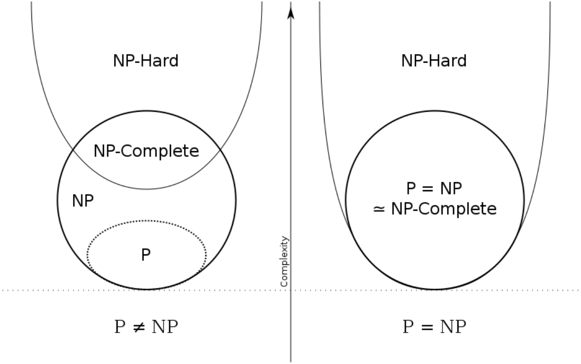
Using QAOA to solve the Clique-problem

Johannes Kroll

Technical University of Munich
Department of Informatics
Advanced Topics in Quantum Computing

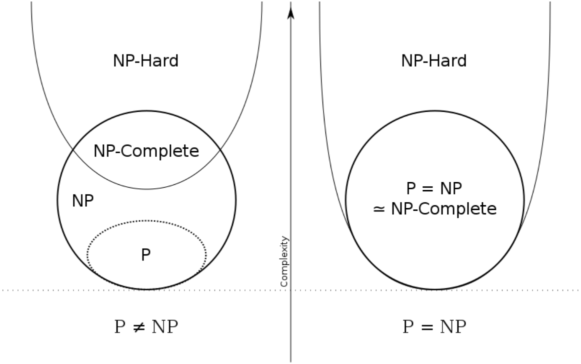
Munich, 30. June 2022

Motivation

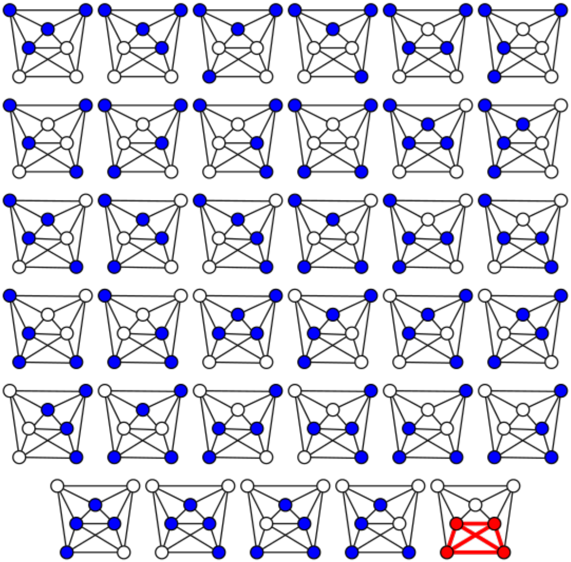


NP-Hard
NP-Complete
NP
P
Complexity
NP-Hard
P = NP
≃ NP-Complete
P ≠ NP
P = NP

Motivation



NP-Hard
NP-Complete
NP
P
Complexity
NP-Hard
P = NP
≃ NP-Complete
P ≠ NP
P = NP



Quantum Approximation Optimization Algorithm

Objective function on bit string z for m clauses:

$$C(z) = \sum_{a=1}^{m} C_a(z) \tag{1}$$

Quantum Approximation Optimization Algorithm

Objective function on bit string z for m clauses:

$$C(z) = \sum_{a=1}^{m} C_a(z) \tag{1}$$

Unitary operator $U(C, \gamma)$ depending on angle γ between 0 and 2π:

$$U(C, \gamma) = e^{-i\gamma C} = \prod_{a=1}^{m} e^{-i\gamma C_a} \tag{2}$$

Quantum Approximation Optimization Algorithm

Objective function on bit string z for m clauses:

$$C(z) = \sum_{a=1}^{m} C_a(z) \tag{1}$$

Unitary operator $U(C, \gamma)$ depending on angle γ between 0 and 2π:

$$U(C, \gamma) = e^{-i\gamma C} = \prod_{a=1}^{m} e^{-i\gamma C_a} \tag{2}$$

Operator B:

$$B = \sum_{j=1}^{n} \sigma_j^x \tag{3}$$

Quantum Approximation Optimization Algorithm

Objective function on bit string z for m clauses:

$$C(z) = \sum_{a=1}^{m} C_a(z) \tag{1}$$

Unitary operator $U(C,\gamma)$ depending on angle γ between 0 and 2π:

