Classical variational simulation of the Quantum Approximate Optimization Algorithm

Matija Medvidović
Center for Computational Quantum Physics, Flatiron Institute, 162 5th Avenue, New York, NY 10010, USA and Department of Physics, Columbia University, New York 10027, USA

Giuseppe Carleo
Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland

Supplementary Information

Supplementary Notes

I. EXACT P=1 FORMULA

In this section we present the exact MaxCut QAOA cost formula used for quantifying the accuracy of our model result.

Theorem 1 *For an arbitrary graph G, the QAOA cost function for the MaxCut problem given in Eq. 3 (main text) takes the form*

$$C(\gamma,\beta)=\frac{1}{2}\sum_{\langle k,l\rangle}\left[\sin(4\beta)\sin(2\gamma)\left(\cos^{q_k}(2\gamma)+\cos^{q_l}(2\gamma)\right)+\sin^2(2\beta)\cos^{q_k+q_l-2\Delta_{kl}}(2\gamma)(1-\cos^{\Delta_{kl}}(4\gamma))\right] \tag{1}$$

at $p=1$. Here, q_k+1 and q_l+1 are degrees of vertices k and l and Δ_{kl} is the number of common neighbors between those vertices.

Proof: The proof repeats much of what has already been done in [1]. We begin by expressing the density operator associated with the $|+\rangle$ state as

$$\rho_0=|+\rangle\langle+|=\prod_i\frac{\mathbb{1}_i+X_i}{2}\ . \tag{2}$$

Then, we can express the MaxCut QAOA cost function at $p=1$ as

$$\langle\gamma,\beta|\,\mathcal{C}\,|\gamma,\beta\rangle=\sum_{\langle k,l\rangle}\mathrm{Tr}\left[\rho_0U_C^\dagger(\gamma)U_B^\dagger(\beta)Z_kZ_lU_B(\beta)U_C(\gamma)\right] \tag{3}$$

In what follows, we will make repeated use of the following identities:

$$e^{i\beta X}Ze^{-i\beta X}=\cos(2\beta)\ Z+\sin(2\beta)\ Y \tag{4}$$

$$e^{-i\gamma Z\otimes Z}=\cos\gamma-i\sin\gamma\ Z\otimes Z \tag{5}$$

$$e^{-i\gamma Z}\ Y=Y\ e^{i\gamma Z}\ . \tag{6}$$

The innermost product can easily be expanded using Eq. 4 twice:

$$\begin{aligned}U_B^\dagger(\beta)Z_kZ_lU_B(\beta)&=U_B^\dagger(\beta)Z_kU_B(\beta)U_B^\dagger(\beta)Z_lU_B(\beta)=\\&=\cos^2(2\beta)\ Z_kZ_l+\frac{1}{2}\sin(4\beta)\left(Y_kZ_l+Y_lZ_k\right)+\cos^2(2\beta)\ Y_kY_l\ .\end{aligned} \tag{7}$$

The first term vanishes when averaged against ρ_0. The second and third need to be treated separately. First we look at expressions of the form:

$$\mathrm{Tr}\left[\rho_0\ e^{i\gamma\mathcal{C}}\ Y_k Z_l\ e^{-i\gamma\mathcal{C}}\right] = \mathrm{Tr}\left[\rho_0\ Y_k Z_l \prod_{j\in N(k)} e^{-2i\gamma Z_k Z_j}\right] = \mathrm{Tr}\left[\rho_0\ Y_k Z_l \prod_{j\in N(k)} (\cos 2\gamma - i\sin 2\gamma\ Z_k Z_j)\right], \quad (8)$$

where we denoted the set of all neighbors of node k by $N(k)$, used Eq. 6 and Eq. 5. In Eq. 8, tracing out Paulis not associated to either node k or its neighbors is trivial and produces a factor of unity. Furthermore, tracing out immediate neighbors of k other than l produces a factor of $\cos^{q_k}(2\gamma)$ where q_k+1 is the degree of node k. Keeping those factors aside, we are left with:

$$\mathrm{Tr}_{kl}\left[\frac{\mathbb{1}_k + X_k}{2}\frac{\mathbb{1}_l + X_l}{2}\ Y_k Z_l\,(\cos 2\gamma - i\sin 2\gamma\ Z_k Z_l)\right] = \sin 2\gamma\ . \quad (9)$$

