3rd place solution in QAGC

Kohei Morimoto

Graduate School of Informatics,

Kyoto University

KYOTO UNIVERSITY

Self-Introduction

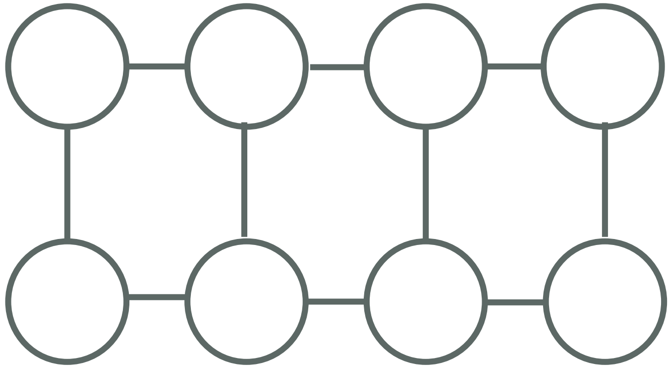
- My major is Control theory and System Identification
  - I even didn’t know what is fermion.

- I am personally interested in Quantum algorithms
  - Mitou-target project (2021~2022)
    - Improve Grover adaptive search by classical optimizer “CMAES”

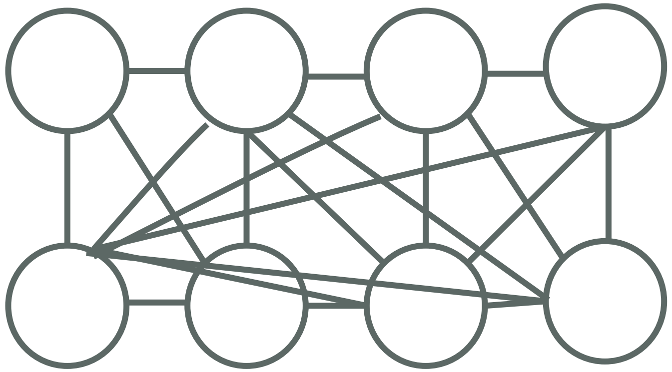
- I joined this competition because
  - VQE is one of the most promising applications in NISQ
  - I want to learn more about VQE through this competition

Difficulty of this competition

- Orbital rotation makes the problem very difficult…
  - Sparse Connection → Dense Connection



Fermi-Hubbard Model



Orbital Rotated Fermi-Hubbard

- Execution time is limited
  - We cannot use heavy ansatz or error mitigation

Summary for my algorithm

- Ansatz
  - symmetry preserving ansatz with Given’s rotation gate
  - gradually increasing layer depth
  - adaptive placement of gates considering Hamiltonian

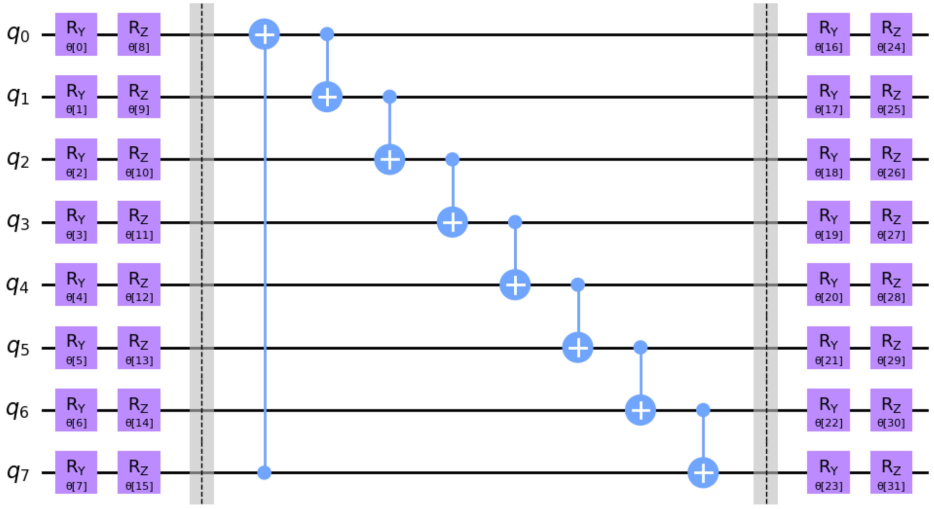
- Optimizer
  - compare several optimizers
    - Finally, I used SPSA