$$U(C,\gamma) = e^{-i\gamma C} = \prod_{a=1}^{m} e^{-i\gamma C_a} \tag{2}$$

Operator B:

$$B = \sum_{j=1}^{n} \sigma_j^x \tag{3}$$

Unitary operator $U(B,\beta)$ depending on angle β between 0 and π:

$$U(B,\beta) = e^{-i\beta B} = \prod_{j=1}^{m} e^{-i\beta\sigma_j^x} \tag{4}$$

Layer architecture

Initial state uniform superposition:

$$|\mathrm{s}\rangle = \frac{1}{\sqrt{2^{\mathrm{n}}}} \sum_{\mathrm{z}} |\mathrm{z}\rangle \tag{5}$$

Layer architecture

Initial state uniform superposition:

$$|\mathrm{s}\rangle = \frac{1}{\sqrt{2^{\mathrm{n}}}} \sum_{\mathrm{z}} |\mathrm{z}\rangle \tag{5}$$

p layers of operator with each different γ_{p} and β_{p}:

$$|\gamma, \beta\rangle = \mathrm{U}(\mathrm{B}, \beta_{\mathrm{p}})\mathrm{U}(\mathrm{C}, \gamma_{\mathrm{p}})...\mathrm{U}(\mathrm{B}, \beta_1)\mathrm{U}(\mathrm{C}, \gamma_1)|\mathrm{s}\rangle \tag{6}$$

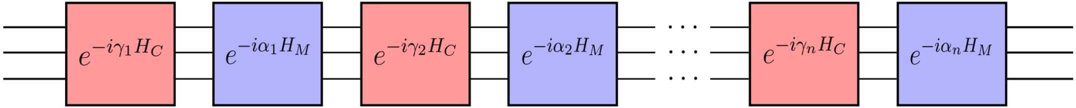
Layer architecture

Initial state uniform superposition:

$$|\mathrm{s}\rangle = \frac{1}{\sqrt{2^{\mathrm{n}}}} \sum_{\mathrm{z}} |\mathrm{z}\rangle \tag{5}$$

p layers of operator with each different γ_{p} and β_{p}:

$$|\gamma, \beta\rangle = \mathrm{U}(\mathrm{B}, \beta_{\mathrm{p}})\mathrm{U}(\mathrm{C}, \gamma_{\mathrm{p}})...\mathrm{U}(\mathrm{B}, \beta_1)\mathrm{U}(\mathrm{C}, \gamma_1)|\mathrm{s}\rangle \tag{6}$$



The Optimization part

F_p is the expectation of C in this state:

$$F_p(\gamma, \beta) = \langle \gamma, \beta | C | \gamma, \beta \rangle \tag{7}$$

The Optimization part

$\mathrm{F_p}$ is the expectation of C in this state:

$$\mathrm{F_p}(\gamma, \beta) = \langle \gamma, \beta | \mathrm{C} | \gamma, \beta \rangle \tag{7}$$

$\mathrm{M_p}$ in the maximum of $\mathrm{F_p}$ for angles γ and β:

$$\mathrm{M_p}(\gamma, \beta) = \max_{\gamma, \beta} \mathrm{F_p}(\gamma, \beta) \tag{8}$$

The Optimization part

F_p is the expectation of C in this state:

$$F_p(\gamma, \beta) = \langle \gamma, \beta | C | \gamma, \beta \rangle \tag{7}$$

M_p in the maximum of F_p for angles γ and β:

$$M_p(\gamma, \beta) = \max_{\gamma, \beta} F_p(\gamma, \beta) \tag{8}$$

Maximum of layer $p - 1$ is constrained by Maximum of layer p:

$$M_p \geq M_{p-1} \tag{9}$$

The Optimization part

F_p is the expectation of C in this state:

$$F_p(\gamma, \beta) = \langle \gamma, \beta | C | \gamma, \beta \rangle \tag{7}$$

