Code Documentation for HiQ Competition

Bin Cheng, Ximing Wang and Yuning Zhang

August 1, 2019

1 Problem 1

The Hamiltonian is given by,

$$H(s) = (1-s)H_0 + sH_I \,, \tag{1.1}$$

where

$$H_0 = -B\sum_{k=1}^{N} X_k \tag{1.2}$$

$$H_I = -\alpha\sum_{k=1}^{N} X_k - \beta\sum_{k=1}^{N} Z_k - J\sum_{k=1}^{N-1} X_k X_{k+1} - J\sum_{k=1}^{N-1} Z_k Z_{k+1} \,, \tag{1.3}$$

and X_k and Z_k are Pauli-X and Pauli-Z acting on the k-th qubit. Here, $N = 13$, $J = 2$, $\alpha = 0.5$, $\beta = 1.0$, $B = 0.5$ and $T = 3$.

Let $s := t/T$ and $H(t) := (1 - t/T)H_0 + (t/T)H_1$. Then the evolution that we want to achieve is,

$$\exp\left(-i\int_0^T H(t)\,\mathrm{d}t\right)|+\rangle^N \,. \tag{1.4}$$

Such an evolution can be accomplished by Trotterization. First,

$$\int_0^T H(t)\,\mathrm{d}t \approx \sum_{j=0}^{M-2} \frac{H(j\Delta t) + H((j+1)\Delta t)}{2}\Delta t = \frac{H(0)}{2}\Delta t + \sum_{j=1}^{M-2} H(j\Delta t)\Delta t + \frac{H(T)}{2}\Delta t \,, \tag{1.5}$$

where $M = 21$ and $(M-1)\Delta t = T$, which implies $\Delta t = 3/20$. So we can implement the evolution operator by,

$$\exp\left(-i\int_0^T H(t)\,\mathrm{d}t\right) \approx e^{-iH(0)\Delta t/2}\prod_{j=1}^{M-2} e^{-iH(j\Delta t)\Delta t} e^{-iH(T)\Delta t/2} \tag{1.6}$$

$$\approx e^{-iH_0\Delta t/2}\prod_{j=1}^{M-2} e^{-i(j\Delta t^2/T)H_I} e^{-i\Delta t(1-j\Delta t/T)H_0} e^{-iH_I\Delta t/2} \tag{1.7}$$

These operators can be decomposed into single- or two-qubit gates.

- $j = 0$:

$$e^{-iH_0\Delta t/2} = \prod_{k=1}^{N} e^{iBX_k\Delta t/2} \tag{1.8}$$

- $1 \leq j \leq M-2$: For the H_0 part, we have,

$$e^{-i\Delta t(1-j\Delta t/T)H_0} = \prod_{k=1}^{N} e^{iB\Delta t(1-j\Delta t/T)X_k} \,. \tag{1.9}$$

  For the H_I part, we have,

$$e^{-i(j\Delta t^2/T)H_I} = \prod_{k=1}^{N} e^{i\beta(j\Delta t^2/T)Z_k} \prod_{k=1}^{N-1} e^{iJ(j\Delta t^2/T)Z_kZ_{k+1}} \tag{1.10}$$

$$\prod_{k=1}^{N} e^{i\alpha(j\Delta t^2/T)X_k} \prod_{k=1}^{N-1} e^{iJ(j\Delta t^2/T)X_kX_{k+1}} \,. \tag{1.11}$$

  Combining two parts, the evolution operator for the j-th time step is given by,

$$\prod_{k=1}^{N} e^{i\beta(j\Delta t^2/T)Z_k} \prod_{k=1}^{N-1} e^{iJ(j\Delta t^2/T)Z_kZ_{k+1}} \tag{1.12}$$

$$\prod_{k=1}^{N} e^{i\left(\alpha(j\Delta t^2/T)+B\Delta t(1-j\Delta t/T)\right)X_k} \prod_{k=1}^{N-1} e^{iJ(j\Delta t^2/T)X_kX_{k+1}} \,. \tag{1.13}$$

- $j = M-1$:

$$e^{-i(\Delta t/2)H_I} = \prod_{k=1}^{N} e^{i\beta(\Delta t/2)Z_k} \prod_{k=1}^{N-1} e^{iJ(\Delta t/2)Z_kZ_{k+1}} \tag{1.14}$$

$$\prod_{k=1}^{N} e^{i\alpha(\Delta t/2)X_k} \prod_{k=1}^{N-1} e^{iJ(\Delta t/2)X_kX_{k+1}} \,. \tag{1.15}$$

We apply these gates to the initial state $|+\rangle^N$ sequentially for $j = 0, 1, \cdots, 20$. Let the prepared state be $|\psi\rangle$. Then $E_D = \langle\psi|H_I|\psi\rangle$.

