Quantum Optimization

Quantum Relax-and-Round (QRR) Algorithm

Eraraya Ricardo Muten

Technical University of Munich, Quantum Science & Technology

Mentor: Lilly Palackal

27 June 2024

“Extending relax-and-round combinatorial optimization solvers with quantum correlations” (PhysRevA.109.012429). Paper by Maxime Dupont and Bhuvanesh Sundar.

QAOA in a Nutshell (Refresher)

$$|\psi(\vec{\gamma},\vec{\beta})\rangle = \left(\prod_{k=1}^{p}\left[e^{-i\hat{H}_M\beta_k}e^{-i\hat{H}_C\gamma_k}\right]\right)|\psi_{\text{init}}\rangle$$

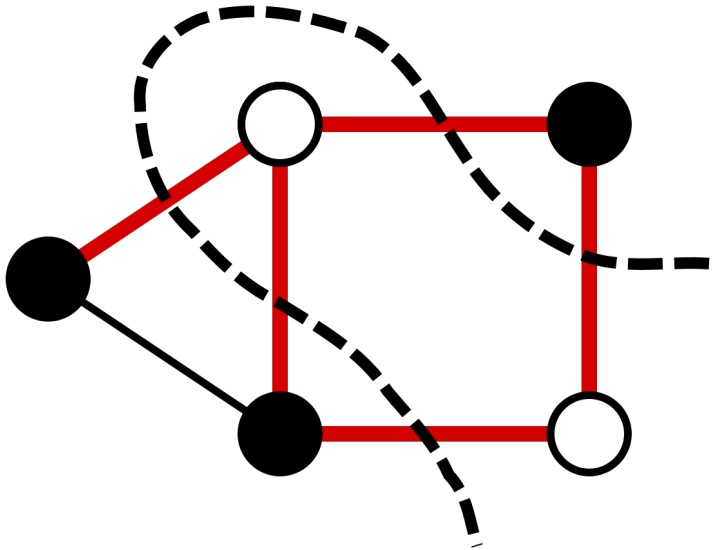
$$|\psi_{\text{init}}\rangle = |+\rangle^{\otimes N}$$

$$\hat{H}_M = \hat{X}^{\otimes N}$$

Minimize $\langle\hat{H}_C\rangle$ with respect to the $(\vec{\gamma},\vec{\beta})$ parameters.

MaxCut in a Nutshell (Refresher)

Given a graph, find a way to separate the vertices into two groups so that it maximizes the total number of edges that are “cut” (two vertices of that edge belong to different group).



$$C(z) = \sum_{\alpha=1}^{m} C_\alpha(z)$$

$$\hat{H}_C = -\frac{1}{2}\sum_{i,j} w_{i,j}\left(I - \hat{Z}_i\hat{Z}_j\right)$$

Motivations for the QRR algorithm

- A desirable quantum algorithm for current NISQ era: low circuit depth but high performance.
- QAOA with infinite depth converges to the optimal solution, but not very good at low depth p.
- Ideas to improve QAOA even at low depth?
  - Difficulty in binary optimization comes from the integer restriction.
  - One idea from approximate classical algorithms is to “relax” the restriction from -1 or 1 to [-1, 1].
  - Can we borrow this principle to improve QAOA?

Idea Proposed by the Paper

- “Quantum” version of the classical relax-and-round (RR).
- QRR at $p = 1$ at least as good as the classical RR for certain problem instances (given both as mathematical derivation and numerical results in the paper).

QRR Algorithm Step-by-Step

1. Optimize a regular QAOA circuit at depth p (low, e.g., $p = 1$), with the cost function $C(z)$ as the cost for the QAOA.

2. Find the 2-point correlation matrix by sampling the solution from the optimized QAOA circuit.

$$\boldsymbol{Z_{ij}} = (\delta_{ij} - 1)\langle \hat{Z}_i \hat{Z}_j \rangle$$

$\mathbf{Z}$ = the correlation matrix
δ_{ij} = Kronecker delta ($= 1$ if $i = j$, $= 0$ otherwise)

$\langle \hat{Z}_i \hat{Z}_j \rangle$ = measurement expectation of Pauli-Z on the i-th and j-th qubits.

