Algorithm: QAOA (2 executions), circuit-folding.

Scaling	Extrapolation	Error % (dep.)	Error % (amp. damp.)
none	unmitigated	29.9 ± 5.1	16.7 ± 4.0
circuit	linear $(d = 1)$	14.6 ± 4.6	5.40 ± 2.3
circuit	quadratic $(d = 2)$	6.35 ± 3.6	3.53 ± 3.4
circuit	Richardson $(d = 3)$	17.6 ± 11	17.9 ± 16
circuit	exponential $(a = 0.25)$	2.73 ± 1.9	2.06 ± 1.6
circuit	adapt. exp. $(a = 0.25)$	$\mathbf{1.27} \pm \mathbf{1.1}$	2.69 ± 2.8
at random	linear $(d = 1)$	15.6 ± 5.3	5.20 ± 2.4
at random	quadratic $(d = 2)$	5.54 ± 4.4	8.00 ± 8.1
at random	Richardson $(d = 3)$	30.0 ± 24	24.0 ± 18
at random	exponential $(a = 0.25)$	2.84 ± 1.8	$\mathbf{0.95} \pm \mathbf{1.0}$
at random	adapt. exp. $(a = 0.25)$	1.77 ± 1.4	2.18 ± 1.2
from left	linear $(d = 1)$	14.4 ± 4.5	5.16 ± 2.3
from left	quadratic $(d = 2)$	6.73 ± 3.7	3.88 ± 3.7
from left	Richardson $(d = 3)$	18.4 ± 12	16.1 ± 13
from left	exponential $(a = 0.25)$	3.17 ± 2.1	2.19 ± 2.0
from left	adapt. exp. $(a = 0.25)$	1.43 ± 1.1	3.08 ± 3.6

Figure 7: Absolute errors for various mitigation techniques. Mean depth: 27.

Quantum Fourier Transform

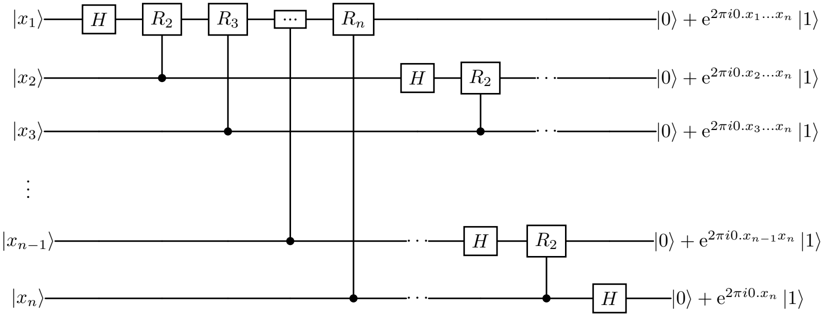
- Classical Fourier transform:

$$\mathbf{FT} : \mathbb{C}^N \longrightarrow \mathbb{C}^N$$
$$(x_0, x_1, \dots, x_{N-1}) \longmapsto (y_0, y_1, \dots, y_{N-1})$$
$$\text{where} \quad \forall k, \quad y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \, \omega_N^{-nk}$$
$$\text{and} \quad \omega_N = \mathrm{e}^{\frac{2\pi i}{N}}$$

- QFT: maps the **probability amplitudes** according to the above formulas. That is:

$$|x\rangle = \sum_{i=0}^{N-1} x_i \, |i\rangle \xrightarrow{\mathbf{QFT}} \sum_{k=0}^{N-1} y_k \, |k\rangle$$

QFT Algorithm



- $x_1 \ldots x_n$ is the bit-wise expression of x
- H is the **Hadamard** gate and the R_k's are **phase gates**:

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^k} \end{pmatrix}$$

ZNE with QFT : 3-qubit case

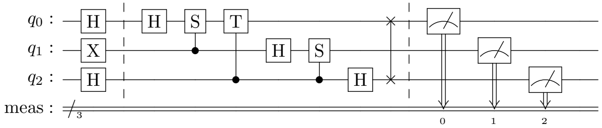


Figure 8: Simplification of the QFT algorithm in the 3-qubit case

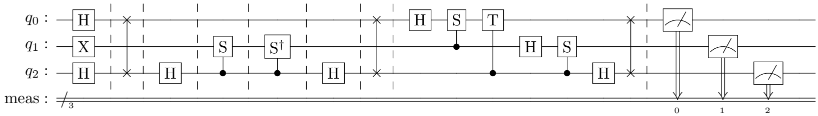


Figure 9: Circuit folding for $\lambda = 2$ ($n = 0$, $k = 7$, $s = 3$, $d = 7$) in the 3-qubit case

