Projecting NISQ-era quantum advantage with QAOA

Gian Giacomo Guerreschi[(1)], Jason Larkin[(2)] , Matias Jonsson [(2)] , Daniel Justice [(2)]

(1) Intel Labs, (2) Carnegie Mellon Univ - SEI

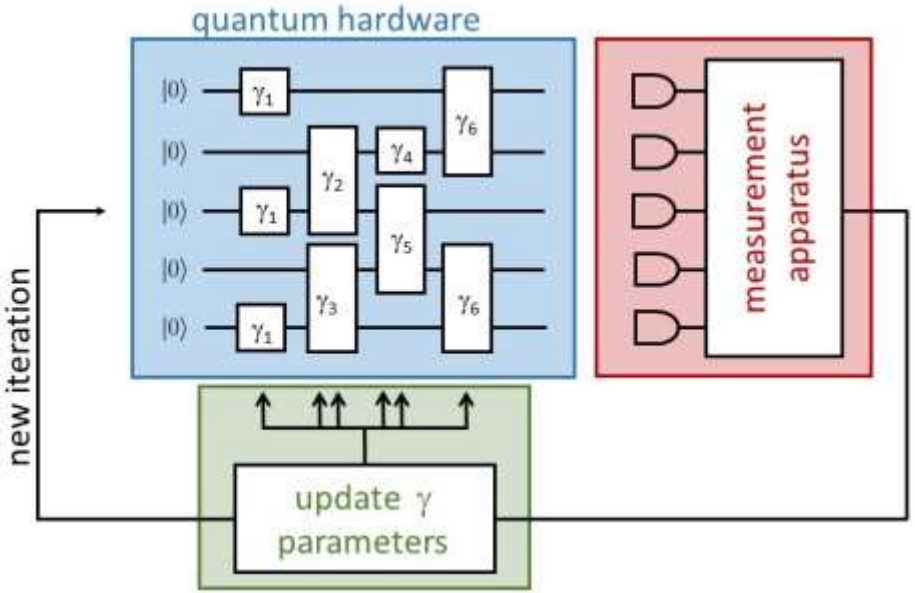
[Guerreschi & Matsuura, *Scientific Reports*, *9*, 6903 (2019)]
[Larkin, Justice & Guerreschi, arXiv:2005.XXXX] <in preparation>

Legal

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Variational quantum algorithms

- minimization problems
- approximate solution



Variational quantum algorithms

Variational Quantum Algorithms (VQAs) are ideally suitable for short-term applications:

- hybrid quantum-classical algorithms: part of the computational cost is offloaded to classical processors
- circuits are relatively shallow: without expensive subroutines like quantum phase estimation
- robust to systematic noise: from the optimization of the circuit's parameters
- address very important problems: energy of molecular systems (VQE), binary optimization (QAOA), …

Variational quantum algorithms

Variational Quantum Algorithms (VQAs) are ideally suitable for short-term applications:

-
-
-
-

Quantum supremacy have been claimed last autumn.
Are we ready for applications?
Today? Tomorrow?

Variational quantum algorithms

Hardware has limited connectivity.
Extra operations required for routing.

Execution time includes quantum circuit and classical optimization, but also communication latency.