M_p in the maximum of F_p for angles γ and β:

$$M_p(\gamma, \beta) = \max_{\gamma, \beta} F_p(\gamma, \beta) \tag{8}$$

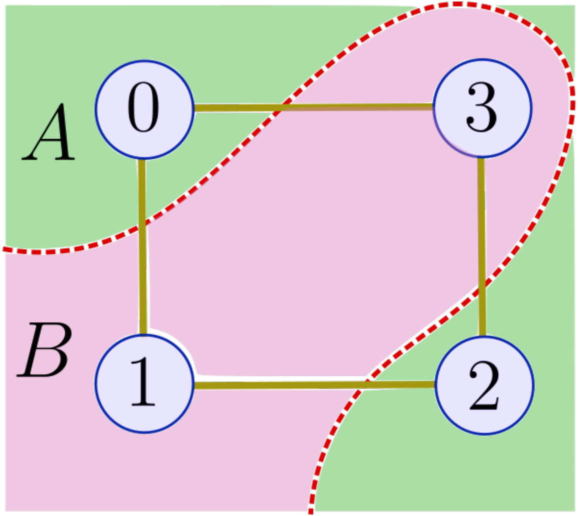
Maximum of layer $p - 1$ is constrained by Maximum of layer p:

$$M_p \geq M_{p-1} \tag{9}$$

With the number of layer p high enough we can find a good approximation:

$$\lim_{p \to \infty} M_p = \max_z C(z) \tag{10}$$

MaxCut



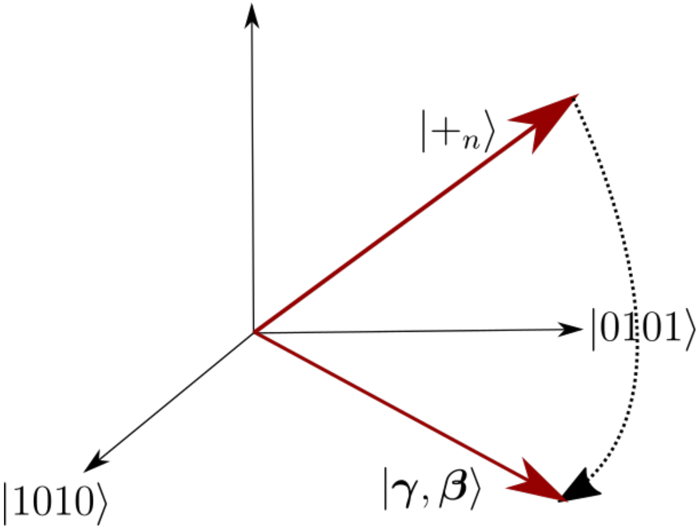
Hamiltonians:

$$H_{cost} = \frac{1}{2} \sum_{(i,j)\in E(G)} (Z_i Z_j - I) \tag{11}$$

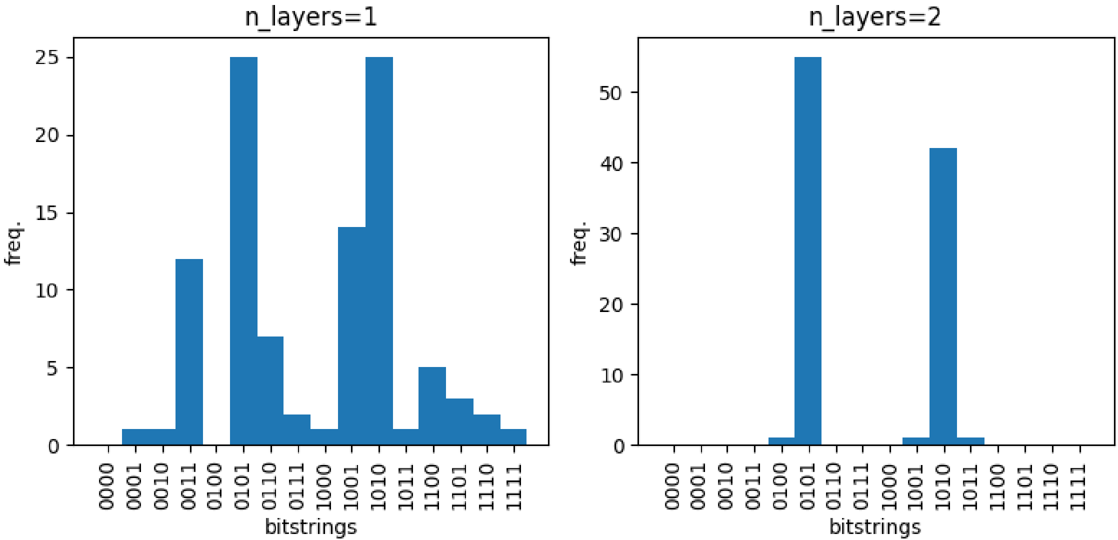
$$H_{mixer} = \sum_{i\in V(G)} X_i \tag{12}$$