QRR Algorithm Step-by-Step

3. “Relax” step:
   Find the set of eigenvectors $\{z\}$ of the correlation matrix $\mathbf{Z}$.

4. “Round” step:
   Round the eigenvectors entrywise based on their sign.
   If not negative = +1, if negative = -1
   Also, do the other way around to cover in case the problem has non-degenerate solution.
   If negative = +1, if not negative = -1

5. Check which eigenvector from all the rounded eigenvectors gives the best (lowest) cost $C(z)$.

6. This best eigenvector is the solution output from the algorithm.

Notes on classical RR:
Step-by-step algorithm of a classical RR is similar to QRR, but the classical version uses the adjacency matrix $\mathbf{W}$ instead of correlation matrix $\mathbf{Z}$.

Intuitions about the QRR

- QRR potentially more powerful since $\mathbf{Z}$ potentially contains more non-trivial information than $\mathbf{W}$ (the one used in classical RR).

- At $p = \infty$ limit (optimal QAOA), the $\mathbf{Z}$ matrix will have the optimal solution $\pm z_{opt}/\sqrt{N}$ as its eigenvector.
  - The “rounding” step will round this eigenvector into $\pm z_{opt}$ (removing the scaling of $\sqrt{N}$).
  - As the $\mathbf{Z}$ matrix improves with deeper QAOA layer, the resulting eigenvectors from $\mathbf{Z}$ might comes closer and closer to $\pm z_{opt}/\sqrt{N}$.

QRR Robustness to Depolarizing Noise

- The system got mixed under depolarizing noise:

$$\hat{\rho}_F = F|\Psi\rangle\langle\Psi| + (1-F)\hat{I}/2^N$$

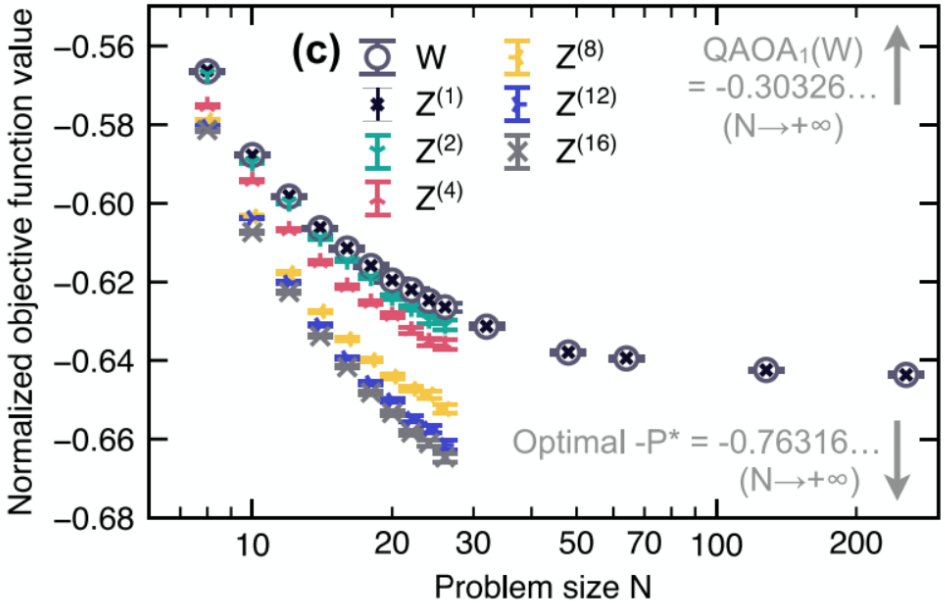
- The expectation value of the two-point correlation:

$$Z_{ij}^{(F)} = (\delta_{ij} - 1)\mathrm{tr}(\hat{\rho}_F \hat{Z}_i \hat{Z}_j) = F Z_{ij}^{(F=0)}$$

- Just a rescaling of the noiseless $\mathbf{Z}$ matrix by a parameter F.
- The rounding step of eigenvectors won't be affected since scaling doesn't affect rounding → will retrieve the same solution.

Numerical result 1: Sherrington-Kirkpatrick (SK) Spin Glasses Graphs

- Random weight of ±1.
- Finite N vertices.
- Each data point is averaged over 10^4 to 10^5 independent random problem graphs.
- Cost value is normalized by $2N^{3/2}$.