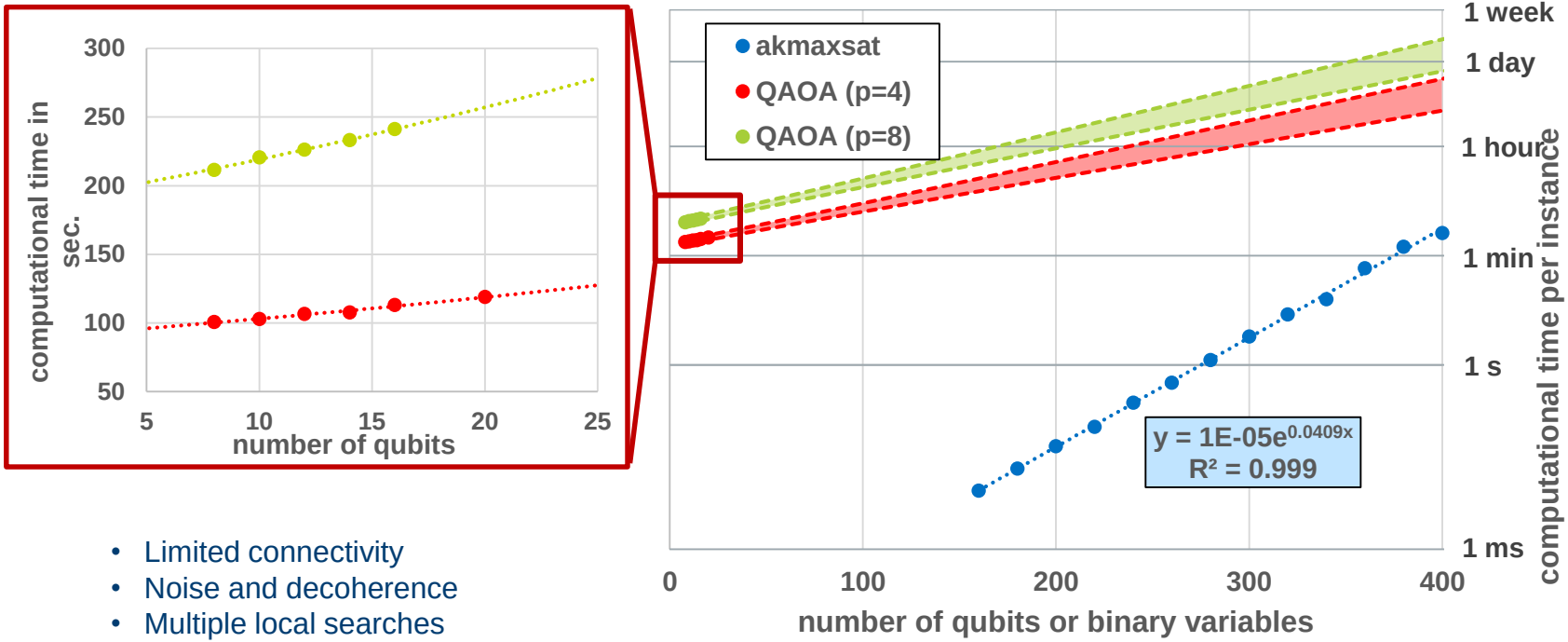
Variational Quantum Algorithms (VQAs) are ideally suitable for short-term applications:

- hybrid quantum-classical algorithms: part of the computational cost is offloaded to classical processors
- circuits are relatively shallow: without expensive subroutines like quantum phase estimation
- robust to systematic noise: from the optimization of the circuit's parameters
- address very important problems: energy of molecular systems (VQE), binary optimization (QAOA), …

Fierce competition from classical alternatives

100s or 1000s of different parameter choices

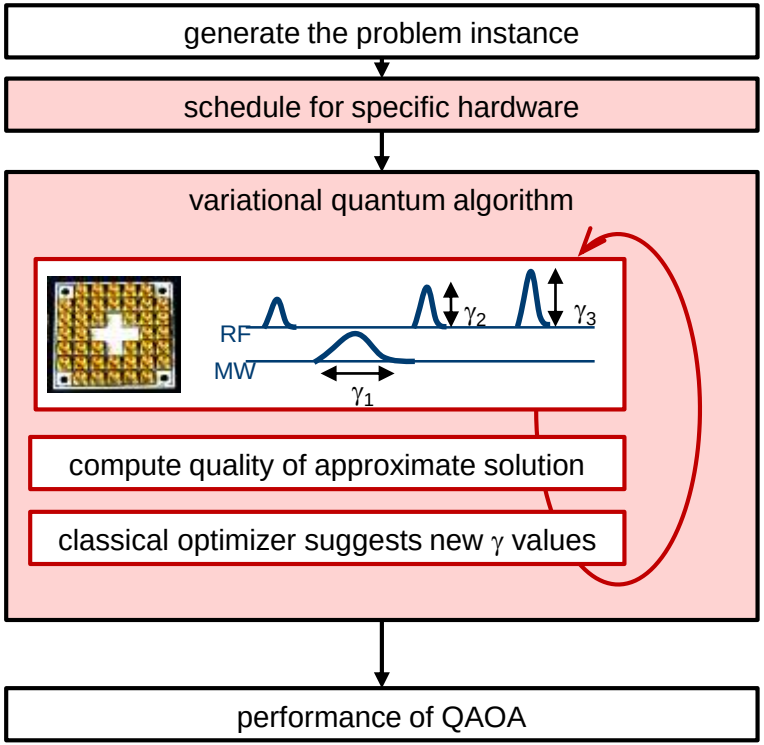
Benchmarking QAOA for Max-Cut on 3-regular graphs