Therefore, the final contribution from the second term in Eq. 7 is

$$\frac{1}{2}\sin(4\beta)\,\mathrm{Tr}\left[\rho_0\ e^{i\gamma\mathcal{C}}\,(Y_k Z_l + Y_l Z_k)\ \ e^{-i\gamma\mathcal{C}}\right] = \frac{1}{2}\sin(2\gamma)\sin(4\beta)\left[\cos^{q_k}(2\gamma) + \cos^{q_l}(2\gamma)\right] \quad (10)$$

Looking at the last factor in Eq. 7, we compute:

$$\begin{aligned}\mathrm{Tr}\left[\rho_0\ e^{i\gamma\mathcal{C}} Y_k Y_l e^{-i\gamma\mathcal{C}}\right] &= \mathrm{Tr}\left[\rho_0\ Y_k Y_l \left(\prod_{\substack{j\in N(k)\\ j\neq l}} e^{-i\gamma Z_k Z_j}\right)\left(\prod_{\substack{i\in N(l)\\ i\neq k}} e^{-i\gamma Z_k Z_j}\right) e^{i\gamma Z_k Z_l} e^{-i\gamma\mathcal{C}}\right] = \\ &= \mathrm{Tr}\left[\rho_0\ Y_k Y_l \left(\prod_{\substack{j\in N(k)\\ j\neq l}} e^{-2i\gamma Z_k Z_j}\right)\left(\prod_{\substack{i\in N(l)\\ i\neq k}} e^{-2i\gamma Z_k Z_j}\right)\right],\end{aligned} \quad (11)$$

where we separate the factor in $\mathcal{C}$ corresponding to the edge (k,l) out before using Eq. 6. In the second equality, the factor corresponding to the edge (k,l) cancels and others are absorbed into product expressions.

Like before, operators corresponding to nodes other than k or l or their neighbors can be traced out trivially. The next step is to trace out Paulis that are neighbors of k but not l. Each of those contributes with a factor of $\cos(2\gamma)$ so we get a factor of $\cos^{q_k-\Delta_{kl}}(2\gamma)$ for node k and a corresponding factor for node l resulting in $\cos^{q_k+q_l-2\Delta_{kl}}(2\gamma)$. Here, we denoted the number of common neighbors of k and l by Δ_{kl} and label their set by $CN(k,l)$ in what remains of Eq. 11:

$$\mathrm{Tr}\left[\rho_0\ e^{i\gamma\mathcal{C}} Y_k Y_l e^{-i\gamma\mathcal{C}}\right] = \cos^{q_k+q_l-2\Delta_{kl}}(2\gamma)\ \ \mathrm{Tr}\left[\rho_0'\ Y_k Y_l \prod_{i\in CN(k,l)} \left(e^{-2i\gamma Z_k Z_i} e^{-2i\gamma Z_l Z_i}\right)\right], \quad (12)$$

where ρ_0' is the reduced density operator. Using Eq. 5, one can trace out nodes in $CN(k,l)$ to obtain a factor of $(\cos^2 2\gamma + \sin^2 2\gamma\ Z_k Z_l)^{\Delta_{kl}}$. Finally, we expand that expression using the binomial theorem and interchange the trace operation with the sum. The trace in Eq. 12 becomes

$$\cos^{2\Delta_{kl}}(2\gamma) \sum_{\substack{n=0\\ \text{odd } n}}^{\Delta_{kl}} \binom{\Delta_{kl}}{n} \tan^{2n}(2\gamma) = \frac{1}{2}\left(1 - \cos^{\Delta_{kl}}(4\gamma)\right) \quad (13)$$