- A systematic improvement as p increases vs vanilla QAOA and classical relax-and-round (using $\mathbf{W}$ matrix).
- Results from vanilla QAOA with $p = 1$ and the correct optimal solution P is outside the range of the plots.

Numerical result 2: Comparison to Goemans-Williamson (GW)

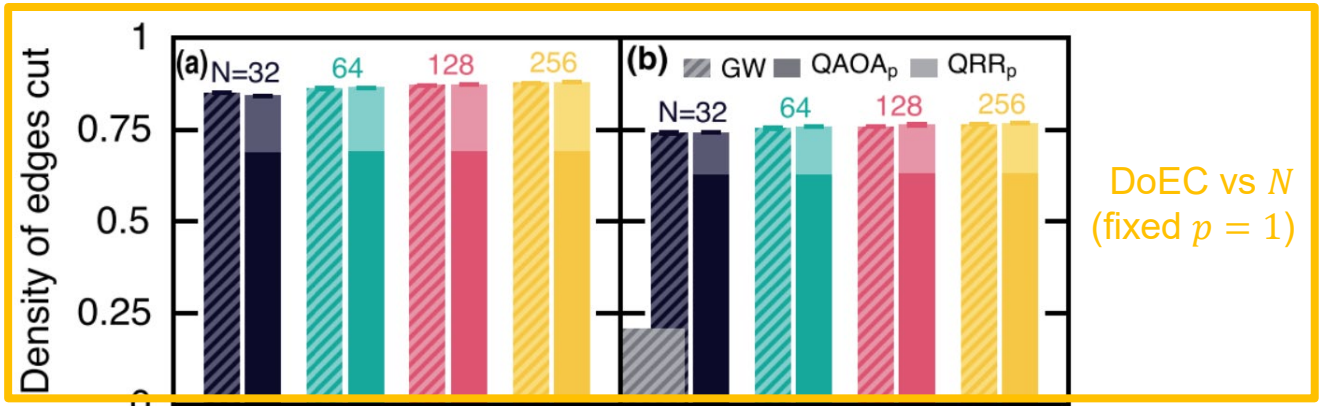
- Problems considered: 3REG and NWS graphs.

- GW algo vs $\mathrm{QRR}_{\mathrm{GW}}$ (QRR adaptation as a “quantum” version of GW algorithm on MaxCut, using the $\mathbf{Z}$ matrix instead of $\mathbf{W}$) vs vanilla QAOA.

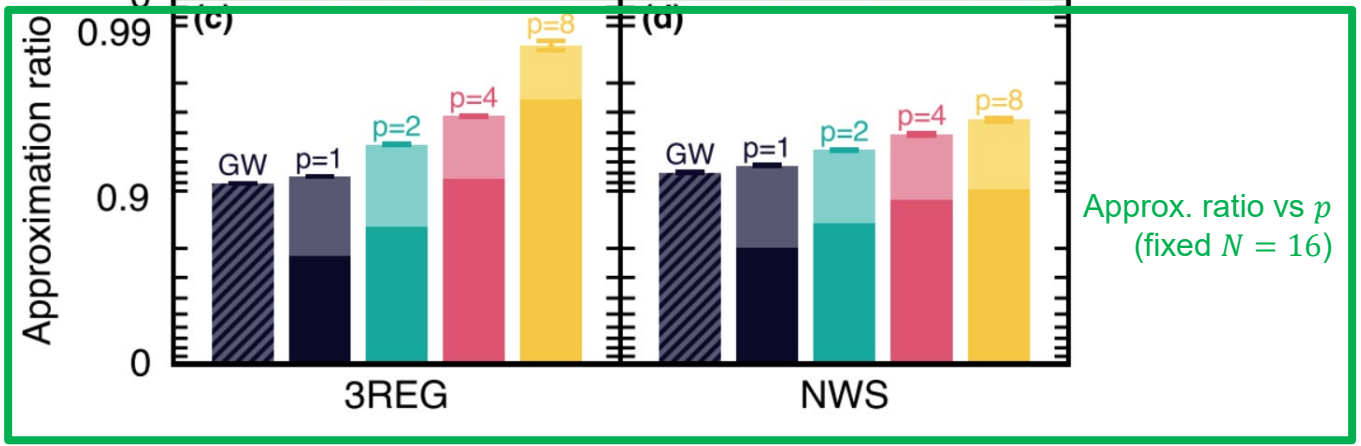
- Approximation Ratio: # of edges cut / the maximum edges can be cut (theoretical optimum value)
- Density of Edges Cut: # of edges cut / total edges

- Data point is averaged from between 10^3 to 10^4 independent random problem graphs.