Solution for this Graph,
represented by bit string $z = z_0 z_1 z_2 z_3$:
$z = 0101$
and
$z = 1010$

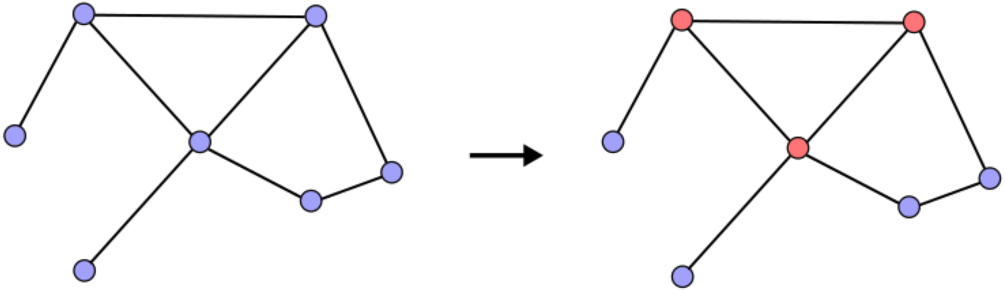
Optimal State on Bloch Sphere



MaxCut result for 1 and 2 layers



MaxClique

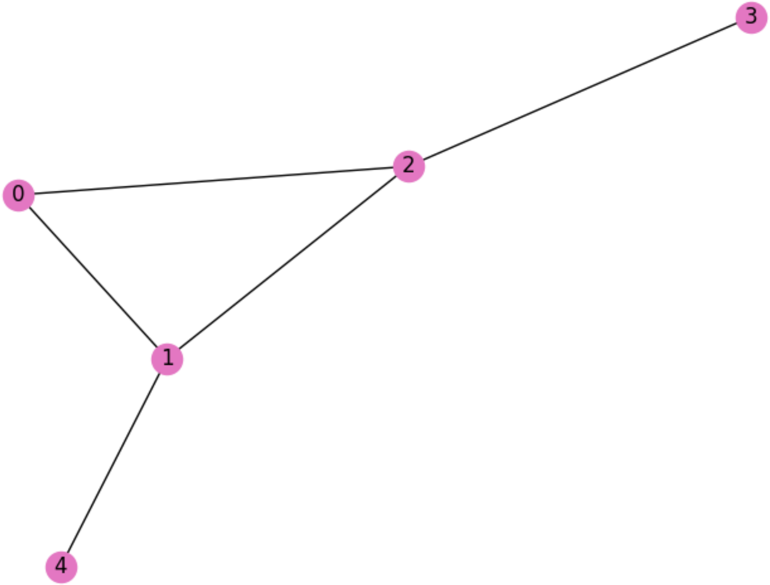


Hamiltionians

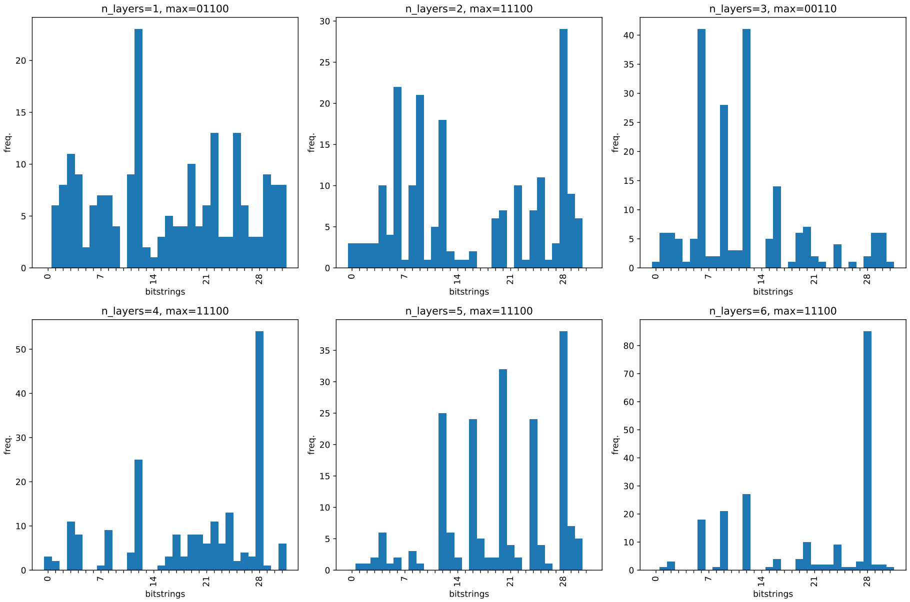
$$H_{\text{cost}} = 3 \sum_{(i,j)\in E(\overline{G})} (Z_i Z_j - Z_i - Z_j) + \sum_{i\in V(G)} Z_i \tag{13}$$

