Journal of Physics Communications

OPEN ACCESS

RECEIVED
25 April 2022

REVISED
13 June 2022

ACCEPTED FOR PUBLICATION
29 June 2022

PUBLISHED
14 July 2022

PAPER

Quantum circuit architectures via quantum observable Markov decision process planning

Tomoaki Kimura[1], Kodai Shiba[1,2], Chih-Chieh Chen[2,*], Masaru Sogabe[2], Katsuyoshi Sakamoto[1,3] and Tomah Sogabe[1,2,3,*]

[1] Engineering Department, The University of Electro-Communications, 182-8585 Tokyo, Japan
[2] Grid Inc., 107-0061 Tokyo, Japan
[3] i-PERC, The University of Electro-Communications, 182-8585 Tokyo, Japan
[*] Authors to whom any correspondence should be addressed.

E-mail: chen.chih.chieh@gridsolar.jp and sogabe@uec.ac.jp

Keywords: quantum circuit, partially observable markov decision process, circuit design

Abstract

Algorithms for designing quantum circuit architectures are important steps toward practical quantum computing technology. Applying agent-based artificial intelligence methods for quantum circuit design could improve the efficiency of quantum circuits. We propose a quantum observable Markov decision process planning algorithm for quantum circuit design. Our algorithm does not require state tomography, and hence has low readout sample complexity. Numerical simulations for entangled states preparation and energy minimization are demonstrated. The results show that the proposed method can be used to design quantum circuits to prepare the state and to minimize the energy.

1. Introduction

Quantum computers are attracting attention as computers with computing power that surpasses that of classical computers [P18, AAB19]. In fact, algorithms that efficiently solve specific problems such as Grover's algorithm [G96] and Shor's algorithm [S94] have been proposed. In recent years, variational quantum algorithms [C21] have been actively researched, and quantum technology has been applied to various fields such as chemistry [PMS14, KMT17] and machine learning [DB18, MNK18, SK19, HCT19]. However, the design of a quantum circuit for solving a specific task under hardware constraints requires efforts [SBM06, FM17, LSJ15, SSP14, MFM08, AAH16, HNYN11], sometimes including empirical rules and domain knowledge as well.

Reinforcement learning (RL) [SB18, RN21] has been successful in the areas such as robot control [KCC13] and games [MKS15, SSS17]. Since there is a possibility that RL can solve complicated control problems, research has been conducted to apply RL to the control of quantum systems in recent years [BSK21, SEL21, NBS19, NY17, HWN21]. Most of these studies consider low level control at the hardware (Hamiltonian) level. But it is also important to control at the circuit level [NMM18], which is a higher level of abstraction [AU22], in order to perform concrete quantum computation. For simple circuits, it is demonstrated that the closed-loop control can lead to better control performance for trapped-ion quantum processors [NMM18]. State-of-the-art ion trap qubits have coherence time more than 10 min [WUZ17, WLQ21], which provides enough running time for on-line decision process on a classical computer.

In this paper, we consider applying RL to more general quantum feedback control at the circuit level. The basic RL algorithms solve for Markov Decision Process (MDP), where the current state of the agent can be exactly known from the observation of the environment. But for a quantum system, the Born rule asserts that an observation result is drawn from a probabilistic distribution over the state space. Therefore, it is necessary to formulate the problem as a partially observable problem. Quantum Observable Markov Decision Process (QOMDP) [BBA14, C16, YY18, YFY21] was proposed as a quantum extension of the Partially Observable Markov Decision Process (POMDP) framework for the classical partially observable problems [PT87, RN21], but no specific application of QOMDP was proposed. Our QOMDP planning approach is Bayesian, and does not rely on state tomography [NC11, YC21, KFC21] or expectation evaluation [ZHZY20, PT20, MLWEV21]. Hence it improves the quantum machine sample complexity per time step from $O(\epsilon^{-2}N_{obs})$ (or $O(\epsilon^{-4}(\log N_{obs})^4\ \log(2^n))$ with shadow tomography [A18]) to $O(1)$ for number of observables N_{obs} and accuracy ϵ. However, our approach still requires exponentially expensive classical planning.

In this study, we formulate quantum control at the circuit level as a QOMDP reinforcement learning problem to solve for the quantum circuit design problem [K22]. The exact QOMDP Bellman equation for value iteration is derived. As a concrete algorithm, we propose a QOMDP planning algorithm with reference to planning in POMDP. In the exact POMDP planning for quantum state, there are three computational intractable parts. Firstly, the size of history set grows exponentially in time. Secondly, the Hilbert space is an uncountable set. Thirdly, the Hilbert space dimension grows exponentially with respect to the circuit width. We introduce the point-based value iteration (PBVI) algorithm from classical POMDP to make the approximating planning tractable and resolve the first and second issues. For the quantum Hilbert space, we perform exact filtering and do not make any approximation. Hence the calculations involving the belief state scale exponentially with respect to the number of qubits. We further consider circuit design problem for two types of applications: the problem of state preparation and energy minimization. The proposed algorithm was able to make Bell state and GHZ state [GHZ89] for state preparation. Regarding energy minimization, it was able to discover a low energy state with respect to the H2 and H-He+. The experimental results show the applicability of QOMDP to quantum control at the circuit level. Comparing to variational quantum eigen solver (VQE) [PMS14, MRB16, KMT17, C21] approach where the variational ansatz has to be chosen empirically, the QOMDP approach allows automatic search over a wide range of possible ansatzes.

This paper is organized as follows. Related works are reviewed in section 2. The POMDP planning algorithm is introduced in section 3. Numerical experiments are presented and analyzed in section 4, followed by a concluding section.

2. Related work

Quantum circuit synthesis has been addressed in many works without using RL [SBM06, FM17, LSJ15, SSP14, MFM08, AAH16, HNYN11]. In recent years, RL has been applied to quantum control problem in various settings. The applications to physical design at the Hamiltonian level is studied in various literatures [BSK21, SEL21, B18, MDW21, ZWA19, BAHH21]. RL has also been used for optimization of quantum circuit architecture [YC21, KFC21, ZHZY20, PT20, MLWEV21]. Our work is different from these works regarding the sample complexity (the number of measurement shots) from the real quantum machines. The RL approaches based on state tomography or the expected cost function [YC21, KFC21, ZHZY20, PT20, MLWEV21] requires $O(4^n\epsilon^{-2})$ or $O(\epsilon^{-2}N_{obs})$ shots for reward evaluation, where N_{obs} is the number of observables in the cost function. Better scaling $O(\epsilon^{-4}(\log N_{obs})^4\ \log(2^n))$ could be obtained by using shadow tomography [A18]. Our method takes only $O(1)$ shots for online decision making, at the expense of exponentially expensive classical pre-computing. On the other hand, variational quantum algorithm has been also applied to RL. For example, variational quantum circuit has been applied to value function approximation [C20, MK21, LS20, SJD22, LS21, CHHGK21, KSCS21][SJD22] and policy approximation [J21, K21] in RL. Value iteration algorithm for classical POMDP planning has a long history and many variants [SS73, KLC98, PGT03, TBF05, SV10]. Variational quantum eigen solver (VQE) for molecule energy minimization is studied in [PMS14, MRB16, KMT17, C21].

3. Methods

The overview of our QOMDP-PBVI method is depicted in figure 1. The offline planning is computed with a classical simulator. The output of the offline planning is a matrix set η which approximates the value function. The set η is then stored in a classical agent, and the agent is able to make online decision in a hybrid Quantum–classical computer. The theory and algorithm are explained in the following sections.