ZNE with QFT : 3-qubit case

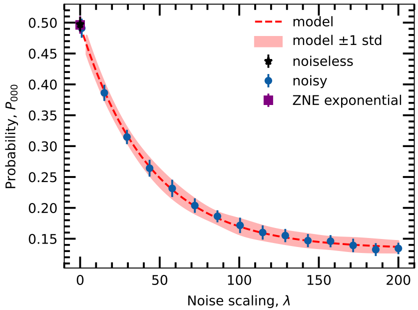


Figure 10: **Exponential ZNE**: probability of the $|000\rangle$ outcome as a function of the **noise scaling**

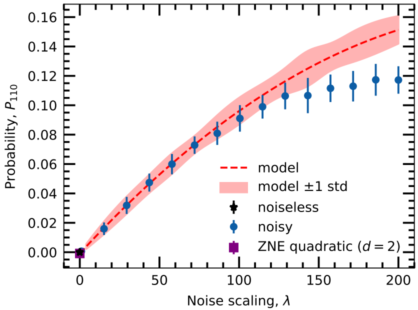


Figure 11: **Quadratic ZNE**: probability of the $|110\rangle$ outcome as a function of the **noise scaling**

ZNE with QFT : 3-qubit case

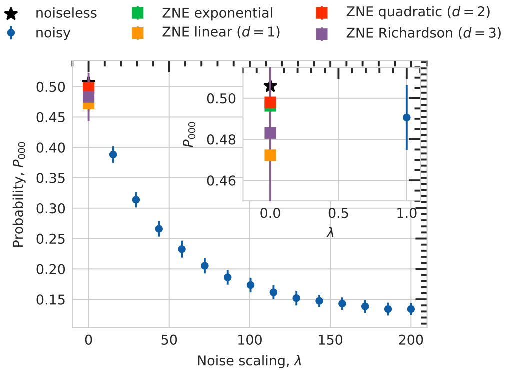


Figure 12: **ZNE** for various extrapolation methods. **Layer** folding, 50 replications

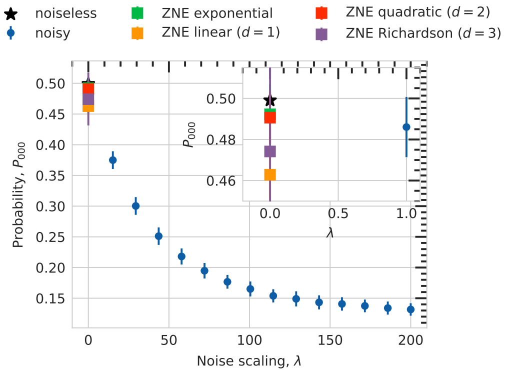


Figure 13: **ZNE** for various extrapolation methods. **Circuit** folding, 50 replications

ZNE with QFT : 3-qubit case

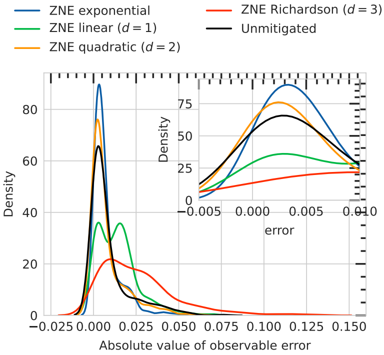


Figure 14: KDE plot of the absolute value of the **error** for various extrapolation methods. **Layer** folding, 50 replications

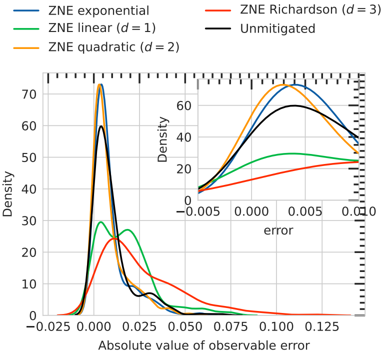


Figure 15: KDE plot of the absolute value of the **error** for various extrapolation methods. **Circuit** folding, 50 replications

ZNE with QFT : 3-qubit case

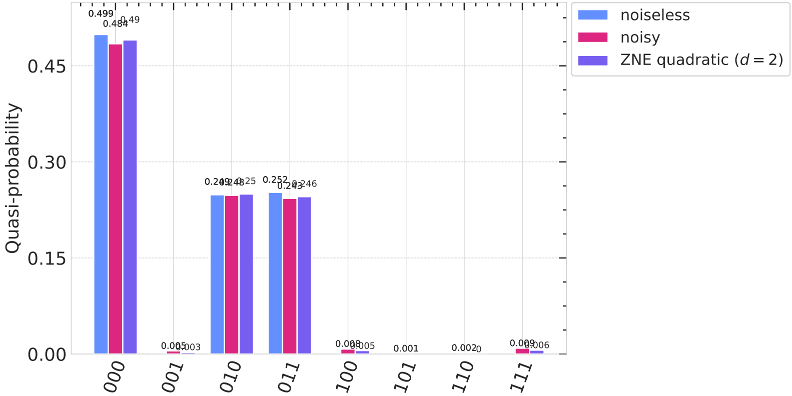


Figure 16: **Probability distribution** of all binary outputs. Averaged over 50 replications, using circuit folding and **quadratic extrapolation**.