$$H_{\text{mixer}} = \sum_{i\in V(G)} X_i \tag{14}$$

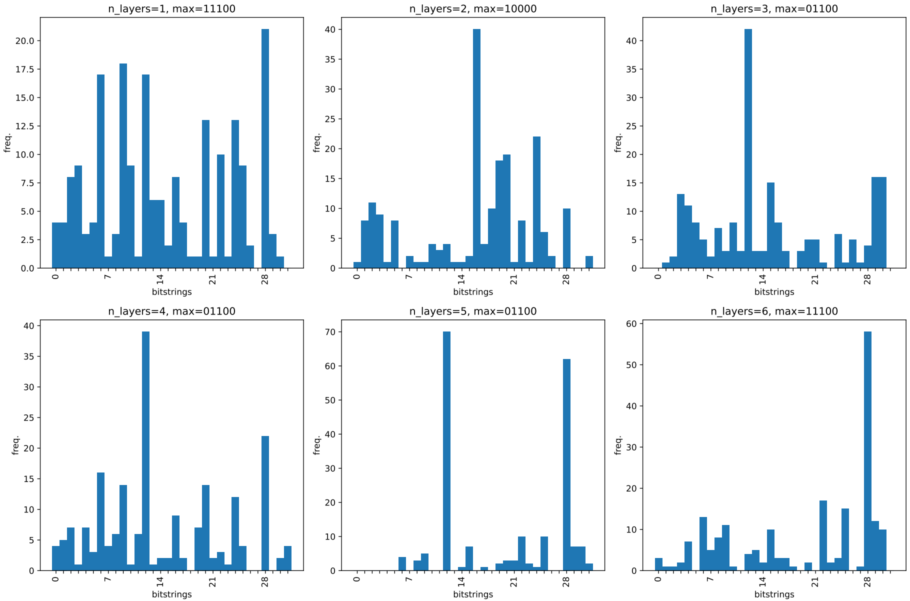
MaxClique example



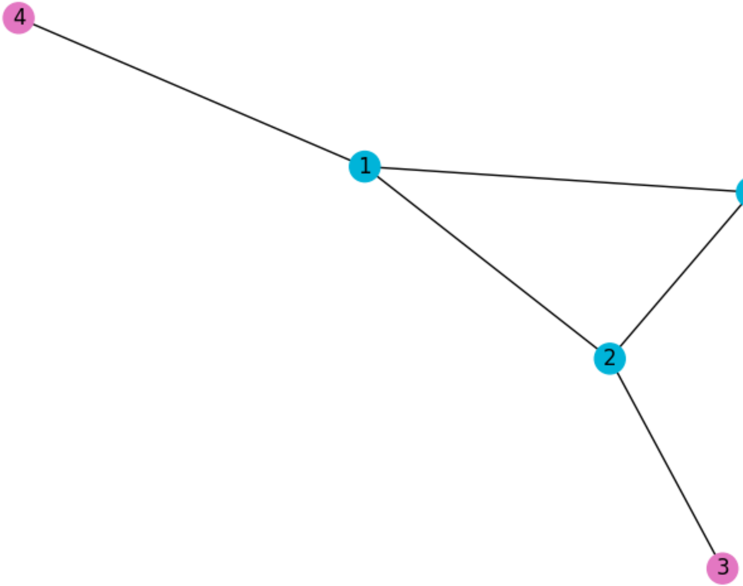
MaxClique result for example with different layers