3.1. Quantum observable markov decision process

QOMDP[BBA14] is defined by $\mathrm{Q} = \{\mathscr{S},\ \mathcal{O},\ A,\ R,\ \gamma,\ |\ s_0\rangle\}$. $\mathscr{S}$ is the Hilbert space of the system. $\mathcal{O} = \{o^1,\cdots,o^{|\mathcal{O}|}\}$ is the set of observations, where $|X|$ denotes the cardinality of a set X. $A = \{A^{a^1},\cdots,A^{a^{|A|}}\}$ is the set of transition operators, and each operator $A^a = \{A_{o^1}^{a},\cdots,A_{o^{|\mathcal{O}|}}^{a}\}$ has $|\mathcal{O}|$ Kraus matrices. The conditional probability of getting the observation o when executing the action a in the state $|s\rangle$ is

$$\Pr(o||\ s\rangle,\ a) = \langle s|A_o^{a\dagger}A_o^a|s\rangle. \tag{1}$$

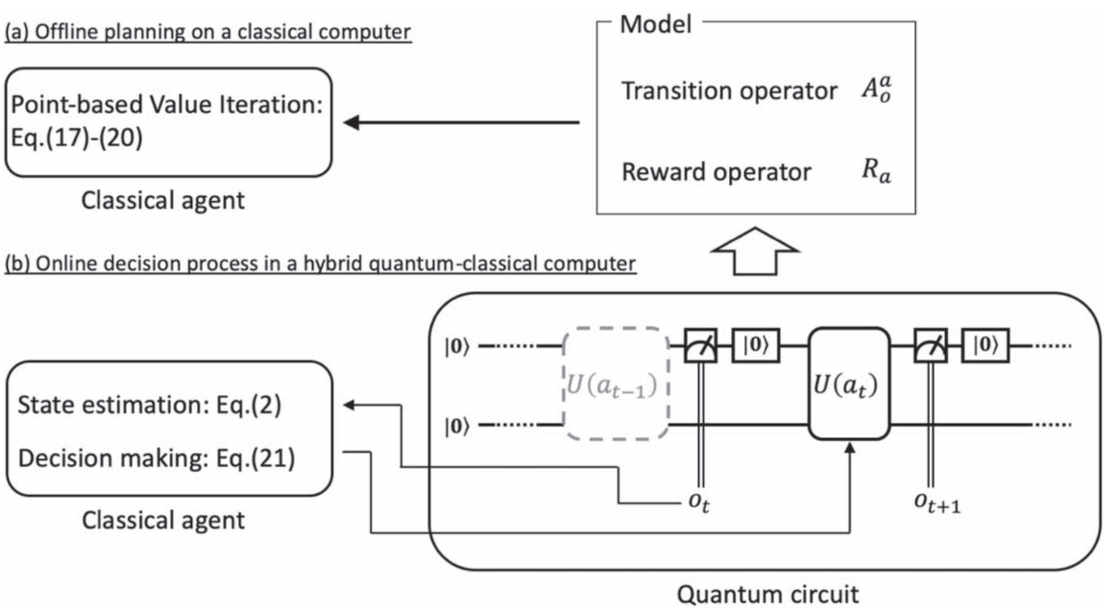


Figure 1. The setting of the point-based value iteration (PBVI) for quantum circuit planning. (a). Offline planning is executed in a classical computer based on classical simulation of the quantum circuit. The output of the PBVI algorithm is a set η which is used for approximating value functions. (b). The classical agent then uses η for efficient online decision making to control a real quantum machine.

The state transition is defined by

$$|s'\rangle(|s\rangle, a, \quad o) \leftarrow \frac{A_o^a \mid s\rangle}{\sqrt{\langle s|A_o^{a\dagger} A_o^a|s\rangle}}. \tag{2}$$

$R = \{R_{a^1},\cdots,R_{a^{|A|}}\}$ is the set of operators for rewards. The reward of executing action a in state $|s\rangle$ is calculated by

$$r(|s\rangle, \quad a) = \langle s|R_a|s\rangle. \tag{3}$$

γ is the discount rate. $|s_0\rangle$ is the initial state. Regarding the interaction between the agent and the environment in QOMDP, the agent selects an action according to the policy and executes the action for the environment. The operation A_o^a corresponding to the action a performed in the environment is executed, and the observation o is fed back to the agent. The agent also receives a reward according to the equation (3). The above action-observation-reward sequence is for a single time step, and this is repeated until the end of the episode. The agent's goal is to maximize expected future rewards. Note the relationship between POMDP and QOMDP. The state in QOMDP corresponds to the belief state in POMDP, and the formula (2) corresponds to the belief state update in POMDP. Therefore, it is natural to think that it is possible to extend the planning method in POMDP and devise a planning method to solve QOMDP. In the next section, we propose a planning algorithm in QOMDP based on this idea.

Firstly, we derive the value function of QOMDP. Let $\pi: A \times \mathscr{S} \rightarrow [0, 1]$ be the policy in a QOMDP described by $\mathrm{Q} = \{\mathscr{S}, \mathscr{O}, A, R, \gamma, |s_0\rangle\}$, and the policy is defined by $\pi(a||s\rangle) = \Pr(a||s\rangle)$. The value function for Q is calculated by

$$V_q^\pi(|s\rangle) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(|s_t\rangle, \quad a_t) \,\middle|\, |s_0\rangle = |s\rangle, \mathrm{Q}(\pi)\right]. \tag{4}$$

The Bellman equation is

$$V_q^\pi(|s\rangle) = \sum_a \pi(a||s\rangle)\left\{r(|s\rangle, a) + \gamma \sum_o \Pr(o||s\rangle, a) V_q^\pi(|s''\rangle)\right\}. \tag{5}$$

Since it is known that the value function can be expressed in a simple form of piece-wise linear and convex function in the classical POMDP, it seems that the value function can be expressed in some simple form in QOMDP as well. In the following paragraph, we derive the expression of value function in QOMDP.

Let $h_t \in \mathscr{H}_t$ be the history up to time step t and the history is expressed by

$$h_t = \{a_0, o_1, \quad a_1, \quad o_2, \ldots, \quad a_{t-1}, o_t\}. \tag{6}$$

Let $S{:}H_t \times \mathscr{S} \to \mathscr{S}$ be the mapping from the initial state $|s\rangle$ and the history h_t to the transitioned state $S(h_t, |s\rangle)$. This function can be calculated by

$$S(h_t, \mid s\rangle) = \begin{cases} \dfrac{\prod_{i=1}^{t} A_{o_{t-i+1}}^{a_{t-i}}}{\sqrt{\prod_{i=1}^{t} Pr(o_{t-i+1} \mid S(h_{t-i}, \mid s\rangle), a_{t-i})}} \mid s\rangle, t \geqslant 1 \\ \mid s\rangle, t = 0 \end{cases}. \tag{7}$$

The probability of obtaining the history h_t given the initial state $|s\rangle$ is calculated by

$$\begin{aligned} \Pr(h_t||s\rangle) &= \Pr(a_{t-1},\ o_t|h_{t-1}, |s)\Pr(h_{t-1}||s\rangle) \\ &= \prod_{i=1}^{t} \Pr(a_{t-i}, \quad o_{t-i+1}|h_{t-i}, \mid s\rangle) \\ &= \prod_{t}^{i=1} \Pr(o_{t-i+1}|h_{t-i}, |s\rangle, a_{t-i})\Pr(a_{t-i}|h_{t-i}, |s\rangle) \\ &= \prod_{t}^{i=1} \Pr(o_{t-i+1}|S(h_{t-i}, |s\rangle), a_{t-i})\pi(a_{t-i}|S(h_{t-i}, |s\rangle)) \end{aligned} \tag{8}$$

Value function is calculated using equations (3), (7), and (8). The detail derivation is presented in appendix. The result is

$$V_q^{\pi}(\mid s\rangle) = \langle s \mid \Upsilon(\pi) \mid s\rangle, \tag{9}$$

where $\Upsilon(\pi) = \sum_t \sum_{a_t} \sum_{h_t} \gamma^t \prod_{i=0}^{t} \pi(a_{t-i}|S(h_{t-i}, \mid s\rangle))(\prod_{i=1}^{t} A_{o_{t-i+1}}^{a_{t-i}})^{\dagger} R_{a_t} \prod_{i=1}^{t} \ A_{o_{t-i+1}}^{a_{t-i}}$. Equation (9) shows that in QOMDP the value function can be expressed in the form of the expectation value of Υ matrix with respect to state. Since the optimal value function $V_q^{*}(\mid s\rangle)$ is the maximum value function,

$$V_q^{*}(\mid s\rangle) = \max_{\pi \in \Pi} V_q^{\pi}(\mid s\rangle) = \max_{\pi \in \Pi} \langle s \mid \Upsilon(\pi) \mid s\rangle, \tag{10}$$

using the set of Υ matrix $\eta_{\text{all}} = \{\Upsilon(\pi)|\pi \in \Pi\}$, equation (10) is represented by

$$V_q^{*}(\mid s\rangle) = \max_{\Upsilon \in \eta_{\text{all}}} \langle s \mid \Upsilon \mid s\rangle. \tag{11}$$