Putting all of the ingredients together, we have:

$$C(\gamma,\beta) = \frac{1}{2}\sum_{\langle k,l\rangle}\left[\sin(4\beta)\sin(2\gamma)\left(\cos^{q_k}(2\gamma) + \cos^{q_l}(2\gamma)\right) + \sin^2(2\beta)\cos^{q_k+q_l-2\Delta_{kl}}(2\gamma)(1-\cos^{\Delta_{kl}}(4\gamma))\right] \quad (14)$$

II. ALGORITHM FOR STOCHASTIC GATE APPLICATION

Algorithm 1: Target state approximation with an RBM.

Input: Initial parameters θ, target state ϕ, target fidelity tolerance `tol`, learning rate η
Result: Parameters $\theta' = \{\mathbf{a}', \mathbf{b}', W'\}$ such that $|\psi_{\theta'}\rangle \approx |\phi\rangle$
$\partial_k \ln F \leftarrow 0$; $S_{kl} \leftarrow 0$; $\theta' \leftarrow \theta$;
while $F < 1 - \texttt{tol}$ **do**
  Generate samples $\mathcal{B}_\psi \sim |\psi_{\theta'}|^2$ and $\mathcal{B}_\phi \sim |\phi|^2$ using MCMC;
  foreach $\mathcal{B}$ *in* $\mathcal{B}_\psi$ **do** evaluate and store $\psi_{\theta'}(\mathcal{B})$, $\phi(\mathcal{B})$ and $\mathcal{O}_k(\mathcal{B})$ for all parameters ;
  foreach $\mathcal{B}$ *in* $\mathcal{B}_\phi$ **do** evaluate and store $\psi_{\theta'}(\mathcal{B})$ and $\phi(\mathcal{B})$;
  evaluate and set $F \leftarrow \mathrm{Re}\left\{ \left\langle \frac{\phi(\mathcal{B})}{\psi(\mathcal{B})} \right\rangle_{\mathcal{B}\sim\mathcal{B}_\psi} \left\langle \frac{\psi(\mathcal{B})}{\phi(\mathcal{B})} \right\rangle_{\mathcal{B}\sim\mathcal{B}_\phi} \right\}$;
  foreach k **do** evaluate and set $\partial_k \ln F \leftarrow \langle \mathcal{O}_k^* \rangle_{\mathcal{B}\sim\mathcal{B}_\psi} - \left\langle \frac{\phi}{\psi_\theta} \mathcal{O}_k^* \right\rangle_{\mathcal{B}\sim\mathcal{B}_\psi} \Big/ \left\langle \frac{\phi}{\psi_\theta} \right\rangle_{\mathcal{B}\sim\mathcal{B}_\psi}$;
  foreach k, l **do** evaluate and set $S_{kl} \leftarrow \left\langle \mathcal{O}_k^\dagger \mathcal{O}_l \right\rangle_{\psi_\theta} - \left\langle \mathcal{O}_k^\dagger \right\rangle_{\psi_\theta} \langle \mathcal{O}_l \rangle_{\psi_\theta}$;
  solve linear system $\sum_l S_{kl} \Delta_l = F \times \partial_k \ln F$;
  foreach k **do** $\theta'_k \leftarrow \theta'_k - \eta\, \Delta_k$;
end
return θ'