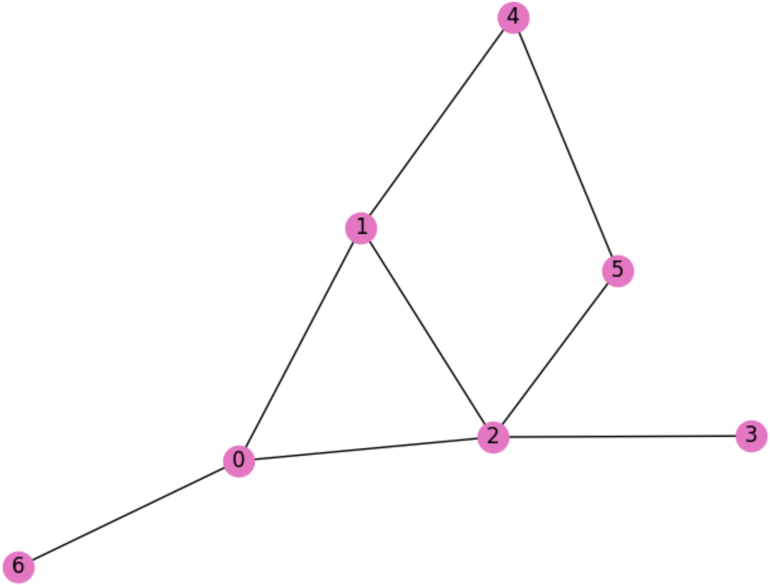
MaxClique result with different initial values