3.2. Point-based value iteration algorithm

Since the policy π has continuous parameters, Υ also has continuous parameters. Therefore η_{all} becomes an uncountably infinite set. Since the equation (11) cannot be calculated for an uncountably infinite set, the optimal value function in equation (11) is approximated as follows using a finite set of Υ matrix $\eta = \{\Upsilon_1, \cdots, \Upsilon_l\}$.

$$V_q(\mid s\rangle) = \max_{\Upsilon \in \eta} \langle s|\Upsilon|s\rangle, \quad \eta = \{\Upsilon_1, \cdots, \Upsilon_l\}. \tag{12}$$

As we have confirmed that the value function can be expressed as equation (12), next we will explain how to update this value function. When we update the value function, we update the Υ matrix set $\eta = \{\Upsilon_1, \cdots, \Upsilon_l\}$. The value function can be calculated from previous value function by Bellman equation as follows.

$$\begin{aligned} V_q(\mid s\rangle) &= \max_a \left\{ r(\mid s\rangle, a) + \gamma \textstyle\sum_o \Pr(o|\mid s\rangle, \quad a) V_q(\mid s'\rangle) \right\} \\ &= \max_a \left\{ r(\mid s\rangle, \quad a) + \gamma \textstyle\sum_o Pr(o|\mid s\rangle, \quad a) V_q\left(\frac{A_o^a \mid s\rangle}{\sqrt{\Pr(o \mid\mid s\rangle, \quad a)}} \right) \right\} = \max_a \left\{ \langle s|R_a|s\rangle + \gamma \textstyle\sum_o \max_{\Upsilon \in \eta} \langle s|A_o^{a\dagger} \Upsilon A_o^a|s\rangle \right\} \\ &= \max_{\Upsilon \in \eta^{|s\rangle}} \langle s|\Upsilon|s\rangle, \end{aligned} \tag{13}$$

where

$$\eta^{|s\rangle} = \{\Upsilon^{a^1, |s\rangle}, \cdots, \Upsilon^{a^{|A|}, |s\rangle}\}, \tag{14}$$

$$\Upsilon^{a,|s} = R_a + \gamma \sum_o A_o^{a\dagger} \left(\operatorname*{argmax}_{\Upsilon \in \eta} s|A_o^{a\dagger} \Upsilon A_o^a|s \right) A_o^a. \tag{15}$$

The Υ matrix set η will be updated by

$$\eta' = \bigcup_{|s \in \mathscr{S}\rangle} \operatorname*{argmax}_{\Upsilon \in \eta^{|s\rangle}} \langle s \mid \Upsilon|s\rangle. \tag{16}$$

However, it should be noted here that equation (16) cannot be calculated because $\mathscr{S}$ is an uncountably infinite space. Therefore, it is necessary to update the Υ matrix sets η without using equation (16). In this research, we propose an algorithm updating Υ matrix sets η based on point-based value iteration [PGT03] classical POMDP planning method for this problem.

In this section, we propose our QOMDP planning algorithm based on the classical POMDP planning algorithm PBVI [PGT03]. In the point-based method, the problem that the union in equation (16) cannot be calculated is dealt with by approximating the uncountable state space. Since the state space $\mathscr{S}$ is a Hilbert space and the number of elements is infinite, we consider approximating this with a set of a finite number of state $\mathscr{S}^{\sim} = \{| s_0\rangle, | s_1\rangle, \cdots, | s_v\rangle\}$. As a result, the calculation of equation (16) can be performed and the Υ matrix set η can be updated as follows.

$$\eta' = \bigcup_{|s\rangle \in \mathscr{S}^{\sim}} \mathrm{backup}(|s\rangle) \tag{17}$$

$$backup(|s\rangle) = \underset{\Upsilon \in \eta^{|s} = \{\Upsilon^{a,|s}\}_{a \in \mathscr{A}}}{\mathrm{argmax}} \langle s| \Upsilon |s\rangle \tag{18}$$

$$\Upsilon^{a,|s\rangle} = \boldsymbol{R_a} + \gamma \sum_{o} \underset{\Upsilon \in \eta^{a,o}}{\mathrm{argmax}} \langle \boldsymbol{s}| \Upsilon |\boldsymbol{s}\rangle | \tag{19}$$

$$\eta^{a,o} = \{A_o^{a\dagger} \Upsilon A_o^a \colon \Upsilon \in \eta\} \tag{20}$$

The approximation state set $\mathscr{S}^{\sim}$ is expanded alternately with the update of the value function in each iteration. Let $\mathscr{S}^{\sim} = \{|s_0\rangle, |s_1\rangle, \cdots\}$ be the state set before the expansion. Each action is executed for each state $|s \in \mathscr{S}^{\sim}$. One observation is sampled, and a new state set $\{| s'_{a^1}\rangle, | s'_{a^2}\rangle, \}$ is generated. The new state $| s'_a\rangle$ is discarded if $| s'_a\rangle \in \mathscr{S}^{\sim}$. For each generated state $| s'_{a^i}\rangle$, get the shortest distance to the state belonging to the state set before expansion $\mathscr{S}^{\sim} = \{| s_0\rangle, | s_1\rangle, \cdots\}$. For the distance, the L2 norm $d_a = | \langle s_j | s'_{a^i}\rangle |^2$ is used in this work. The state with the largest distance among the obtained shortest distances is added to $\mathscr{S}^{\sim}$ as a new state. Since the state set is expanded by executing the above process for all the states belonging to the state set before the expansion, the size of the set become doubled at most. The initial condition for the matrix set is $\eta = \{0_{2^n \times 2^n}\}$ where $0_{2^n \times 2^n}$ is the zero matrix with dimension $2^n \times 2^n$. The initial condition of the point set is $\mathscr{S}^{\sim} = \{| s_0\rangle\}$. The value function is updated as many times as the number of horizons, then the state set is expanded. These value function update and state set expansion are executed alternatively. The pseudocode is shown in figure 2 Algorithm 1 for state set expansion and Algorithm 2 for value function update.

3.3. Policy for decision making

In this section, we explain the policy of how to decide an action based on the updated value function. Let η be the Υ matrix set after executing point-based value iteration algorithm. The value function is represented by

$$V_q(|s\rangle) = \max_{\Upsilon \in \eta} \langle s | \Upsilon | s \rangle.$$

In equations (17)–(19), there is an action corresponding to each Υ matrix. The optimal action a^* is decided as the action corresponding to the highest valued Υ matrix:

$$\Upsilon_{max}^{a^*, |s^*\rangle} = \underset{\Upsilon^{a^*, |s^*\rangle} \in \eta}{\mathrm{argmax}}\, V_q(|s\rangle) = \underset{\Upsilon^{a^*, |s^*\rangle} \in \eta}{\mathrm{argmax}} \langle s| \Upsilon |s\rangle, \tag{21}$$