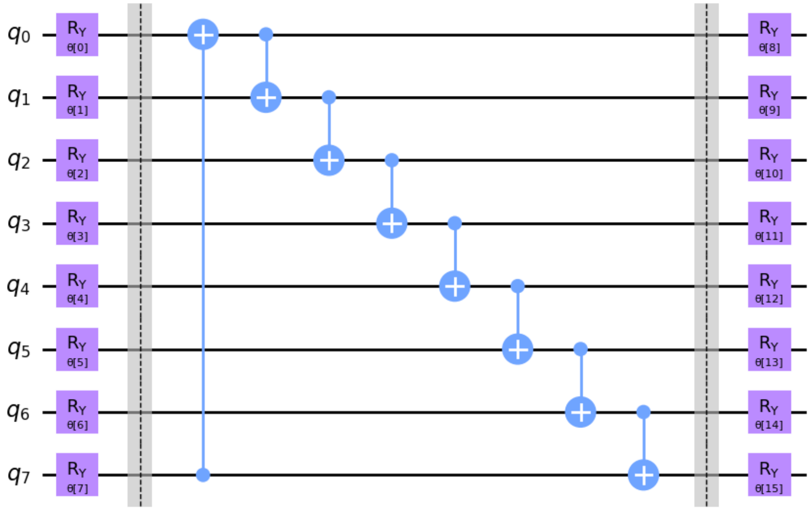
- Error mitigation
  - Partial Symmetry Enforcement
    - But removed in the final submission

Ansatz Selection

Hardware efficient vs Symmetry Preserving Ansatz

- First, I tried Hardware efficient ansatz (A Kandala et al., 2017)
  - RYRZ and RY only, CNOT and CZ



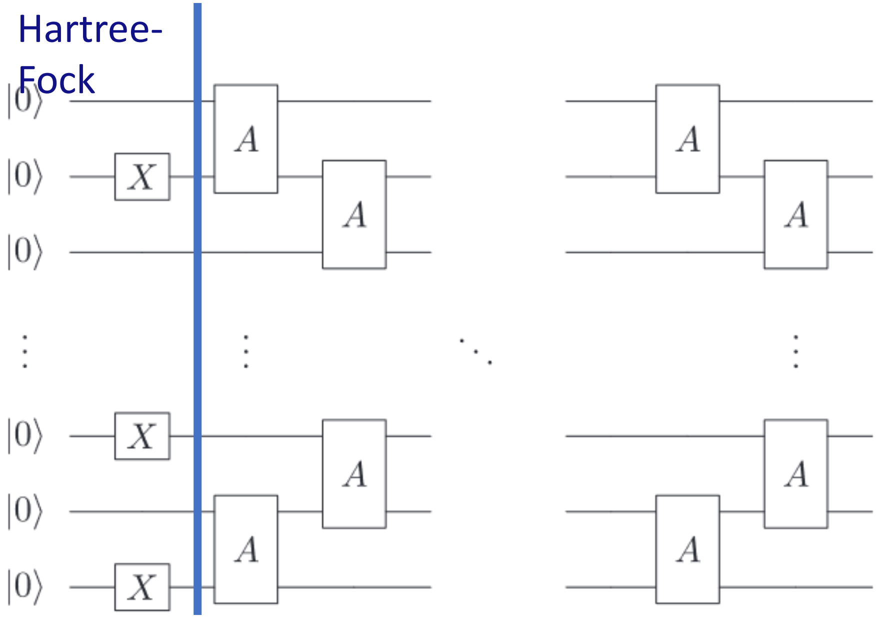


  - HW ansatz doesn't work well
    - score ≈ 1.3
    - HW ansatz doesn't constraint num of particles

Hardware efficient vs Symmetry Preserving Ansatz

- Symmetry Preserving Ansatz (Gard, Bryan T., et al, 2020)
  - Preserve particle numbers

$$A(\theta, \phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & e^{i\phi}\sin\theta & 0 \\ 0 & e^{-i\phi}\sin\theta & -\cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



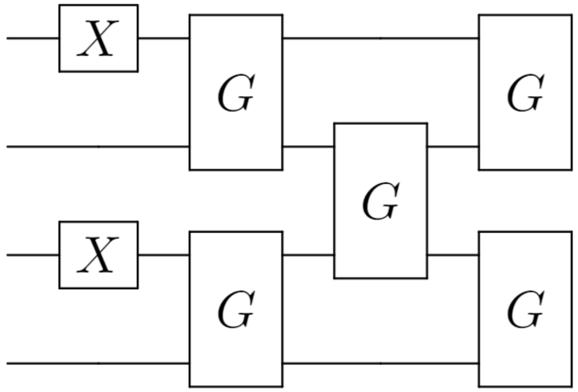
  - better than HW ansatz
  - fix $\phi = 0$ was better than learn ϕ
- The problem with A Gate
  - A(0, 0) is not identity matrix

Hardware efficient vs Symmetry Preserving Ansatz

- I used Givens rotation gate instead of A gate

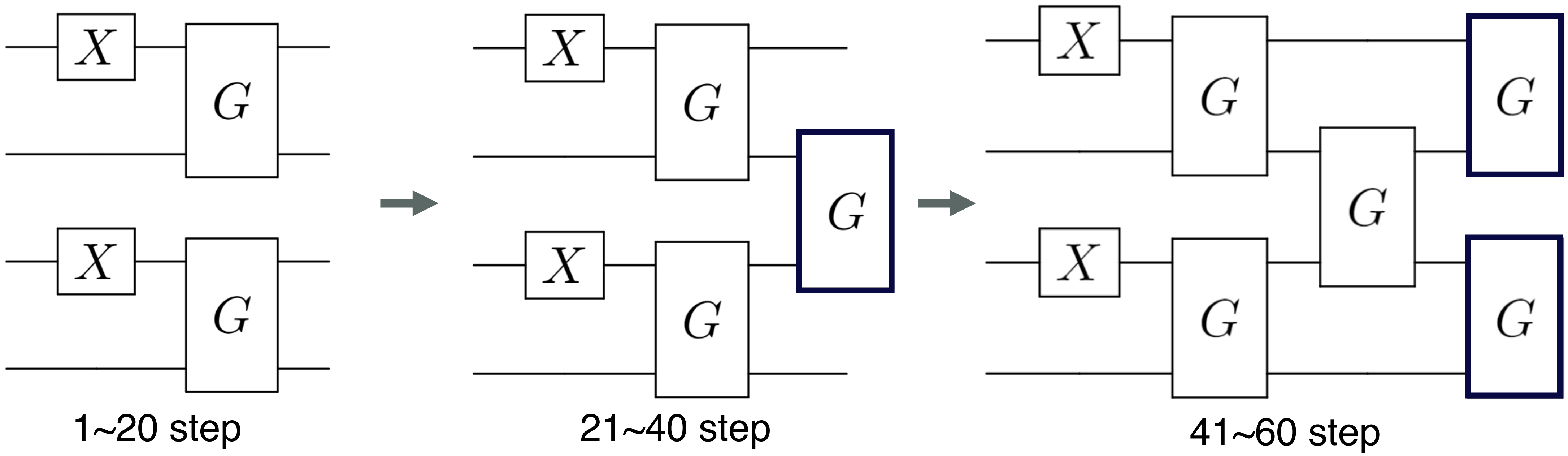
$$G(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad A(\theta, 0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & \sin\theta & -\cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- This gate needs one more CNOT gate
  - However, suits with layer-increasing approach



Gradually Increase Layer number

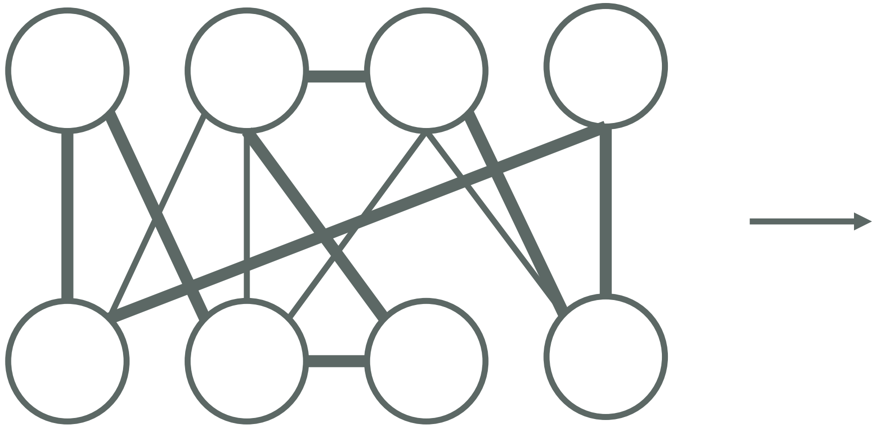
- Deeper layer: complex state ⇔ noise level is increased
- Increase Layer number with fixed step (20 step in SPSA)
- Achieve more stable optimization
- Optimize all parameters in each step