A. Model compression

III. RELATION TO T-VMC

Time-dependent Variational Monte Carlo (t-VMC) [2] is a numerical technique used in many-body quantum physics to approximately capture time evolution of an arbitrary state often captured by the unitary operator e^{-iHt} associated with the system hamiltonian H. The starting point of such calculations is often almost identical to Eq. 19:

$$\left|\psi_{\theta(t+\Delta t)}\right\rangle = C\, e^{-iH\Delta t} \left|\psi_{\theta(t)}\right\rangle \ , \tag{15}$$

after defining an appropriate variational ansatz ψ_θ, usually by defining log-derivative operators $\mathcal{O}_k$, like in the main text of this work. By repeating the calculation outlined in the previous subsection (which in this case is equivalent to plugging into the time-dependent Schrödinger equation), one obtains the so-called optimal equations of motion:

$$\sum_l \langle \mathcal{O}_k^* \mathcal{O}_l \rangle_t^c \, \dot{\theta}_k(t) = -i \langle \mathcal{O}_k^* H \rangle_t^c \ , \tag{16}$$

where $\langle AB \rangle_t^c \equiv \langle AB \rangle_t - \langle A \rangle_t \langle B \rangle_t$ and $\langle \cdots \rangle_t \equiv \langle \psi_\theta(t) | \cdots | \psi_\theta(t) \rangle \,/\, \langle \psi_\theta(t) | \psi_\theta(t) \rangle$. At this point, any standard ODE solver can be employed to propagate parameters θ_k away from the initial condition after using MCMC to stochastically estimate averages $\langle \cdots \rangle_t^c$ at each step. We note that t-VMC requires inverting the matrix $\langle \mathcal{O}_k^* \mathcal{O}_l \rangle_t^c$ which is analogous to the S matrix given in Eq. 22 in the main text.

In the case of QAOA, one might wish to approximate the action of $U_B(\beta) = \exp\left(-i\beta \sum_j X_j\right)$ by employing t-VMC to "propagate" the relevant parameters in β instead of physical time. That approach would have the benefit of being able to apply the entire U_B gate in small $\Delta\beta$ increments instead of doing it qubit-by-qubit. Indeed, if we take Eq. 23 from the main text and set $|\phi\rangle = e^{-i\Delta\beta X_j} |\psi_\theta\rangle$, we obtain:

$$\frac{\partial \mathcal{D}}{\partial \theta_k^*} = \left(\frac{i}{2} \sin(2\Delta\beta) - \sin^2(\Delta\beta) \left\langle \frac{\psi_\theta^{X_j}}{\psi_\theta} \right\rangle_{\psi_\theta}^* \right) \left[\left\langle \mathcal{O}_k^* \frac{\psi_\theta^{X_j}}{\psi_\theta} \right\rangle_{\psi_\theta} - \langle \mathcal{O}_k^* \rangle_{\psi_\theta} \left\langle \frac{\psi_\theta^{X_j}}{\psi_\theta} \right\rangle_{\psi_\theta} \right] \ , \tag{17}$$

where $\psi_\theta^{X_j}(\mathcal{B}) \equiv \langle \mathcal{B} | X_j | \psi_\theta \rangle$. If one expands the first factor to first order in $\Delta\beta$, what is left is exactly the t-VMC parameter update one would obtain by following the t-VMC derivation from the beginning. Therefore, our method is more general than t-VMC for any individual qubit. In addition to generality, it has two other key features:

1. Fidelity $F(\phi, \psi)$ is directly used as a cost function and it is recorded at each optimization step. This provides for a more controlled environment where fidelity is explicitly optimized over rather than implicitly. We can apply as many gradient updates as needed to reach the target fidelity.

2. The computational cost of complete (approximate) application of U_B is similar between the two methods. We report that the explicit fidelity method we used in this work required approximately 30 updates per qubit while t-VMC needed 1000-2000 intermediate β points to reach similar final fidelities on 20-qubit test systems.

---

[1] Wang, Z., Hadfield, S., Jiang, Z. & Rieffel, E. G. Quantum approximate optimization algorithm for MaxCut: A fermionic view. *Phys. Rev. A* **97**, 022304 (2018).

[2] Carleo, G., Becca, F., Schiró, M. & Fabrizio, M. Localization and glassy dynamics of many-body quantum systems. *Sci. Rep.* **2**, 243 (2012). 1109.2516.