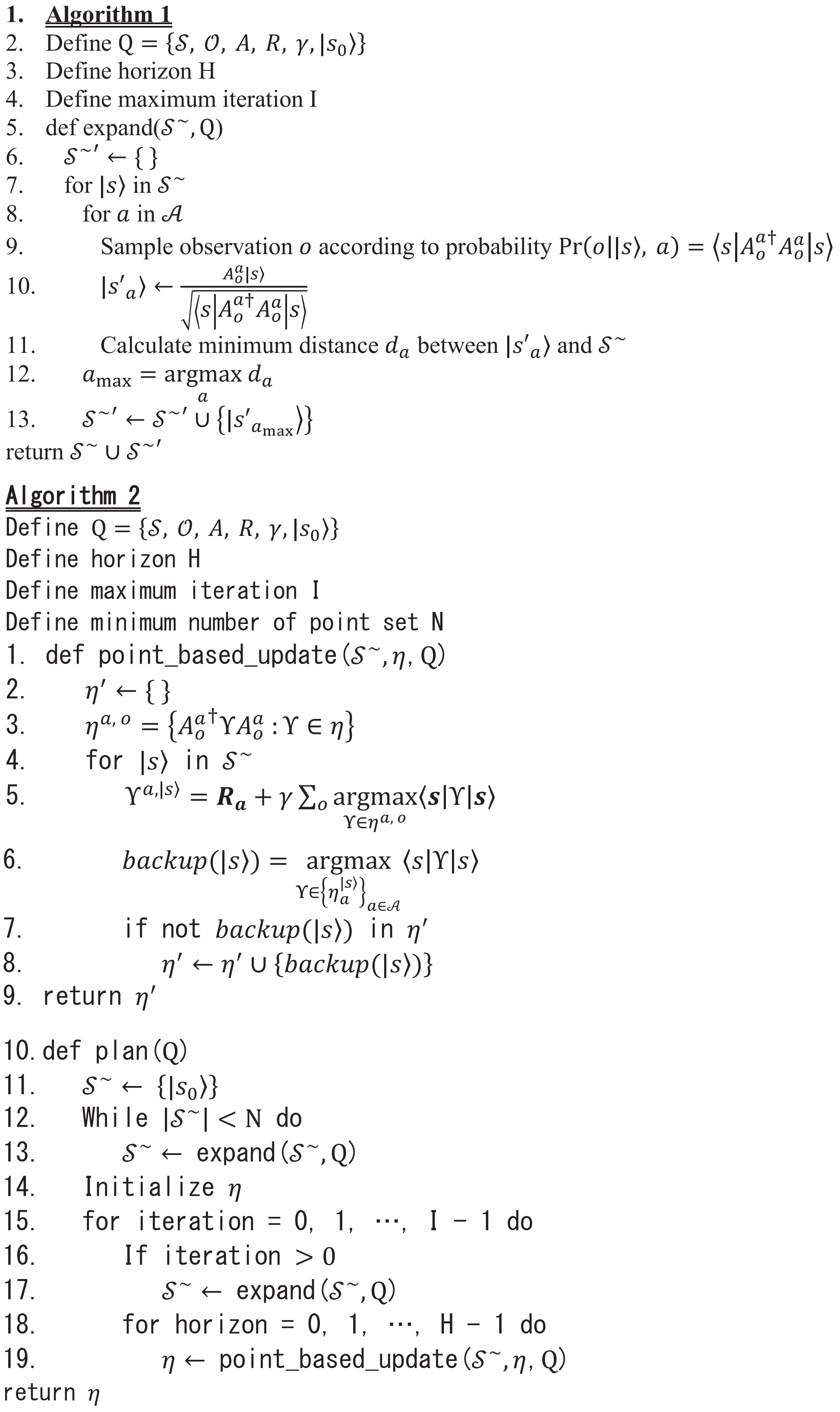
where $\eta = \{\Upsilon^{a', |s'\rangle}\}_{|s'\rangle \in \mathscr{S}^{\sim}}$. Notice that the elements are only indexed by $|s'\rangle$, so the size of the set is $|\eta| = |\mathscr{S}^{\sim}|$.

For a real quantum device, the agent updates value function by point-based value iteration using only classical computer, and then executes the action decided by the policy in the real device. The agent executes an action calculated by the policy and gets an observation from real device, updates belief state by equation (2) using the action and the observation, and calculates a next action based on the updated belief state.

3.4. Complexity analysis

In this section, we explain the computational complexity of point-based method. We first notice the sample complexity advantage of our method over traditional state tomography-based methods [KFC21]. The state tomography of n qubit system $\rho = \frac{1}{2^n} \sum_{3}^{i_1, .., i_n = 0} Tr(\rho \sigma_{i_1} \otimes \ldots \otimes \sigma_{i_n}) \sigma_{i_1} \otimes \ldots \otimes \sigma_{i_n}$ costs $O(4^n)$ Pauli measurements [NC11], and each measurement costs $O\left(\left(\frac{1}{\epsilon}\right)^2\right)$ shots. However, in our QOMDP method it only costs one single shot for each time step.

For the analysis of computational complexity, we assume that all the matrix operations are naïve matrix operations, so the matrix multiplication for $m \times n$ matrix and $n \times r$ matrix has time complexity $O(mnr)$. We first notice the intractability of the exact planning algorithm: (1) The size of history set is $|h_t| = |A|^t |\mathscr{O}|^t$, which grows exponentially in time. (2) The Hilbert space $\mathscr{S}$ is infinitely uncountable. (3) The Hilbert space dimension is $|\mathscr{S}| = 2^n$ for n qubits. We use finite set approximation to tackle the first two intractability. We employ the notation that $|A|$ is the number of actions. $|\mathscr{O}|$ is the number of observations. $|\eta|$ is the number of Υ matrices in the previous update step. $|\mathscr{S}^{\sim}|$ is the number of states in the state set. Equation (20) creates $|A||\mathscr{O}||\eta|$ items in time $O(|\mathscr{S}|^3 |A||\mathscr{O}||\eta|)$. Equation (19) calculates $|\{\Upsilon^{a,|s}\}| = |A||\mathscr{S}^{\sim}|$ items in time $O(|\mathscr{S}|^2 |A||\mathscr{O}||\eta||\mathscr{S}^{\sim}|)$.



```
Algorithm 1
Define Q = {$\mathcal{S}$, $\mathcal{O}$, $A$, $R$, $\gamma$, $|s_0\rangle$}
Define horizon H
Define maximum iteration I
def expand($\mathcal{S}^{\sim}$, Q)
  $\mathcal{S}^{\sim\prime} \leftarrow \{\}$
  for $|s\rangle$ in $\mathcal{S}^{\sim}$
    for $a$ in $\mathcal{A}$
      Sample observation $o$ according to probability $\Pr(o||s\rangle, a) = \langle s|A_o^{a\dagger}A_o^a|s\rangle$
      $|s'_a\rangle \leftarrow \frac{A_o^a|s\rangle}{\sqrt{\langle s|A_o^{a\dagger}A_o^a|s\rangle}}$
      Calculate minimum distance $d_a$ between $|s'_a\rangle$ and $\mathcal{S}^{\sim}$
    $a_{\max} = \underset{a}{\operatorname{argmax}}\, d_a$
    $\mathcal{S}^{\sim\prime} \leftarrow \mathcal{S}^{\sim\prime} \cup \{|s'_{a_{\max}}\rangle\}$
return $\mathcal{S}^{\sim} \cup \mathcal{S}^{\sim\prime}$

Algorithm 2
Define Q = {$\mathcal{S}$, $\mathcal{O}$, $A$, $R$, $\gamma$, $|s_0\rangle$}
Define horizon H
Define maximum iteration I
Define minimum number of point set N
def point_based_update($\mathcal{S}^{\sim}$, $\eta$, Q)
    $\eta' \leftarrow \{\}$
    $\eta^{a,o} = \{A_o^{a\dagger}\Upsilon A_o^a : \Upsilon \in \eta\}$
    for $|s\rangle$ in $\mathcal{S}^{\sim}$
        $\Upsilon^{a,|s\rangle} = \boldsymbol{R_a} + \gamma \sum_o \underset{\Upsilon\in\eta^{a,o}}{\operatorname{argmax}} \langle \boldsymbol{s}|\Upsilon|\boldsymbol{s}\rangle$
        $backup(|s\rangle) = \underset{\Upsilon\in\{\eta_a^{|s\rangle}\}_{a\in\mathcal{A}}}{\operatorname{argmax}} \langle s|\Upsilon|s\rangle$
        if not $backup(|s\rangle)$ in $\eta'$
            $\eta' \leftarrow \eta' \cup \{backup(|s\rangle)\}$
return $\eta'$

def plan(Q)
    $\mathcal{S}^{\sim} \leftarrow \{|s_0\rangle\}$
    While $|\mathcal{S}^{\sim}| < $ N do
        $\mathcal{S}^{\sim} \leftarrow$ expand($\mathcal{S}^{\sim}$, Q)
    Initialize $\eta$
    for iteration = 0, 1, ⋯, I − 1 do
        If iteration $> 0$
            $\mathcal{S}^{\sim} \leftarrow$ expand($\mathcal{S}^{\sim}$, Q)
        for horizon = 0, 1, ⋯, H − 1 do
            $\eta \leftarrow$ point_based_update($\mathcal{S}^{\sim}$, $\eta$, Q)
return $\eta$
```

Figure 2. Pseudocode for our QOMDP-PBVI planning algorithm.