ZNE with QFT : 3-qubit case

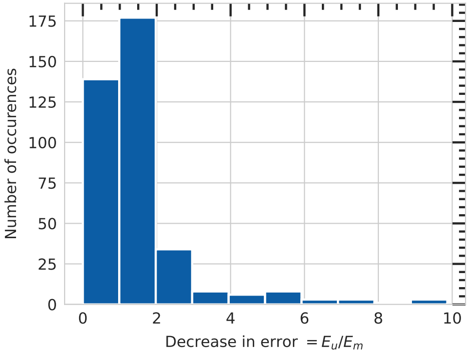


Figure 17: Frequency as a function of the **decrease in the absolute error** for all binary outputs. 50 replications, circuit folding, quadratic extrapolation. **36% of the values are below 1**.

Discussion

This method presents several **advantages**:

- it is **digital**: agnostic of the hardware !
- underlying **noise model** does not need to be exactly known
- error mitigation techniques give practical benefits: **improvement of the results** by several factors have been shown
- it applies to **variational circuits** as well

But it also has **limitations**:

- **measurement** and **state preparation errors** are not addressed (not a function of the depth of the circuit)
- may **fail** to address **systematic or coherent errors** (folding *undoes* these errors).
- folding **does not exploit** any **specific structure of noise**

Thank you for your attention

References

Giurgica-Tiron, Tudor et al. "Digital zero noise extrapolation for quantum error mitigation". In: *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE. 2020, pp. 306–316.

Zhou, Leo et al. "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices". In: *Physical Review X* 10.2 (2020), p. 021067.

Adaptative ZNE

Goal: pick the next noise scaling factor λ and the associated number of shots N_λ **such that the MSE of the estimator is minimal**.
$\rightarrow$ Method in contrast to **non-adaptive methods**, where the noise scaling factors are set beforehand.

Example: exponential ZNE ($E_{exp}(\lambda) = a + be^{-c\lambda}$, and a is known)
While $N_{used} < N_{max}$:

1. compute $\lambda_{next} = \lambda_1 + \alpha/c$ and N_{next}
2. measure expectation values
3. fit *data* to extract c'

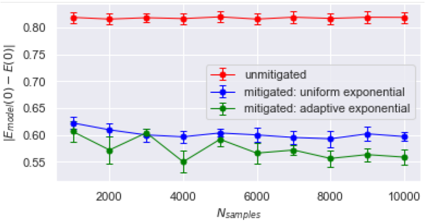


Figure 18: Comparison of adaptive and non-adaptive exponential zero noise extrapolation, given a fixed budget of samples

MAXCUT problem

Suppose to have a graph $G = (V, E)$.

Goal: color the vertices of this graph using two colors in such a way that the maximum possible number of edges has its two vertices of different colors.

($=$ maximise the interactions between two groups)

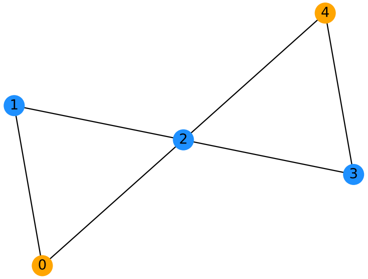


Figure 19: Example of bi-colored graph: solution provided by QAOA

QAOA - Quantum Approximation Optimization Algorithm

Goal: Find a binary string which maximizes a classical objective function

$$\mathbf{C}: \{\pm 1\}^N \longrightarrow \mathbb{R}$$
$$z \longmapsto C(z)$$

Method: Map each binary variable z_i to a quantum spin σ_i^z, and define the Hamiltonians:

$$H_C = C(\sigma^z) = C(\sigma_1^z, ..., \sigma_N^z)$$

$$H_B = \sum_{j=1}^{N} \sigma_j^x$$

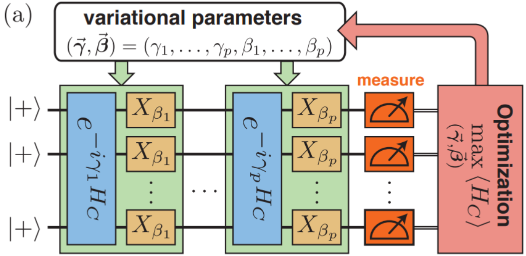


Figure 20: Schematic of a p-level Quantum Approximation Optimization Algorithm[a]

[a] Zhou et al., "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices".