Electronics
Helicopters
Aircraft
Cyber & Security
Space
Unmanned Systems
Aerostructures

Challenges to gate-based quantum optimization algorithms for industrial use-cases

Quantum Computing Research Leonardo

Matteo Vandelli
Research Fellow, HPC Lab - Quantum Computing
Leonardo Labs, Genova – Fiumara, Italy
matteo.vandelli.ext@leonardo.com

CINECA, HPCQC Workshop, 15.12.2023

Combinatorial optimization: the antenna placement problem

Ising problems:

$$H_S = \sum_{i \neq j}^{N} W_{ij}\, s_i s_j + \sum_{i=1}^{N} A_i s_i \quad with \quad s_i \in \{-1, 1\}$$

Solution: combination of $S = \{s_i\}$ that minimizes H_S ?

Industrial problems:
1. Unstructured problems with few or no symmetries
2. Weighted graphs, Constraints (graphs become fully connected)
3. Focus is on the solution string

Countless applications: logistics, task scheduling, antenna placement, satellite mission planning, resource management, components on an aircraft…

Use case in our work

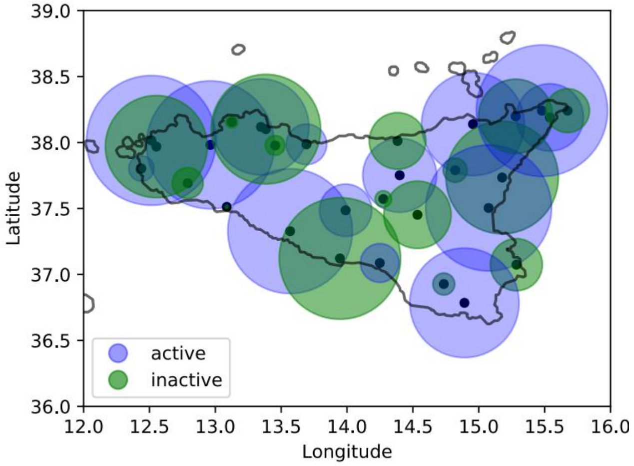
M. Vandelli, A. Lignarolo, C. Cavazzoni, D. Dragoni, arXiv:2311.11621

Optimization of a network of antennas operating in an emergency scenario:
- with scarce resources
- only a specified number of antennas can be simultaneously operated.

- Maximize the coverage of the signal generated by the antennas
- Minimize the overlaps between the signals
- Limited number of antennas

Antenna problem

Ising model



$$H(z;\ \xi, \lambda, N_t) = H_O(z) + H_C(z) + H_P(z)$$

with

$$H_O(z) = \sum_{i \neq j} z_i J_{ij} z_j$$

$$H_C(z) = -\xi \sum_i A_i z_i$$

$$H_P(z) = \lambda \left(\sum_{i \neq j} z_i z_j + \delta N_t \sum_i z_i \right).$$

$\lambda = 0 \ \rightarrow$ unconstrained
$\lambda > 0 \ \rightarrow$ constrained

Emulating quantum algorithms on the *davinci-1* cluster

Two algorithms described by the same circuit:

$|\beta,\gamma\rangle = U(\beta,\gamma)\,|+\rangle^{\otimes n}$ with $U(\beta,\gamma) = \prod_{n=1}^{p} e^{-i\beta_n \sum_i X_i}\, e^{-i\gamma_n H_S}$

QAA: linear Trotterization → $\beta = \Delta\left(1-\frac{k}{p}\right),\ \gamma = \Delta\frac{k}{p}$

QAOA: optimization in 2*p*-dimensions, $(\beta,\gamma) = \mathrm{argmin}\, f(\beta,\gamma)$

[QAA: Fahri *et al.*, *arXiv*:0001106; QAOA: Fahri *et al.*, *arXiv*:1411.4028]

All the calculations performed on the proprietary davinci-1 cluster equipped with AMD EPYC 7402 24-Core CPUs and NVIDIA A100 GPUs.

M. Vandelli, A. Lignarolo, C. Cavazzoni, D. Dragoni, arXiv:2311.11621

Total number of layers

$p_{tot} = p \cdot N_{iter}$

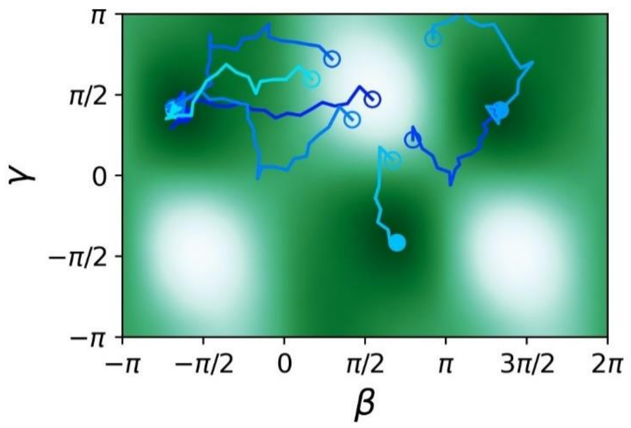
$N_{iter}(QAA) = 1$

$N_{iter}(QAOA) = 50$

p → Single-circuit depth

p_{tot} → Time complexity: Total number of sequentially applied layers

- exact emulation based on Qiskit
- MPI framework for global optimization using a **multi-walker optimization** initialized around an **INTERP** strategy
- COBYLA minimizer for local optimization



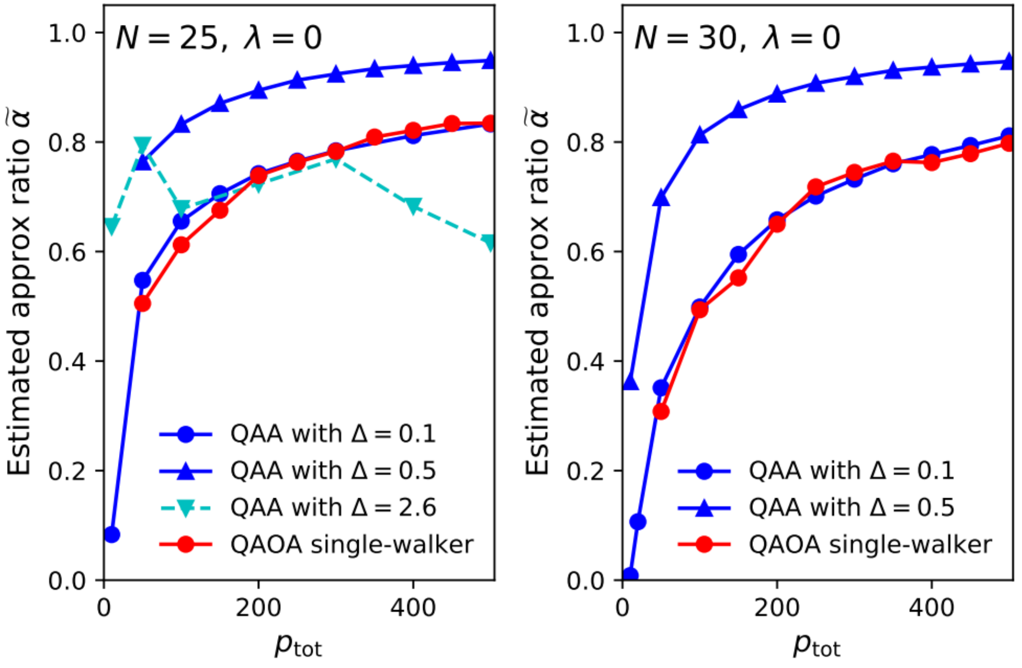
QAOA: 32 walkers x (8 CPU cores + 1 GPU)

QAA & single-walker QAOA: 48 CPU cores + 1 GPU

Approximation ratio at large depths

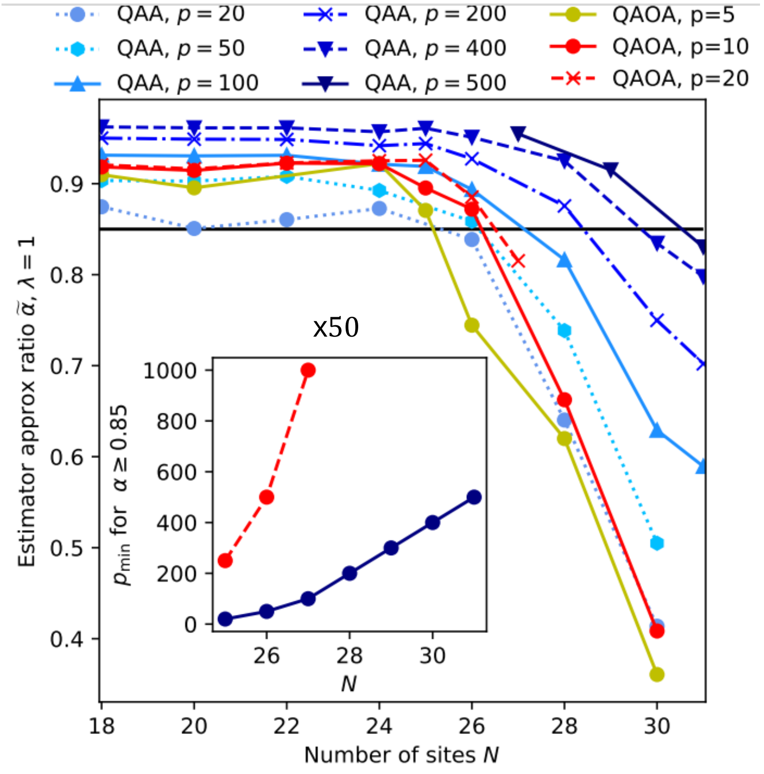
Approximation ratio: $\alpha = \langle H_\lambda \rangle / H_{min}$

Fixed shots estimator: $\tilde{\alpha} = \frac{(\sum_k H(z_k)/N_{meas})}{H_{min}}$



Approximation ratio as a function of the number of qubits:
Constrained case with $\lambda = 1$

$$p_{min}(QAA) \sim 100\ (N - 26)$$
$$p_{min}(QAOA) \sim \exp(N)\ \text{or}\ poly(N)\ \text{(not linear)}$$



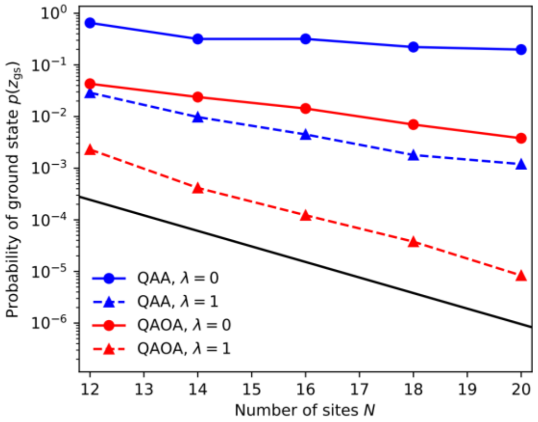
M. Vandelli, A. Lignarolo, C. Cavazzoni, D. Dragoni, arXiv:2311.11621

Generation of high-quality solutions

Probability of measuring the ground state

Emulation using the full statevector:

- QAOA has depth $p = 10$, while QAA has depth $p = 500$ (fixed computational time $p_{tot} = 500$ in both cases).
- In the constrained case $(\lambda = 1)$, $p(z_{gs})$ is much lower.



Method	Measurements
$QAA, \lambda = 0$	1.07^N
$QAOA, \lambda = 0$	1.36^N
$QAA, \lambda = 1$	1.30^N
$QAOA, \lambda = 1$	1.67^N
Flat state	2.00^N

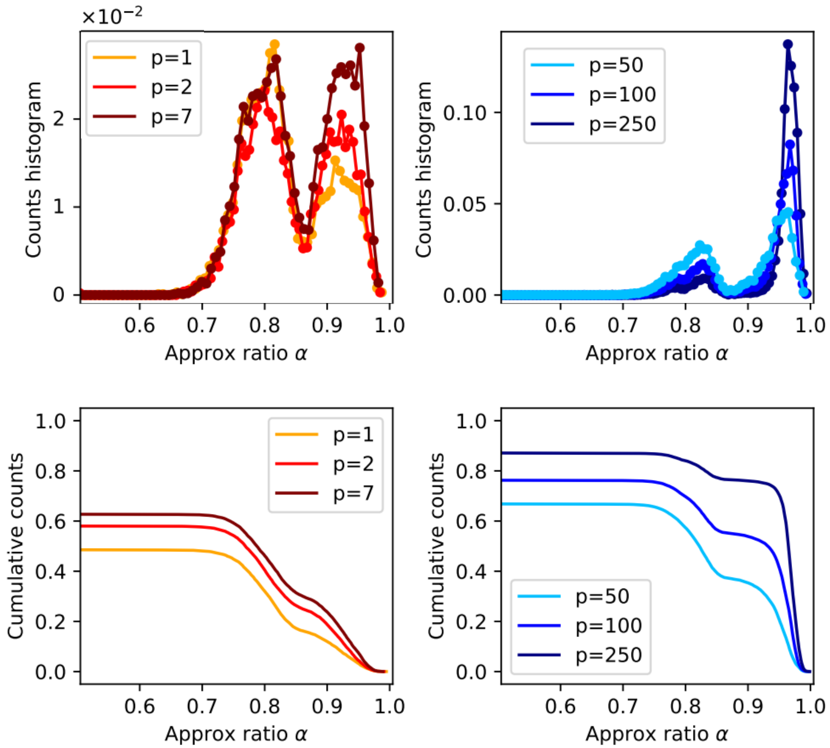
For this use case, the probability $p(z_{gs})$ decreases exponentially with the system size in all cases.

M. Vandelli, A. Lignarolo, C. Cavazzoni, D. Dragoni, arXiv:2311.11621

Generation of high-quality strings

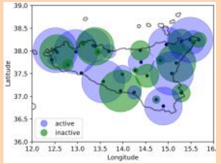
$N = 30$, $N_{shots} = 10^4$, different values of p

- The cumulative counts estimate the probability of measuring a string with higher approx ratio.
- The probability is already large enough to guarantee a measurement of an accepatble string if we fix $\alpha \leq 0.95$



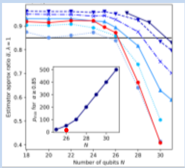
Summary & conclusions

Antenna placement problem: Unstructured problem with a weighted graph
$\lambda = 0$: sparse connectivity
$\lambda > 0$: full connectivity

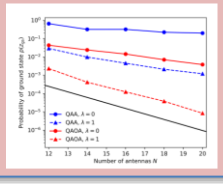


Approximation ratio with QAA and QAOA: rather robust against shot noise
Fixed α requires $p \sim N$ for QAA (and increases faster for QAOA)

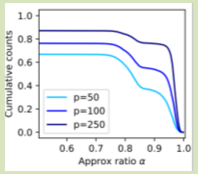
QAA: very deep circuits; **QAOA**: NP-hard optimization in p



Probability: for this protoypical unstructured problem, $p(z_{gs})$ decreases exponentially *vs* N for both QAA and QAOA



Cumulative distribution: These algorithms allow to measure good strings with high probability, if some tolerance is allowed



M. Vandelli, A. Lignarolo, C. Cavazzoni, D. Dragoni, arXiv:2311.11621

THANK **YOU**
FOR YOUR ATTENTION

leonardo.com

E-mail: matteo.vandelli.ext@leonardo.com

Academic *vs* practical optimization: our contribution

Quantum algorithms from an «academic» point of view:

1. Regular problems with symmetries
2. Unweighted graphs/integer parameters, sparse connectivity
3. Focus often on approximation ratio (cost, energy,…)

Practical point of view:

1. Unstructured problems with few or no symmetries
2. Weighted graphs, Constraints (graphs become fully connected)
3. Focus is on the solution string

Additionally, industrial problems are large → number of variables > 10^3-10^4

Comparison between 2 algorithms:

1) QAA
2) QAOA

Comparison between 2 different problems:

1) sparsely connected antennas ($\lambda = 0$, unconstrained)
2) Fully connected antennas ($\lambda = 1, N_t = \lfloor N/2 \rfloor$)

Metrics:

1) Approximation ratio α (single instance)
2) Probability of exact state $p(z_{gs})$
3) Cumulative probability $p(z \mid \alpha(z) \geq \alpha)$

Approximation ratio: $\alpha = \langle H_\lambda \rangle / H_{min}$ ← Emulator of the full state-vector

Its estimator is $\tilde{\alpha} = \frac{(\sum_k H(z_k)/N_{meas})}{H_{min}}$ ← Emulation with fixed number of shots