Equations (17) and (18) gets $|\eta'| = |\mathscr{S}^{\sim}|$ items in time $O(|\mathscr{S}|^2|A||\mathscr{S}^{\sim}|)$. We notice that the overall time complexity of the algorithm is polynomial in ($|\mathscr{S}|$, $|A|$, $|\mathscr{O}|$, $|\eta|$, $|\mathscr{S}^{\sim}|$), and the worst part of the planning algorithm has time complexity $O(|\mathscr{S}|^2|A||\mathscr{O}||\eta||\mathscr{S}^{\sim}|)$. Also notice that the size of the point set growth exponentially with respect to the planning horizon $|\mathscr{S}_{\tilde{T}}| = O(2^T|\mathscr{S}_{\tilde{0}}|)$. However, this exponential growth could

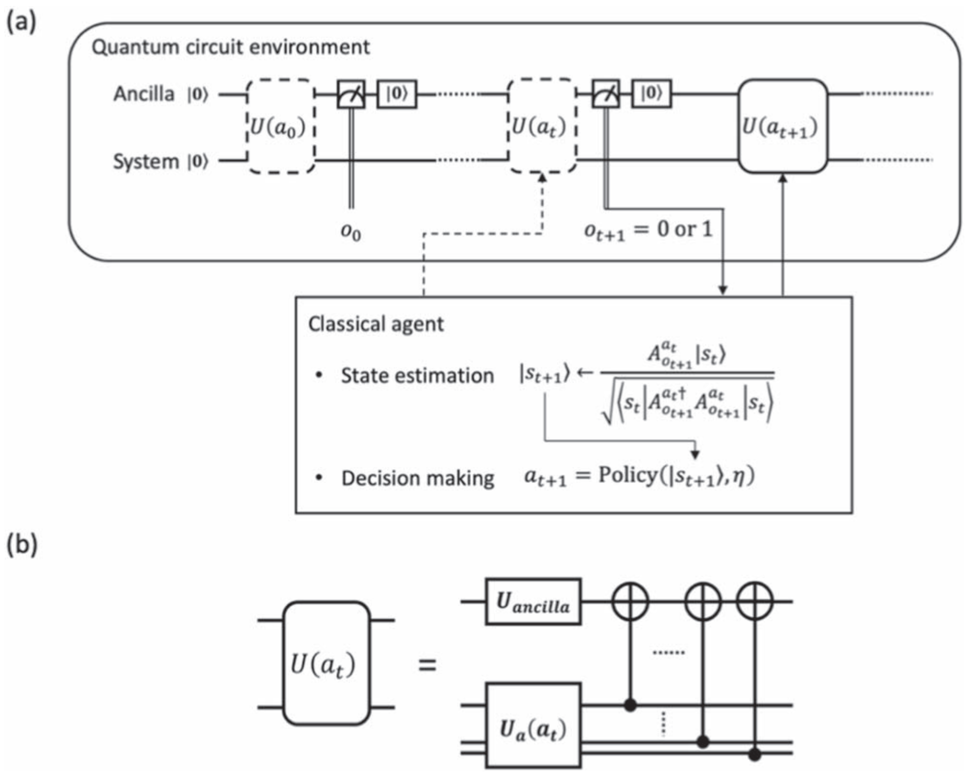

(a)
Quantum circuit environment
Ancilla |0⟩
System |0⟩
$U(a_0)$
|0⟩
$U(a_t)$
|0⟩
$U(a_{t+1})$
o_0
$o_{t+1} = 0$ or 1
Classical agent
• State estimation $|s_{t+1}\rangle \leftarrow \frac{A^{a_t}_{o_{t+1}}|s_t\rangle}{\sqrt{\langle s_t|A^{a_t\dagger}_{o_{t+1}}A^{a_t}_{o_{t+1}}|s_t\rangle}}$
• Decision making $a_{t+1} = \mathrm{Policy}(|s_{t+1}\rangle, \eta)$
(b)
$U(a_t)$
=
$U_{ancilla}$
$U_a(a_t)$
......

Figure 3. Quantum circuit design. (a) The overall circuit structure. Each decision-making step for the classical agent consists of some unitary $U\ (a)$ and a measurement on the Ancilla qubit. (b) The unitary $U\ (a)$ can be decomposed into $U_{ancilla}$, $U_a(a_t)$, and some ancilla-system entanglers.

be easily fixed by imposing a truncation threshold in the expansion subroutine Algorithm 1. For the on-line decision making, the time complexity is $O(|\mathscr{S}|^2|\mathscr{S}^{\sim}|)$ for equation (21) and $O(|\mathscr{S}|^2)$ for equation (2).

3.5. Applications: quantum circuit design

We define the quantum circuit design as a RL problem using QOMDP. Quantum circuit design is a task to arrange the gates and bits in a circuit in order to solve a specific problem in a quantum computer. When trying to formulate this quantum circuit design with the framework of RL, it is necessary to pay attention to the partially observability in the quantum system. Therefore, it is appropriate to use the QOMDP, which can handle partially observability.

We will implement the quantum circuit design using QOMDP as follows. We prepare two subsystems, 'System' and 'Ancilla', and apply a unitary to the whole system as an action. An observation is a measurement outcome of 'Ancilla'. The circuit is designed with an action sequence that maximizes the sum of rewards defined based on the state of 'System'. 'System' is a main system and 'Ancilla' is an auxiliary system for indirectly acquiring 'System' information. Since the quantum state would collapse if 'System' is measured, it is a partially observable problem in which the state of 'System' is inferred by measuring 'Ancilla' without measuring 'System'.

The specific flow is as shown in figure 3(a). The agent is classically implemented, and the environment contains the quantum circuit. At each step, the agent selects a quantum gate to be executed in the circuit as an action from the action set, and then executes the action in the environment. The operations performed for a given action include the selected quantum gate in the circuit, measuring 'Ancilla' and obtaining the measurement result. The reward is calculated from the state of 'System', and the measurement result and the reward are fed back to the agent. The agent makes a classical update based on the obtained measurement result and performs the next action based on it. The above flow is repeated until the evaluation result of the task reaches the threshold value or the number of steps reaches the maximum number of steps.

In this quantum circuit design, each item of QOMDP is as follows. $\mathscr{S}$ is the Hilbert space of 'System + Ancilla.' $\mathscr{O} = \{0,\ 1\}$ is the set of measurement outcome of 'Ancilla'. The unitary applied for an action is shown in figure 3(b). Each action unitary $U\ (a)$ consists of three parts: (1) A unitary $U_a(a_t)$ acting on the 'System' part. (2) Another unitary $U_{ancilla}$ acting on the 'Ancilla' part. (3) Some entanglers between 'System' and 'Ancilla.' While $U_a(a_t)$ is action-dependent, $U_{ancilla}$ and the system-ancilla entanglers are treated as tunable hyperparameters. An action a_t determines the unitary $U_a(a_t)$ applied for 'System' from the action set $\mathscr{A}$. The action set $\mathscr{A}$ is defined by

$$A = \bigcup_{i=0}^{n-1}\left\{\left\{\mathrm{Rx}_i\left(-\frac{\pi}{9}\right),\ \mathrm{Rx}_i\left(\frac{\pi}{9}\right),\ \mathrm{Ry}_i\left(-\frac{\pi}{9}\right),\mathrm{Ry}_i\left(\frac{\pi}{9}\right),\ \mathrm{Rz}_i\left(-\frac{\pi}{9}\right),\ \mathrm{Rz}_i\left(\frac{\pi}{9}\right),\ H_i\right\} \times \cup \bigcup_{j=0}^{n-1} \mathrm{CX}_{i,\, j\neq i}\right\} \tag{22}$$

where $\mathrm{Rx}_i(\theta)$, $\mathrm{Ry}_i(\theta)$, and $\mathrm{Rz}_i(\theta)$ are Rx, Ry, and Rz gates of the rotation angle θ applied to the i-th qubit, and H_i is the Hadamard gate applied to the Ã°ÂÂ–-th qubit. $\mathrm{CX}_{i,\,j}$ is the controlled NOT gate whose control bit is the i -th qubit and target bit is the j -th qubit.

At the end of each action, an measurement is performed on the ancilla and hence the corresponding Kraus operator A_o^a is defined by

$$A_o^a = (I_{system} \otimes \langle e_o|)U(a)(I_{system} \otimes |0\rangle), \tag{23}$$

where I_{system} is identity gate of 'System' and $|e_o\rangle$ is an orthonormal basis vector of state space of 'Ancilla'. R is defined for each task so that the reward can be calculated. γ is the discount rate. $|s_0\rangle$ is $|0\rangle$. We note that this approach requires fast resetting of qubits, which could be done for superconducting qubits [YT21]. In this paper, we demonstrate two quantum circuit design examples: state preparation and energy minimization. Each task is explained as follows.

3.6. Task 1: state preparation

State preparation is a task to design a quantum circuit to generate a target state $|s_{\mathrm{Target}}\rangle$. This task is implemented by using fidelity as the reward in above quantum circuit design. When fidelity is used for reward, the reward is calculated by

$$\begin{aligned} r(|\boldsymbol{s},\ a\rangle) = \mathbb{E}_o[\text{fidelity}] = \sum_o Pr(o||s,\ a\rangle)\ \ |\langle s'||\boldsymbol{s}_{\mathbf{Target}}\rangle|^2 = \sum_o\ |\langle \boldsymbol{s}|\ A_o^{a\,\dagger}|\boldsymbol{s}_{\mathbf{Target}}\rangle|^2 \\ = \sum_o \langle \boldsymbol{s}|\ A_o^{a\,\dagger}|\boldsymbol{s}_{\mathbf{Target}}\rangle\langle \boldsymbol{s}_{\mathbf{Target}}|\ A_o^a|\boldsymbol{s}\rangle = \langle \boldsymbol{s}|\sum_o A_o^{a\,\dagger}|\boldsymbol{s}_{\mathbf{Target}}\rangle\langle \boldsymbol{s}_{\mathbf{Target}}|\ A_o^a|\boldsymbol{s}\rangle. \end{aligned} \tag{24}$$

In QOMDP, the reward is calculated by equation (3). Equation (3) becomes equation (24) when R is set as follows.

$$\boldsymbol{R_a} = \sum_o A_o^{a\,\dagger}|\boldsymbol{s}_{\mathbf{Target}}\rangle\langle \boldsymbol{s}_{\mathbf{Target}}|\ A_o^a \tag{25}$$

3.7. Task 2: energy minimization

Energy minimization is a task to design a quantum circuit to generate a state which has a lowest energy of a molecule. When the state of a molecule is $|s$ and the Hamiltonian of a molecule is H, the energy is calculated by $s|H|\ s$. To minimize this energy, reward is calculated by

$$\begin{aligned} r(|\boldsymbol{s}\rangle,\ a) = -1\ \times\ \mathbb{E}_o[\text{Energy}] \\ = -1\ \times \sum_o Pr(o||\boldsymbol{s}\rangle,\ \ a)\langle \boldsymbol{s}'|^H|\boldsymbol{s}'\rangle \\ = -1\ \times \sum_o \langle \boldsymbol{s}|\ A_o^{a\,\dagger} H A_o^a|\boldsymbol{s}\rangle = \langle \boldsymbol{s}|\ - \sum_o A_o^{a\,\dagger} H A_o^a|\boldsymbol{s}\rangle \end{aligned} \tag{26}$$

In QOMDP, as reward is calculated by equation (3). Equation (3) becomes equation (26) when R is set as follows.

$$\boldsymbol{R_a} = -\textstyle\sum_o A_o^{a\dagger} H A_o^a \tag{27}$$

4. Results and discussions

4.1. Task 1: state preparation

We show the results where the target state $|s_{\mathrm{Target}}$ is the Bell state or the GHZ state. The simulations are conducted by using Qiskit library [A21]. The experiments were averaged over 10 different random seeds. The hyperparameters were set as follows. The maximum number of steps in circuit design is 100, the threshold of fidelity to end an episode is 0.99, the hyperparameters for the value iteration algorithm in Algorithm 2 are 10 for horizon H, 9 for number of iterations I, and 10 for the minimum initial size of the state set N. To evaluate the results, state generation is executed 100 times using the obtained policy after the update is completed for each iteration. The evaluation is performed by averaging the obtained fidelity and the number of steps taken. Higher fidelity and fewer steps are better.

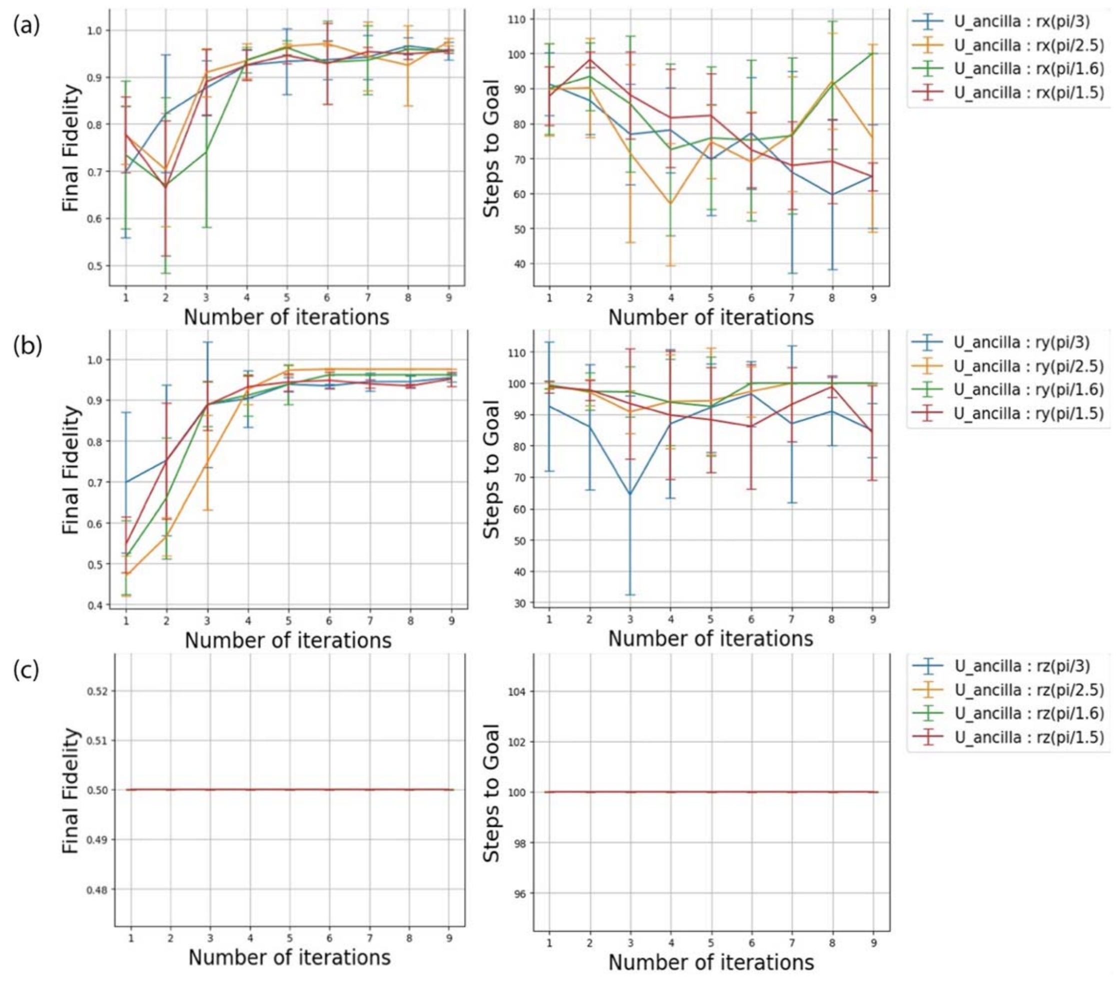


Figure 4. Planning result when the target state is the Bell state. The final fidelity and the number of steps to reach the goal are plotted with respect to the number of planning iterations. The error bar stands for one standard deviation over 10 random seeds. (a). U_ancilla is an Rx rotation. (b). U_ancilla is an Ry rotation. (c). U_ancilla is an Rz rotation.

First, we describe the case where the target state is a Bell state. The planning result is shown in figure 4. The experiment was performed by changing the U_ancilla gate, applied for 'Ancilla' in figure 3(b), to {Rx, Ry, Rz} gates with the rotation angles $\{\pi/3, \pi/2.5, \pi/1.6, \pi/1.5\}$. The horizontal axis shows the number of iterations of the value iteration method, and the vertical axis shows the fidelity gotten and the number of steps taken when the circuit design was executed by the policy obtained from the iterations. Figures 4(a)–(c) show the results when the U_ancilla gate is changed to the Rx, Ry, and Rz gates with various rotation angles. One example of the circuit obtained by performing the QOMDP planning algorithm is shown in figure 5(a). The state obtained when the circuit was executed is shown in figure 5(b).

Second, we describe the case where the target state is the GHZ state and the number of qubits of 'System' is 3. Planning result is shown in figure 6. The experiment was performed by changing the U_ancilla gate in figure 3(b) to {Rx, Ry, Rz} gates with the rotation angles $\{\pi/3, \pi/2.5, \pi/1.6, \pi/1.5\}$. The horizontal axis shows the number of iterations of the value iteration method, and the vertical axis shows the fidelity gotten and the number of steps taken when the circuit design was executed by the policy obtained from the iterations. Figures 6(a)–(c) show the results when the U_ancilla gate is changed to the Rx, Ry, and Rz gates and various rotation angles. One example of the generated circuits is shown in figure 7(a). The state obtained when the circuit was executed is shown in figure 7(b).

Third, we describe the case where the target state is the 4-qubit GHZ state. The presentation is similar to that of the 3-qubit case. The data is depicted in figure 8. The circuit and the generated density matrix are depicted in figure 9.

In figures 4(c), 6(c), and 8(c), we observe that the learning curves for Bell-GHZ states are constant functions and are independent of the ancilla rotation angle ϕ if $U_{ancilla}(\phi) = R_z(\phi)$. We also observe similar behavior in figure 6(a) for the case when $U_{ancilla}(\phi) = R_x(\phi)$ and number of system qubits $n_s = 3$, but not for $n_s = 2$ and $n_s = 4$. To explain these observations, we introduce a lemma. The proof for the lemma is provided in appendix.

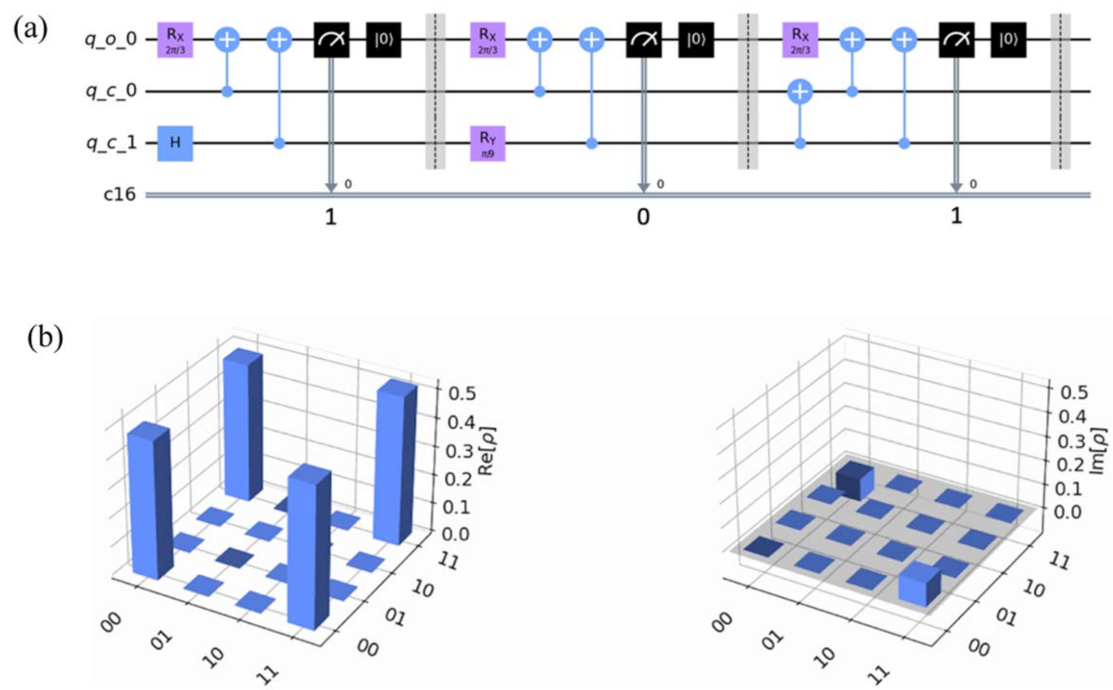


Figure 5. Execution result of state preparation for Bell state. (a). The quantum circuit. (b). The cityscape diagrams for the real and the imaginary parts of the density matrix.