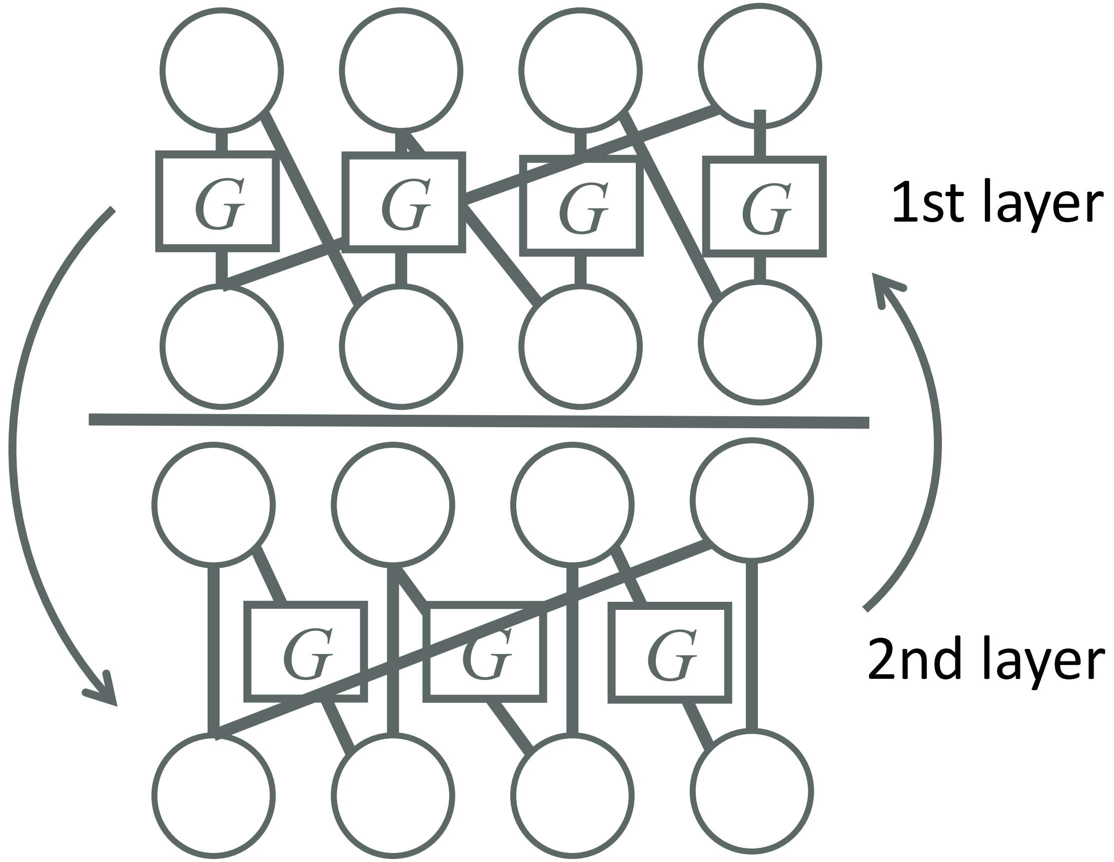
Adaptive Gate Placement Selection

- unitary coefficients are random
    - Highly connected qubits and less connected qubits
    - Place G gate between highly connected qubits

1. Extract the most connected loop



2. Determine where to place G gate



Optimizers

Optimizers

- I compared several optimizers

Gradient Based

SGD, Adam

Gradient evaluation
O(params)

Looks goods, but slow

Gradient Free

NFT (K. M. Nakanishi, 2019)
Attractive in noiseless
Difficulty in noisy situations