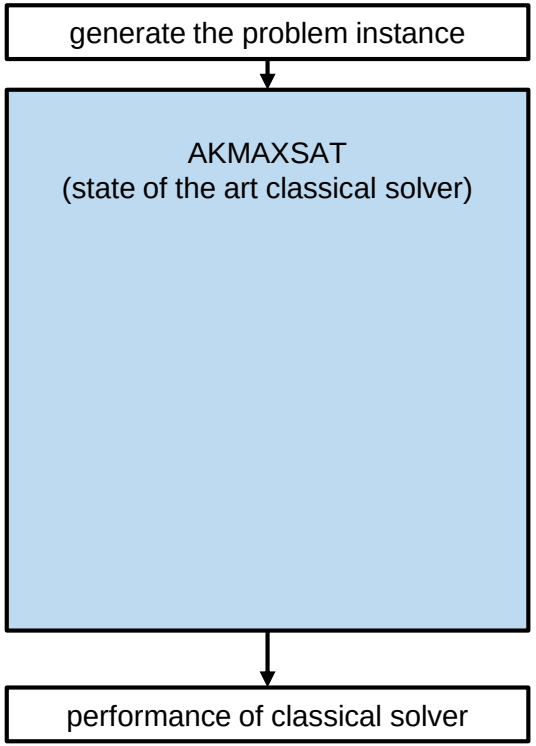


- Limited connectivity
- Noise and decoherence
- Multiple local searches

[Guerreschi & Matsuura, *Scientific Reports*, *9*, 6903 (2019)]

Last year's approach





Last year's approach… and its extension

We learned that QAOA must improve by orders of magnitude!

- Quantum hardware: more qubits, high-fidelity gates, 2D connectivity, parallel operations

  [Kandala et al., Nature, 567, 491-495 (2019)] [Arute et al., *Nature*, *574*, 505–510 (2019)]

- Software tools: compilers and mappers

  [Lao et al., arXiv:1908.04226] [Guerreschi, arXiv:1912.00035]

- Parameter optimization: less iterations, using gradients, different figure of merit guiding optimization, …

  [Guerreschi & Smelyanskiy, arXiv:1701.01450] [Barkoutsos et al., arXiv:1907.04769]

- Initial guess of parameters' values: correlated with problem instances in same class

  [Wecker et al., *Physical Review A*, *94*, 022309 (2016)] [Brandão et al., arXiv:1812.04170]

- Problem selection: look for classical hard instances

  [Moussa, Calandra, & Dunjko, a*rXiv:2001.08271*]

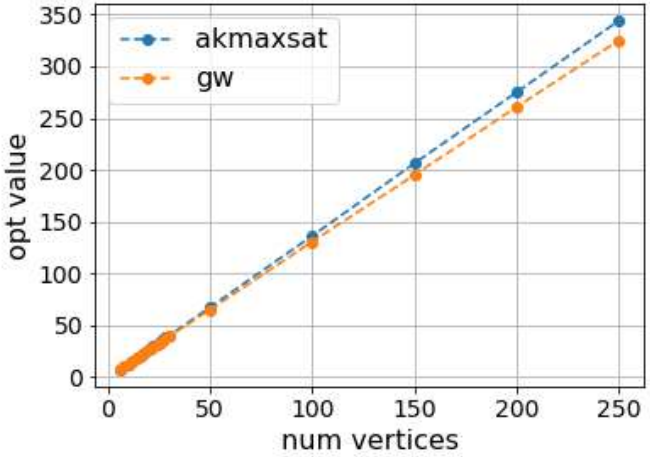
- But also… use classical approximate solvers to compare against

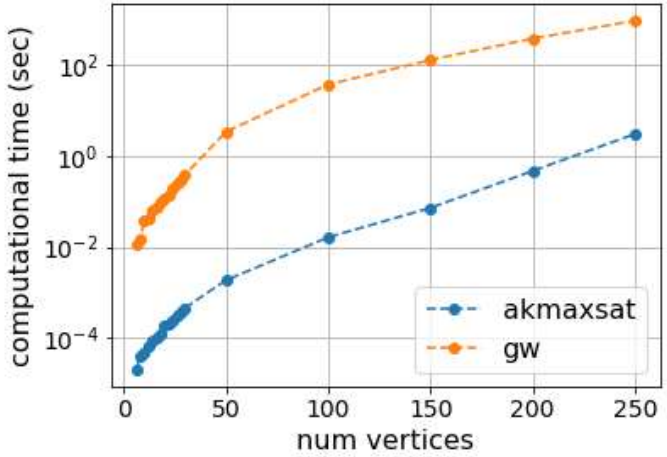
  [Goemans & Williamson, Journal Of The ACM, 42(6), 1115-1145 (1995)]

Classical (approximate) solvers: akmaxsat, GW, …

Random 3-regular graphs, avgeraged over 20 instances:

- **akmaxsat → exact solver for SAT**
- **Goemans-Williamson → heuristic alg. for MaxCut**

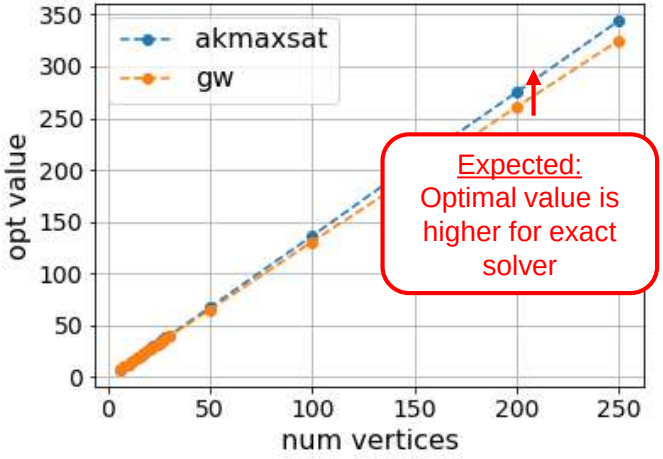


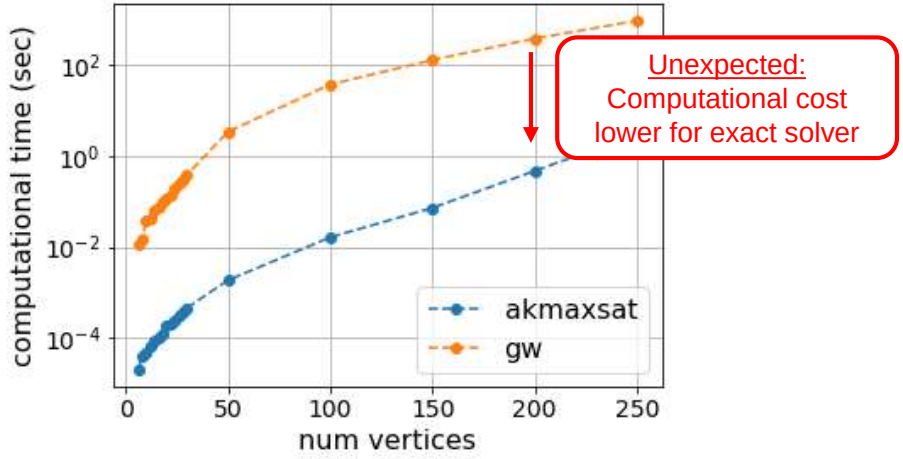


Classical (approximate) solvers: akmaxsat, GW, …

Random 3-regular graphs, avgeraged over 20 instances:

- **akmaxsat → exact solver for SAT**
- **Goemans-Williamson → heuristic alg. for MaxCut**

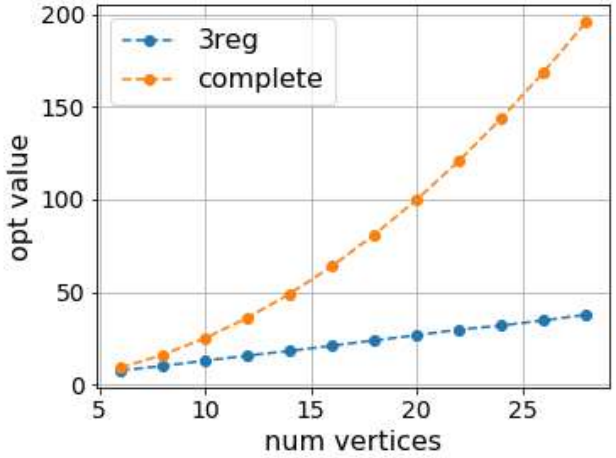


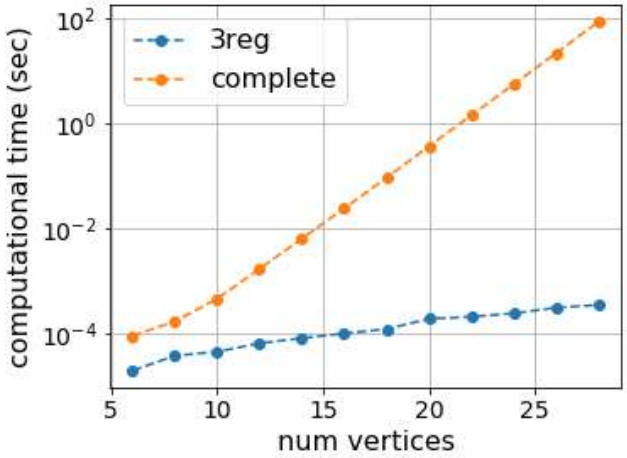


Problem selection: k-regular, complete, Ergdos, …

akmaxsat:

- **3-regular graphs → num edges = 3 n/2**
- **complete graphs → num edges = n (n-1)/2**

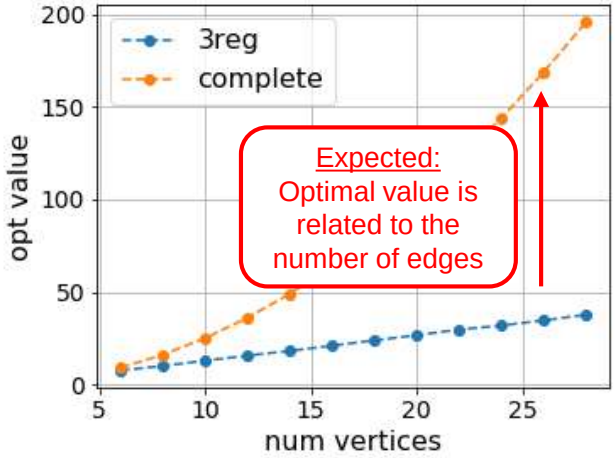


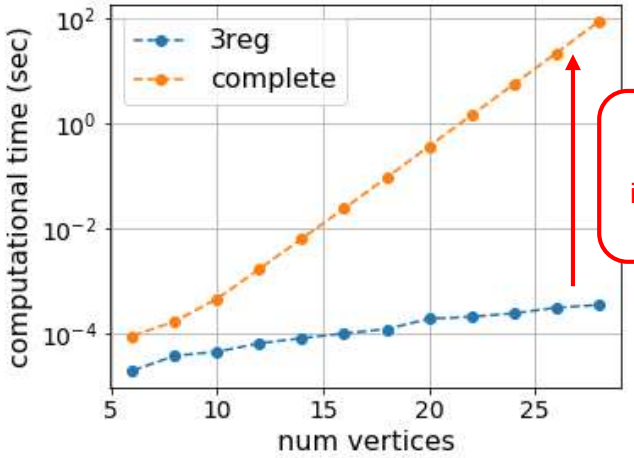


Problem selection: k-regular, complete, Ergdos, …

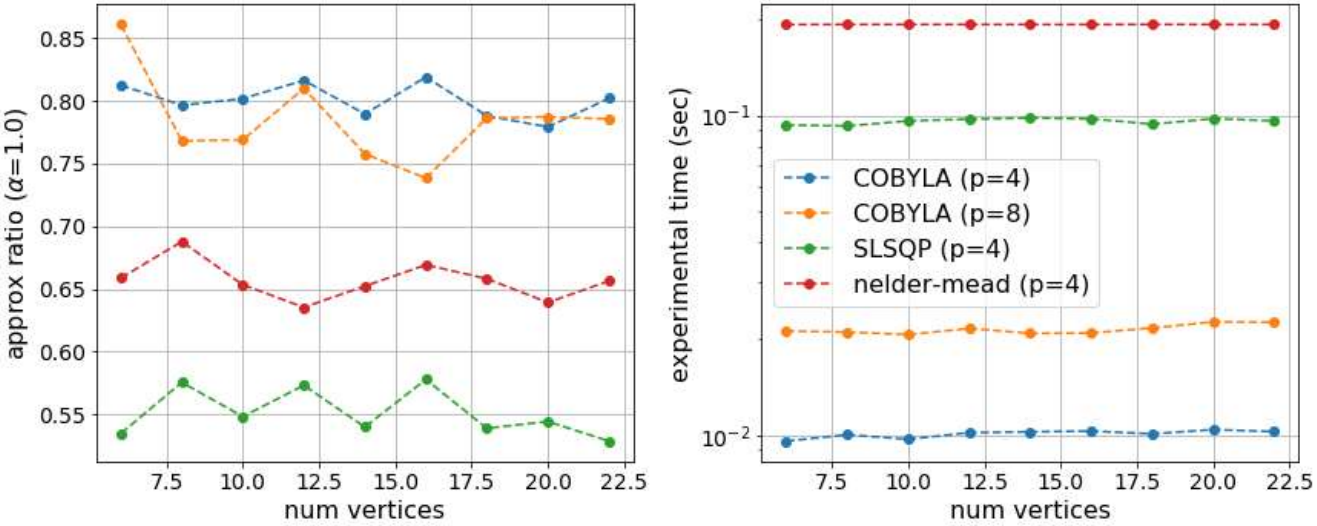
akmaxsat:

- **3-regular graphs → num edges = 3 n/2**
- **complete graphs → num edges = n (n-1)/2**



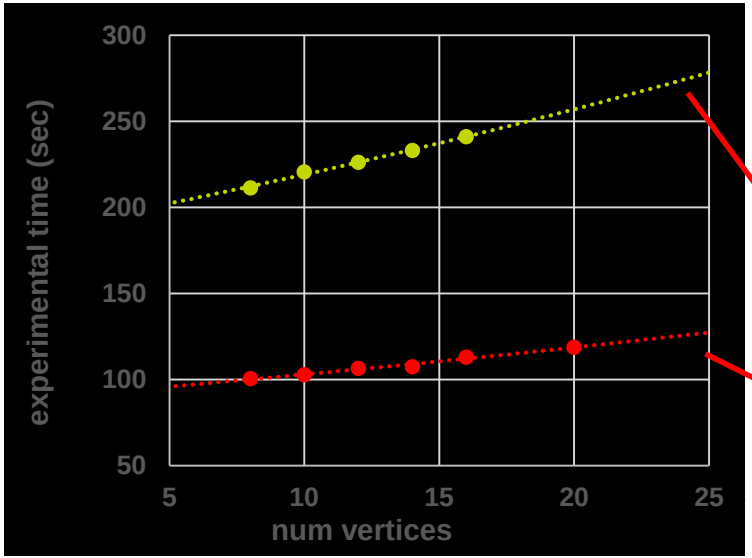


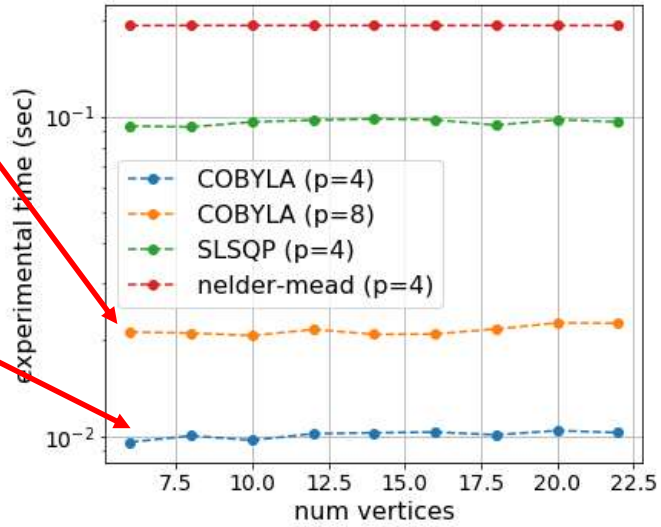
Parameter optimization: COBYLA, SLSPQ, NM, …