We implemented the circuit with ProjecQ engine and revoke the expectation measurement function in backend to get the corresponding energy after simulated annealing.

The ideal ground state energy can be fairly approximated by setting the number of bins M in Trotterization to be sufficient large (100 as example).

2 Problem 2

The target of this task is to perform state preparation on restricted quantum chips. Noting that the expressive power of a highly entangled parametrized quantum circuit (PQC) is very strong. A sufficient number of ansatzs should estimate any quantum states properly. To build up an highly entangled PQC, we only need strong entangling gates between parameterized ansatzs.

In this case, the ansatz contains only single qubit quantum gates, which doesn't depends on the connectivity of the quantum chips. Therefore, all we need is to construct an entangling gate based on the topology of the chips. For a fully connected quantum chip, a simplest idea is to connect each neighboring qubits with a CNOT gate. But any Multilayer Parameterized Quantum Circuits (MPQC) (see Fig. 1) should work similarly. By the definition of MPQC, we need $n-1$ CNOT gates that connects all qubits.

With restricted connectivity, a good idea is to centralize the CNOT gates on qubits with high connectivity (so the entanglement spread fast). Therefore, we can choose the qubit with highest connectivity as the control bit (root) and performs CNOT with all its neighbors. Then we should choose the child qubit connected to most untouched qubits and perform CNOT onto them.

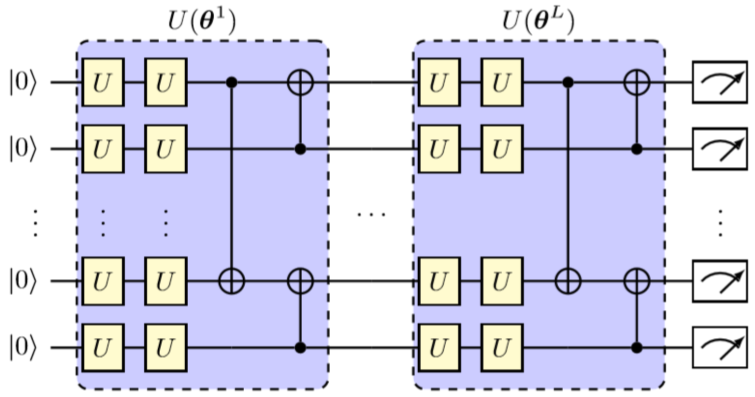

Figure 1: MPQC. arXiv:1810.11922

2.1 Cost function

Suppose $\mathbf{a}$ is the target state vector. We separate its real and imaginary parts $\mathbf{a} = \mathbf{b} + i\mathbf{c}$, where $\mathbf{b}$ and $\mathbf{c}$ are real. Let $\mathbf{a}' = \mathbf{b}' + i\mathbf{c}'$ be our prepared state. Then we need to minimize $|\mathbf{b} - \mathbf{b}'|$ and $|\mathbf{c} - \mathbf{c}'|$. The distance $|\mathbf{b} - \mathbf{b}'|$ is defined as $\sqrt{\sum_i (b_i - b'_i)^2}$.

3 Problem 3

The basic idea of problem 3 is simple–with parameterized circuit, we can set a cost function to be the probability of given target state and tune the parameters to minimize the cost.

However, since the parameterized circuit, with 14 qubits width, contains more that 3000 quantum gates, the seems-easy problem can be actually hard to solve. Traditional optimization techniques based on cost and gradient evaluation require intensive computational resource and what's worse is these algorithms can't resist the attraction of local minimums.

To handle this problem, we use a heuristic technique – the Paricles Swarm Optimization (PSO) algorithm to avoid local minimum. This heuristic comes from an imitation of animal colony searching for food supply or water resources and is very strong against high-dimension complex problem.

In our instance, we initialized a PSO with 40 particles on given circuit to search for optimal parameters that maximize the target probability.

The initial positions of particles are randomly set and generally with in 50 epoches we can achieve the required probability level. Typically this step will consume less than one hour since PSO is slow to converge.