SPSA
Stable and fast

I chose **SPSA** because fast and stable in noisy situation

Implementation detail

- Super Conductor vs Ion Trap simulator
    - SC is10x-100x noisy, but 10000x faster than IT
    - ZNE for SC was difficult
        - Noise level was very high

    - I used IT type
        - Full connectivity
        - IT type doesn’t need to use heavy mitigation techniques

- Num of shots in each energy evaluation: 2000
    - I think 1000 is lower bound

Mitigation technique

- Partial Symmetry Enforcement (Barron, G. S., et al., 2021)
  - Mitigation technique for fermion operator
  - Check particle number is correct
    without additional measurement

Hamiltonian can be written as sum of Pauli

$$H = XIZI + ZIZI + XXXX + \cdots.$$

For Pauli string only contains Z or I (e.g. ZIZI, ZZZZ)
measurements don’t change particle number

Remove measurement if the number is wrong → leads to high std

Conclusion

- 3rd place: score 8.73437653
  - $|E_{true} - E_{est}| \cong 0.114$

Team	Score	Date
81ueman	100.86392407	2023/7/19
tebasaki	9.60380241	2023/7/23
xyzzy	9.18525584	2023/7/22
morim	8.73437653	2023/7/22

Thank you for listening!

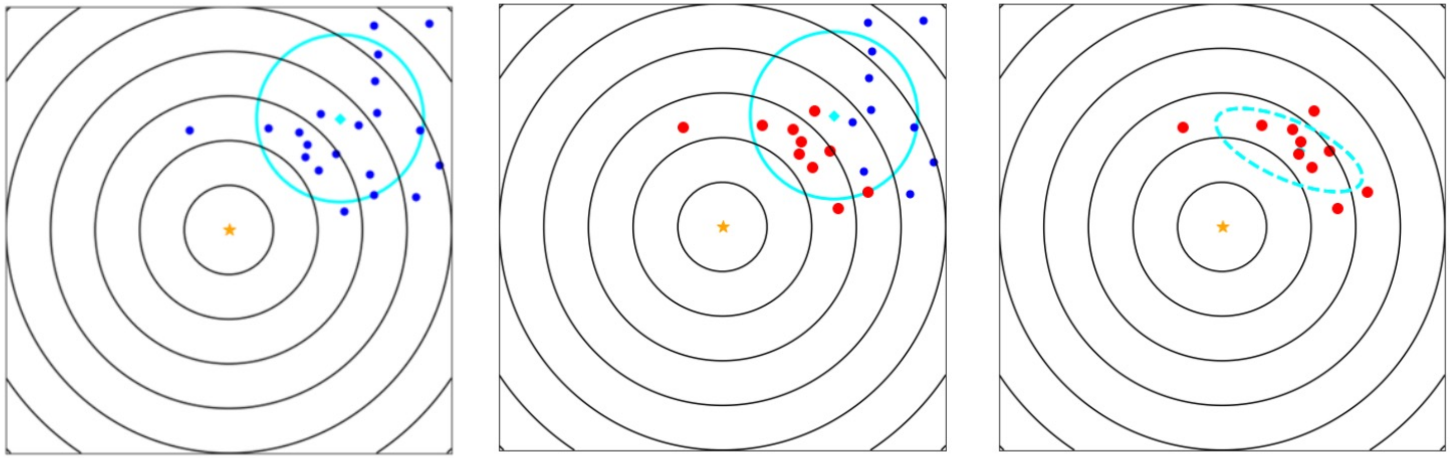
Appendix

Implementation detail

- Shot size for each eval: 2000

Optimizers

- First-Order SGD $$(\hat{g_n}(u_n))_i = \frac{J(u_n + c_n e_i) - J(u_n - c_n e_i)}{2c_n}$$
- SPSA $$(\hat{g_n}(u_n))_i = \frac{J(u_n + c_n \Delta_n) - J(u_n - c_n \Delta_n)}{2c_n (\Delta_n)_i}$$
- CMAES



Optimizers

- NFT

$$\mathcal{L}_j^{(n)}(\theta_j) = \sum_{k=1}^{K} w_k \langle \varphi_k | U_j^{(n)\dagger}(\theta_j) \mathcal{H}_k U_j^{(n)}(\theta_j) | \varphi_k \rangle$$
$$= a_{1j}^{(n)} \cos\left(\theta_j - a_{2j}^{(n)}\right) + a_{3j}^{(n)},$$