Numerical result 2: Comparison to Goemans-Williamson (GW)



DoEC vs N (fixed $p = 1$)

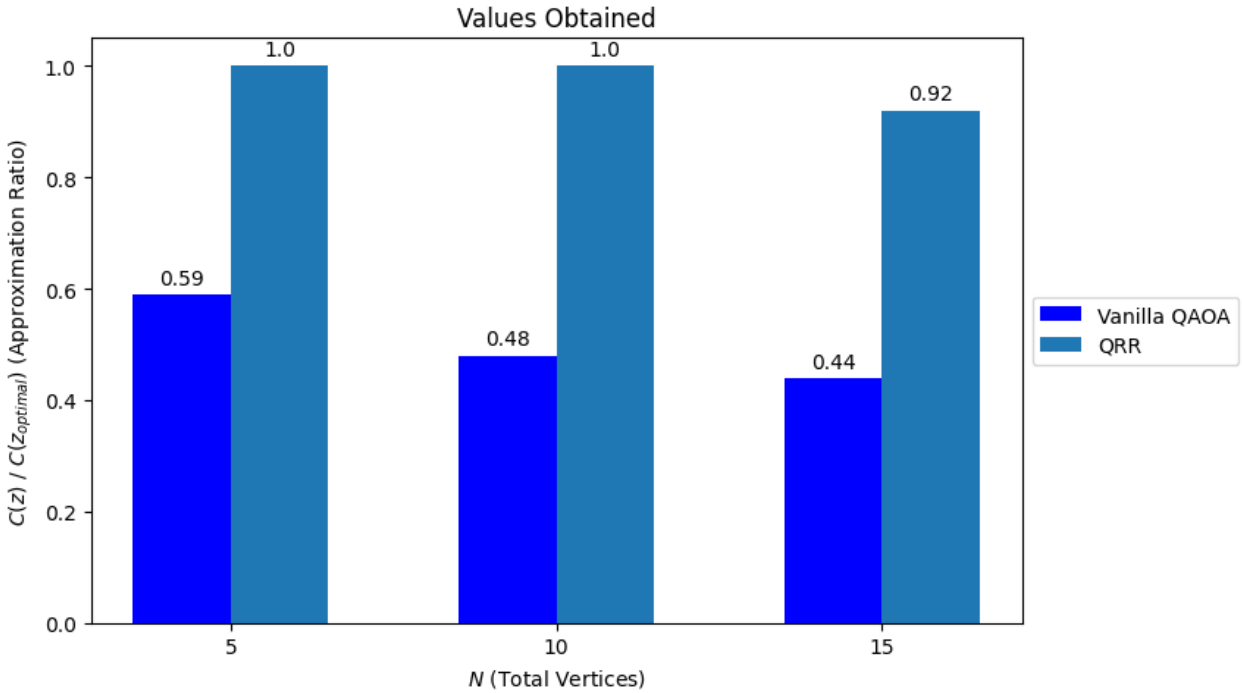


Approx. ratio vs p (fixed $N = 16$)

- GW performance guarantee is graph agnostic, QRR_{GW} agnostic guarantee is still open question.
- But numerical results show that QRR_{GW} surpassed GW on certain problem graphs for $p > 1$.

