Solving the Four Corners Map Coloring Problem with a Variational Quantum Algorithm

SeokHyeon (Lucas) Kong [1] Chanhee Park [1] Jang-il Moon [1] Denny Dahl (Advisor) [2]

[1]Sungkyunkwan University (SKKU) [2]IonQ

Problem Definition

Figure 1. Problem definition for the Four Corners map coloring

The four corners map coloring problem is to assign one of two colors to each of the four states (UT, CO, AZ, and NM) in the Four Corners region, such that neighboring states — those that share a border — have different colors. In this work, we compare two encoding schemes, unary and binary, for mapping into quantum systems. Each encoding is expressed as a QUBO (Quadratic Unconstrained Binary Optimization) formulation representing the cost function to be minimized.

Unary Encoding

In unary (one-hot) encoding, each state ($i \in \{U, C, A, N\}$) has two variables ($x_{i,\alpha}$) representing the two possible color choices ($\alpha \in \{r, b\}$) for the region. The variable $x_{i,\alpha}$ indicates whether region i is assigned color α (1 if assigned, 0 otherwise). With that, we can derive two constraints:

- Constraint 1: Ensures that each state is assigned exactly one color.

$$C_{oh} = \sum_{i\in\{U,C,N,A\}} \left(x_{i,r} + x_{i,b} - 1\right)^2$$

- Constraint 2: Ensures that neighboring states are assigned different colors.

$$C_{nn} = \sum_{\langle i,j\rangle} \left(x_{i,r}x_{j,r} + x_{i,b}x_{j,b}\right)$$

We can set the QUBO as

$$QUBO = C_{oh} + C_{nn}$$

However, to reduce the size of the QUBO (and consequently the Hamiltonian), we can satisfy the first constraint C_{oh} by preparing the state with Bell states. Specifically, each region can be initialized in

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right)$$

With this construction, the QUBO reduces to the remaining constraint C_{nn}.

Binary Encoding

In binary encoding, each state ($i \in \{U, C, A, N\}$) is represented by a single variable x_i (1 if red, 0 if blue), since each state can only take one of the two colors. Accordingly, the first constraint C_{oh} (introduced above) is automatically satisfied. The only constraint C_{nn} to be satisfied is that neighboring states must be assigned different colors.

$$QUBO = C_{nn} = \sum_{\langle i,j\rangle} \left(x_i x_j + (1 - x_i)(1 - x_j)\right)$$

Since there is no 'no assignment' case, binary encoding requires only half the number of qubits compared to unary encoding. However, this requires the number of colors to satisfy $C = 2^r$; otherwise, additional constraint terms must be introduced.

From QUBO to Hamiltonian

Different from the variable $x_i \in \{0, 1\}$ used in the QUBO formulation, quantum Hamiltonians are usually expressed in terms of spin variables $z_i \in \{-1, 1\}$, which correspond to the eigenvalues of the Pauli-Z operator. We map

$$x_i = \frac{1 - z_i}{2}$$

and substitute this into the QUBO expression to rewrite the cost function in terms of z_i as an Ising model. The corresponding Hamiltonians in the Ising form for both encodings, where we promote $z_i \mapsto \sigma_z^{(i)}$, are as follows:

Unary encoding

$$\hat{H}_{\text{unary}} = -\tfrac{1}{2} \sum_{i\in\{U,C,A,N\}} \left(\sigma_z^{(i,r)} + \sigma_z^{(i,b)}\right) + \tfrac{1}{4} \sum_{\langle i,j\rangle} \left(\sigma_z^{(i,r)}\sigma_z^{(j,r)} + \sigma_z^{(i,b)}\sigma_z^{(j,b)}\right) + \text{const}$$

However, the term

$$-\tfrac{1}{2} \sum_{i\in\{U,C,A,N\}} \left(\sigma_z^{(i,r)} + \sigma_z^{(i,b)}\right)$$

can be eliminated by the prepared state $|\psi_0\rangle$, which allows us to simplify $\hat{H}_{\text{unary}}$ as follows:

$$\hat{H}_{\text{unary}} = \tfrac{1}{4} \sum_{\langle i,j\rangle} \left(\sigma_z^{(i,r)}\sigma_z^{(j,r)} + \sigma_z^{(i,b)}\sigma_z^{(j,b)}\right) + \text{const}$$

Binary encoding

$$\hat{H}_{\text{binary}} = \tfrac{1}{2}\left(\sigma_z^{(U)}\sigma_z^{(C)} + \sigma_z^{(A)}\sigma_z^{(N)} + \sigma_z^{(U)}\sigma_z^{(A)} + \sigma_z^{(C)}\sigma_z^{(N)}\right) + \text{const}$$