Experimental time reduced by 4 orders of magnitude

Nelder-Mead with 10,000 samples per function evaluation

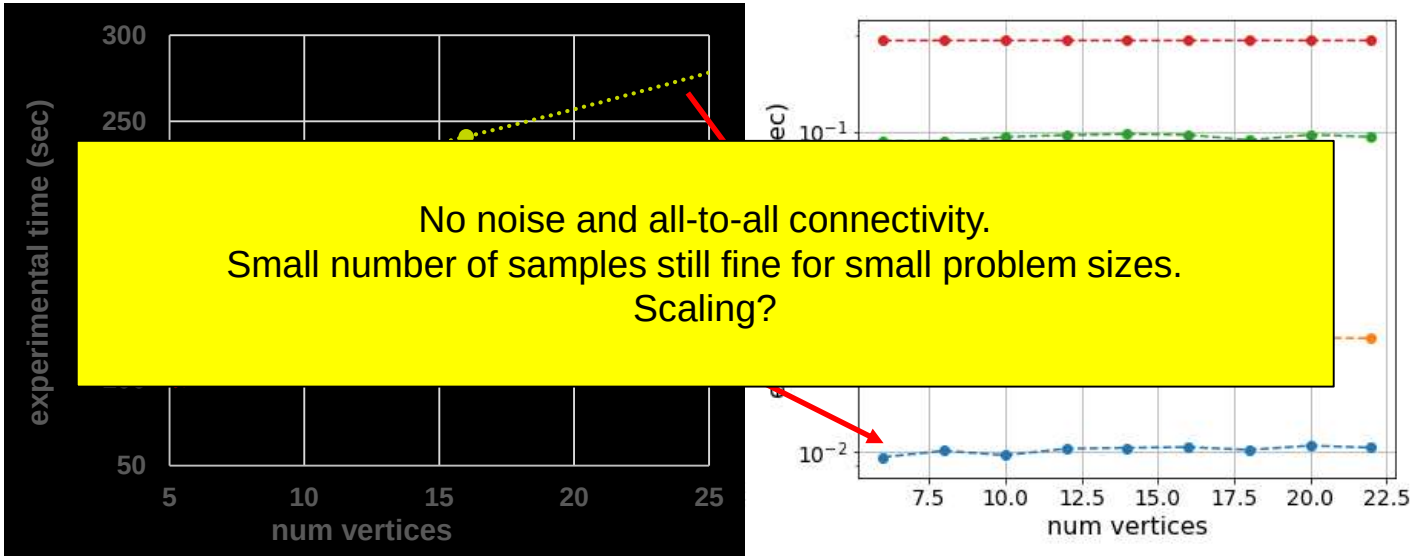




COBYLA with 100 samples per function evaluation

Experimental time reduced by 4 orders of magnitude

Nelder-Mead with 10,000 samples per function evaluation

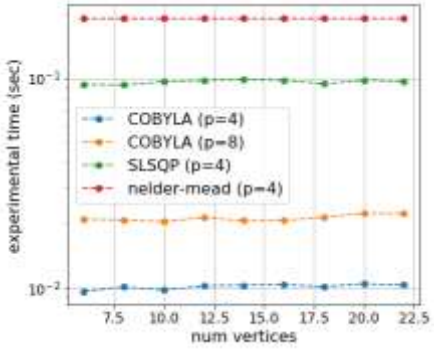


No noise and all-to-all connectivity.
Small number of samples still fine for small problem sizes.
Scaling?

Carnegie Mellon University

[Guerreschi & Matsuura, *Scientific Reports*, *9*, 6903 (2019)]

[Larkin, Justice & Guerreschi, arXiv:2005.XXXX] <in preparation>



Project NISQ-era advantage:
1. compared to (approximate) solvers
2. focusing on classically hard instances
3. with most suitable classical optimizers and protocols

Together with: Anne Matsuura (Intel)
Jongsoo Park (Intel → FB)
Mikhail Smelyanskiy (Intel → FB)

Thanks for the attention!