Self Prototype 1: MaxCut of Random Graphs

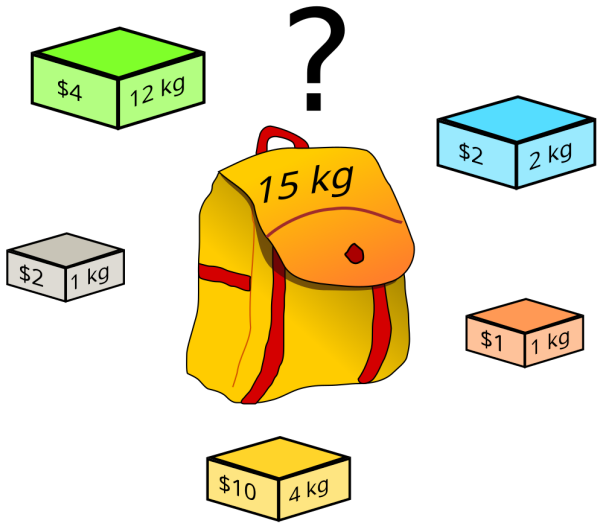
- 1 layer of QAOA, statevector simulator, bitstrings are then sampled, 1000 shots.
- N = 5, 10, and 15 vertices (5, 10, and 15 qubits).
- Averaged from 10 random graphs.
- Normalized with respect to the optimal value (found with classical optimization tool Gurobi, assumed to be optimal).
- Each QAOA result is obtained by averaging all the bitstrings, while the QRR result is obtained from the single best sign-rounded eigenvector.



Self Prototype 2: Knapsack Problems

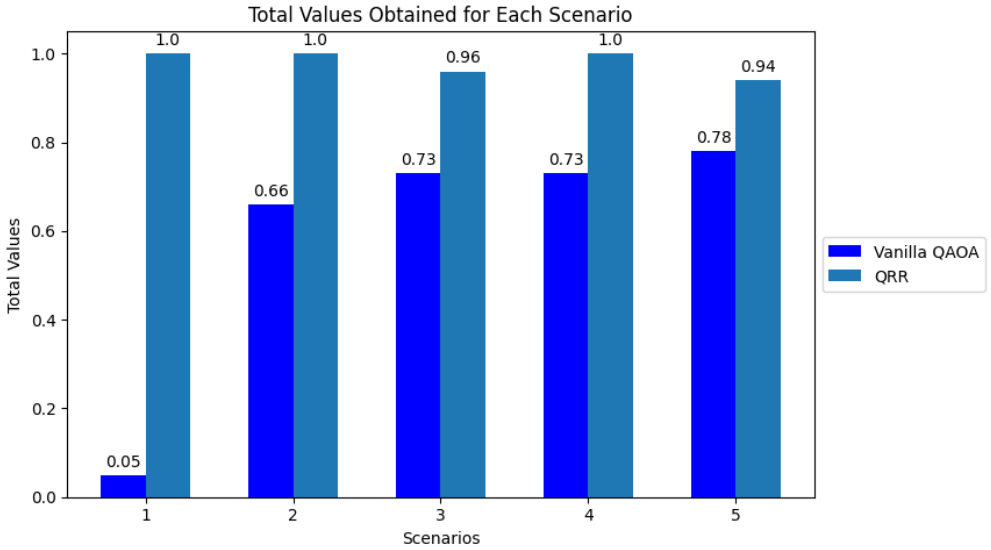
- Knapsack problem: Given a list of items that each has certain value and weight, find the optimal configuration that gives the highest total values on which items to select under a constraint of a maximum weight.

- Unique main difference vs MaxCut: MaxCut has no constraint, while Knapsack is an optimization problem with a constraint.



Self Prototype 2: Knapsack Problems

- 200 layers of QAOA, 10000 shots (statevector simulator, bitstrings are then sampled).
- QAOA parameters aren't optimized, but mimic the annealing process of adiabatic quantum computation to save runtime for this demo.
- 7-12 qubits.
- Each QAOA result is obtained by averaging all the feasible bitstrings, while the QRR result is obtained from the single best feasible sign-rounded eigenvector.
- Normalized with respect to the optimal value.



- QRR almost always found the optimal solution (at least for the scenarios tried).
- The vanilla QAOA results are from the averages of all feasible bitstrings.

Closing Remarks

- QRR is a very interesting way to utilize classical optimization tools to improve the QAOA.

- One important note is that QRR is “just” post-processing steps on top of QAOA, which means:
  - It’s very computationally cheap (finding eigenvectors is ~$O(N^3)$).
  - It can be easily integrated into other quantum optimization protocols. For example VQE and QAOA variants (ADAPT-QAOA, QIRO, etc.).
  - If the bitstrings sampled from the QAOA is “bad”, then there is no guarantee that QRR will improve significantly (or at all).

Thank you for listening! Any questions?

https://github.com/ericardomuten/qrr