XY Mixers in the Unary Encoding Scheme

$$U_{XY}(\beta) = e^{-i\beta\frac{(X\otimes X + Y\otimes Y)}{2}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\beta) & -i\sin(\beta) & 0 \\ 0 & -i\sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As we prepared $|\psi_0\rangle$ as the initial state, we implement a special mixer that preserves the Bell subspace $\{|01\rangle, |10\rangle\}$. In this way, the constraint is preserved: $|01\rangle$ and $|10\rangle$ rotate (mix) by β through the mixer. Note that each mixer acts on disjoint pairs of qubits and therefore commutes.

Initial Parameter Settings

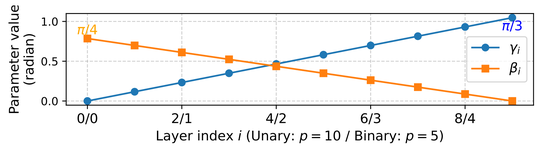


Figure 2. Linear Initialization of QAOA Parameters ($\gamma_0 \sim \gamma_p$, $\beta_0 \sim \beta_p$), where p is the number of layers.

The quantum circuit is parameterized by γ_i and β_i at each layer i. This procedure can be interpreted as a discretized version of adiabatic evolution, where the system is evolved from a simple Hamiltonian to the problem Hamiltonian with the parameters varying linearly and slowly.

QAOA Ansatz

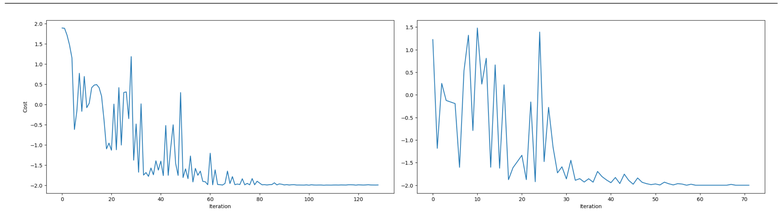
Unary encoding As mentioned previously, the quantum circuit with unary encoding starts with the prepared state $|\psi_0\rangle$, defined as the tensor product of four superpositions of Bell states $\{|01\rangle, |10\rangle\}$ between eight qubits, in order to ensure C_{oh} from the beginning. This initialization reduces the complexity of the Hamiltonian. The quantum circuit is then parameterized with symbolic parameters $\{\gamma_i, \beta_i\}$ for each layer of the cost function and XY-mixer, resulting in a total of 10 layers.

Binary encoding Different from unary encoding, the quantum circuit with binary encoding starts with four superpositions of $|+\rangle$ states to ensure full superposition across four qubits. Since binary encoding requires only half as many qubits as unary encoding, the circuit uses a total of 5 layers, each consisting of a cost-function and a mixer. Here, we employ the standard QAOA X-mixer.

Estimator and Optimizer

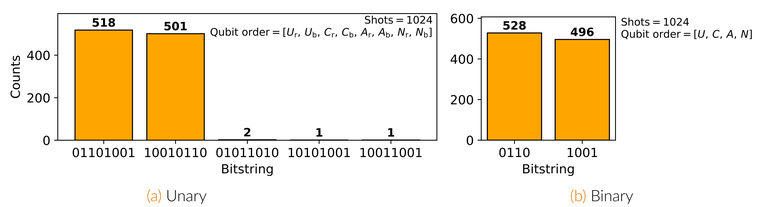
QAOA is a hybrid quantum–classical algorithm. The quantum part—the estimator (here, we used `BackendEstimatorV2` with the IonQ backend)—evaluates the expectation value $\langle H_C \rangle$ of the cost Hamiltonian for a given set of parameters (γ_i, β_i). The classical part—the optimizer (here, `scipy.optimize.minimize` with the COBYLA method)—updates the parameters by searching the parameter space to minimize this cost function.

Results



(a) Unary (b) Binary

Figure 3. Cost per iteration for (a) unary and (b) binary encodings.



(a) Unary (b) Binary

Figure 4. Final bitstring distribution for (a) unary and (b) binary encodings.

As shown in Fig. 3, we observed that the cost decreases as the iterations progress, and we obtained the optimized circuit by assigning the result of the optimizer to the circuit parameters. Consequently, Fig. 4 presents the sampling results with 1024 shots, and the quantum circuits in both encoding schemes perform well, providing the correct solution to the problem: Utah and New Mexico are colored the same, and Colorado and Arizona are colored the same.