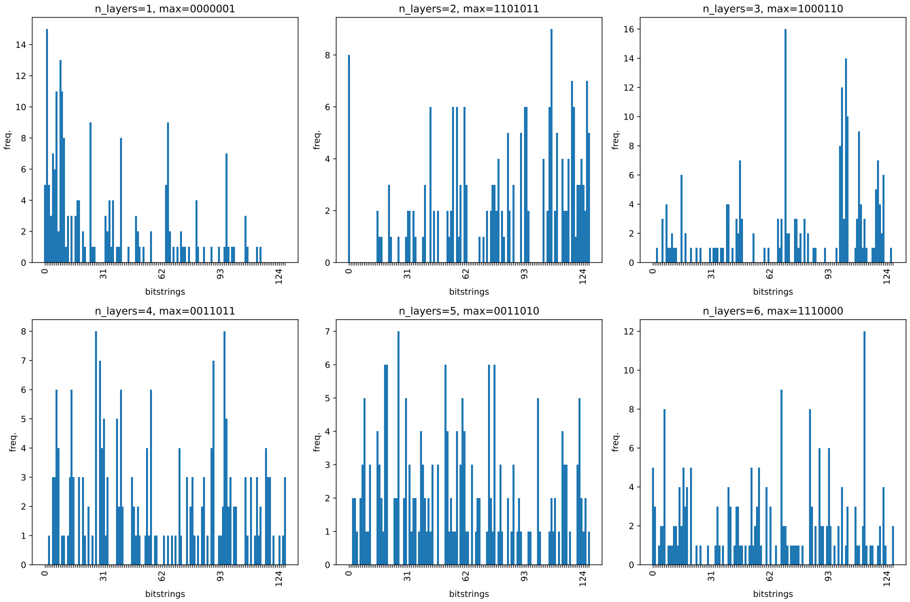
MaxClique example solution



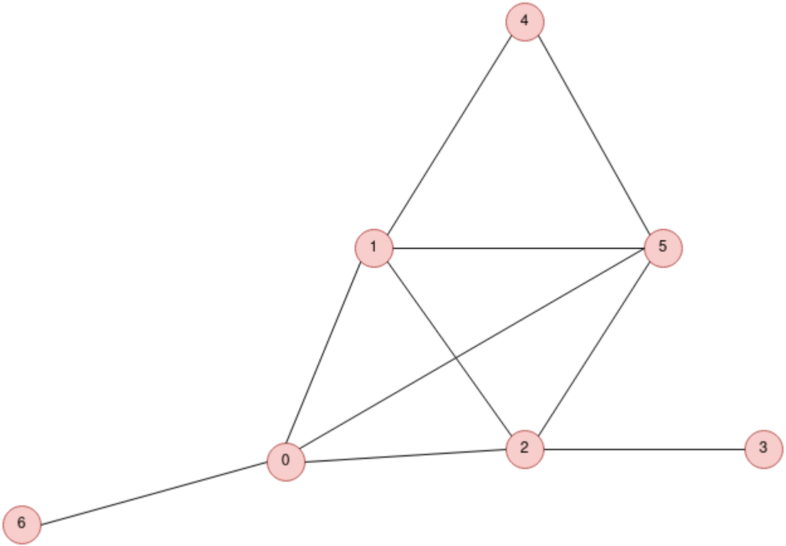
MaxClique large example



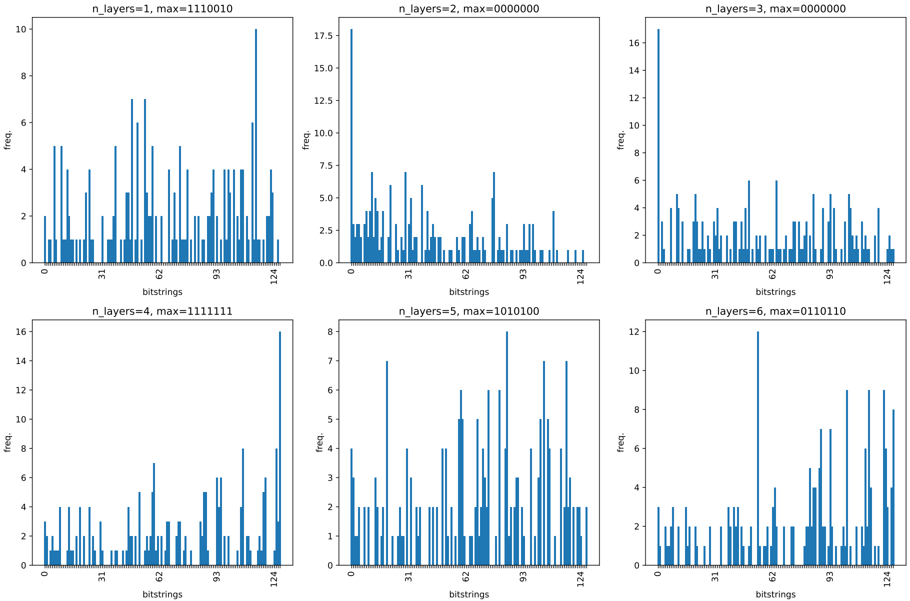
MaxClique result for large example with different layers



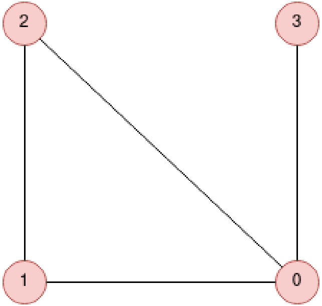
MaxClique large and complex example



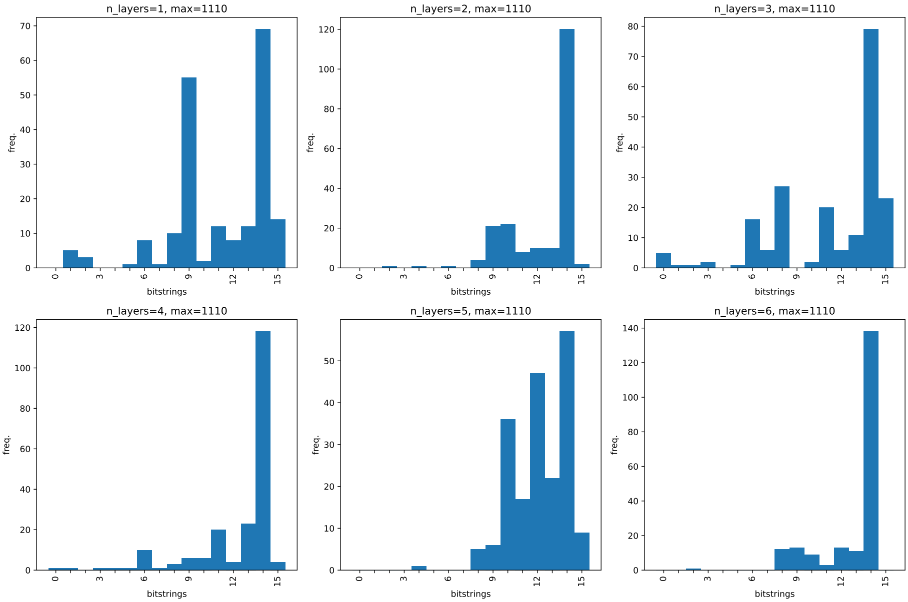
MaxClique result for large and complex example with different layers



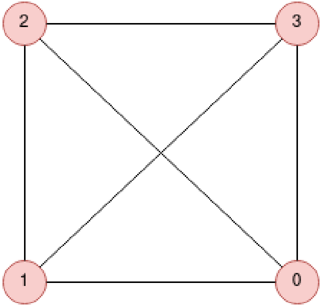
MaxClique small example



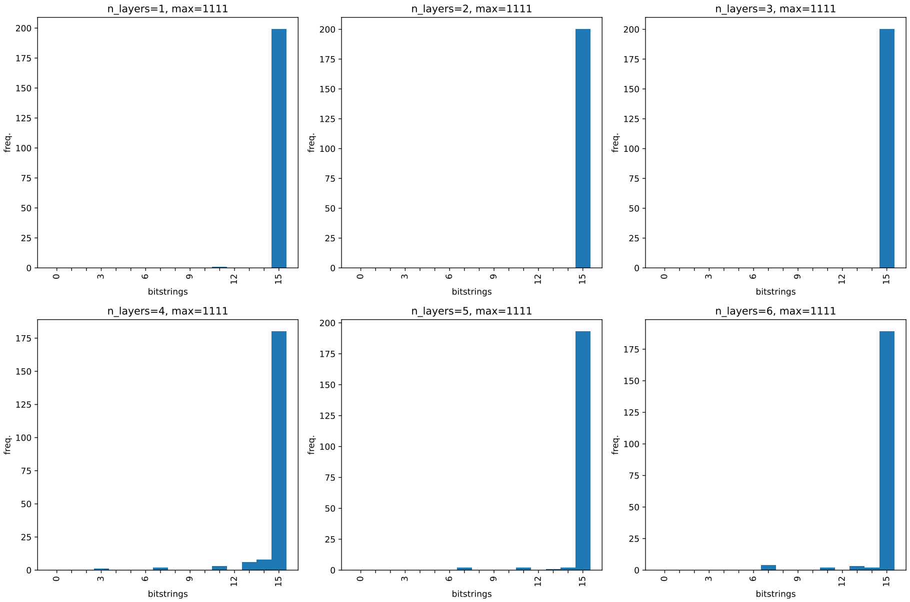
MaxClique result for small example with different layers



MaxClique simple example



MaxClique result for simple example with different layers



Conclusion

- QAOA uses multiple layers with each the angles γ and β

Conclusion

- QAOA uses multiple layers with each the angles γ and β
- Cost and Mixer Hamiltonian have to be defined for each problem

Conclusion

- QAOA uses multiple layers with each the angles γ and β
- Cost and Mixer Hamiltonian have to be defined for each problem
- Then Optimize the angles of each layer

Conclusion

- QAOA uses multiple layers with each the angles γ and β
- Cost and Mixer Hamiltonian have to be defined for each problem
- Then Optimize the angles of each layer
- Experiment with different optimizers, layers and optimization steps

Conclusion

- QAOA uses multiple layers with each the angles γ and β
- Cost and Mixer Hamiltonian have to be defined for each problem
- Then Optimize the angles of each layer
- Experiment with different optimizers, layers and optimization steps
- Initial value of the angles has a big impact on the approximation

Conclusion

- QAOA uses multiple layers with each the angles γ and β
- Cost and Mixer Hamiltonian have to be defined for each problem
- Then Optimize the angles of each layer
- Experiment with different optimizers, layers and optimization steps
- Initial value of the angles has a big impact on the approximation
- Further work: Find optimal parameters for MaxClique

Thank you for your attention!

Any questions?

References

Edward Farhi, Jeffrey Goldstone, and Sam Gutmann.
A quantum approximate optimization algorithm, 2014.

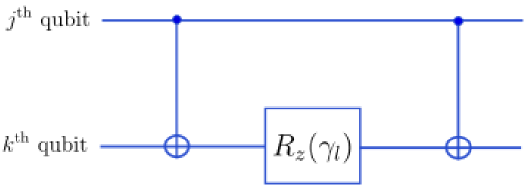
John Golden, Andreas Bärtschi, Stephan Eidenbenz, and Daniel O'Malley.
Evidence for super-polynomial advantage of qaoa over unstructured search, 2022.

Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor Rieffel, Davide Venturelli, and Rupak Biswas.
From the quantum approximate optimization algorithm to a quantum alternating operator ansatz.
Algorithms, 12(2):34, feb 2019.

Alicia B. Magann, Kenneth M. Rudinger, Matthew D. Grace, and Mohan Sarovar.
Feedback-based quantum optimization, 2021.

Gate representation MaxCut

$$e^{-i\gamma_l(1-\sigma_z^j\sigma_z^k)/2}$$



$$e^{-i\beta_l\sigma_x^j}$$