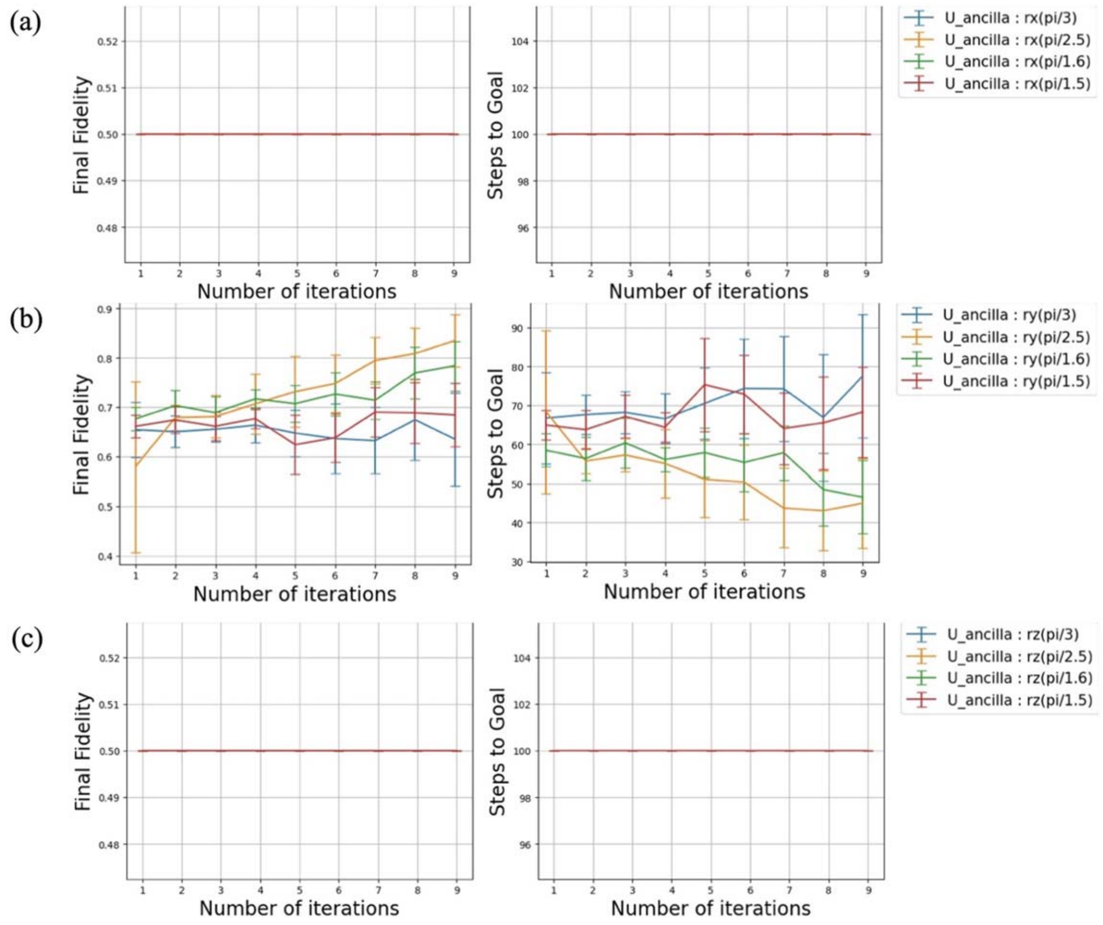


Figure 6. Planning result when the target state is the GHZ state and the number of qubits of system is 3. The final fidelity and the number of steps to reach the goal are plotted with respect to the number of planning iterations. The error bar stands for one standard deviation over 10 random seeds. (a). U_ancilla is an Rx rotation. (b). U_ancilla is an Ry rotation. (c). U_ancilla is an Rz rotation.

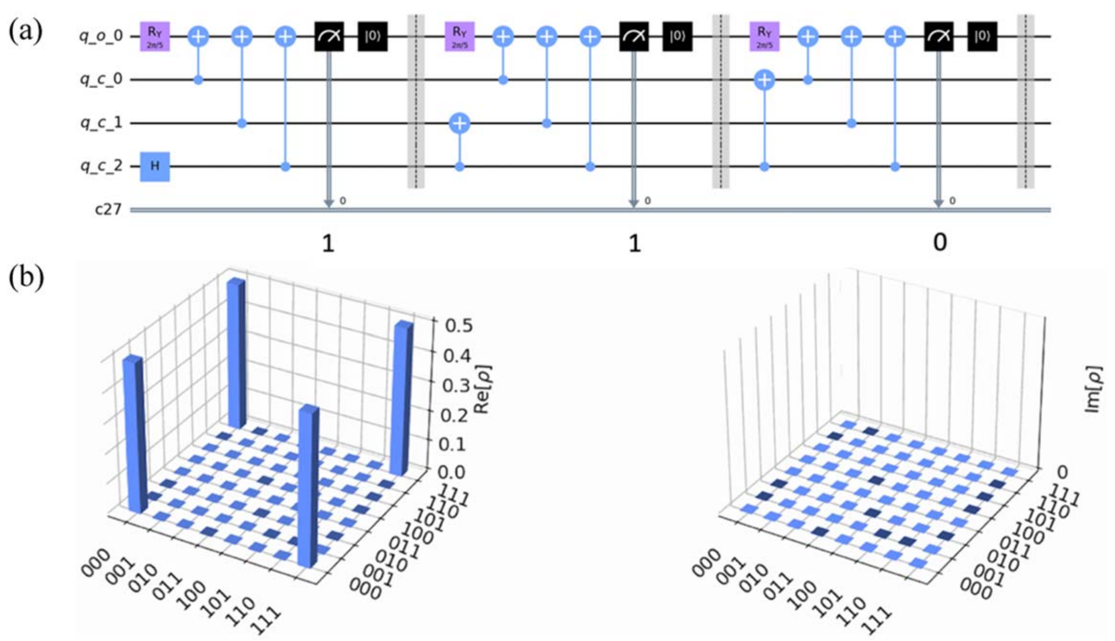


Figure 7. Execution result of state preparation for 3-qubits GHZ state. (a). The quantum circuit. (b). The cityscape diagrams for the real and the imaginary parts of the density matrix.

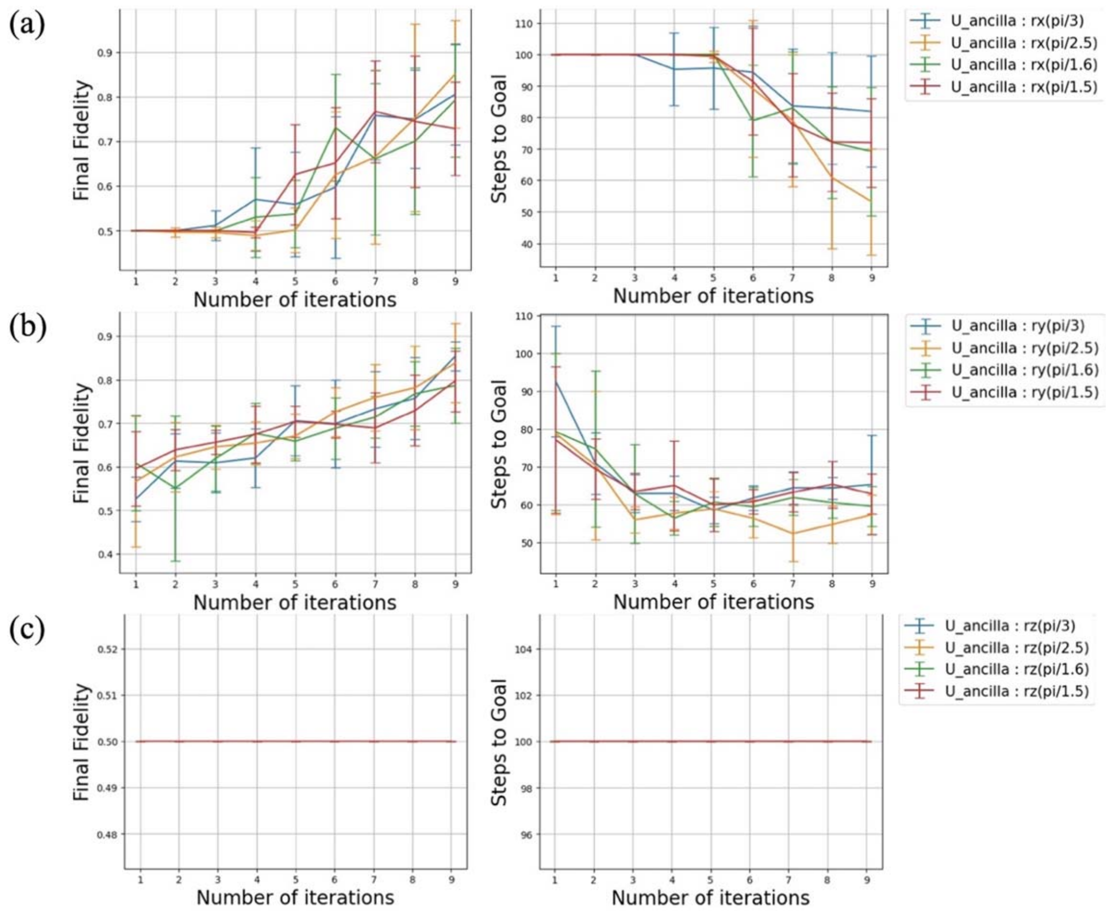


Figure 8. Planning result when the target state is the GHZ state and the number of qubits of system is 4. The final fidelity and the number of steps to reach the goal are plotted with respect to the number of planning iterations. The error bar stands for one standard deviation over 10 random seeds. (a). U_ancilla is an Rx rotation. (b). U_ancilla is an Ry rotation. (c). U_ancilla is an Rz rotation.

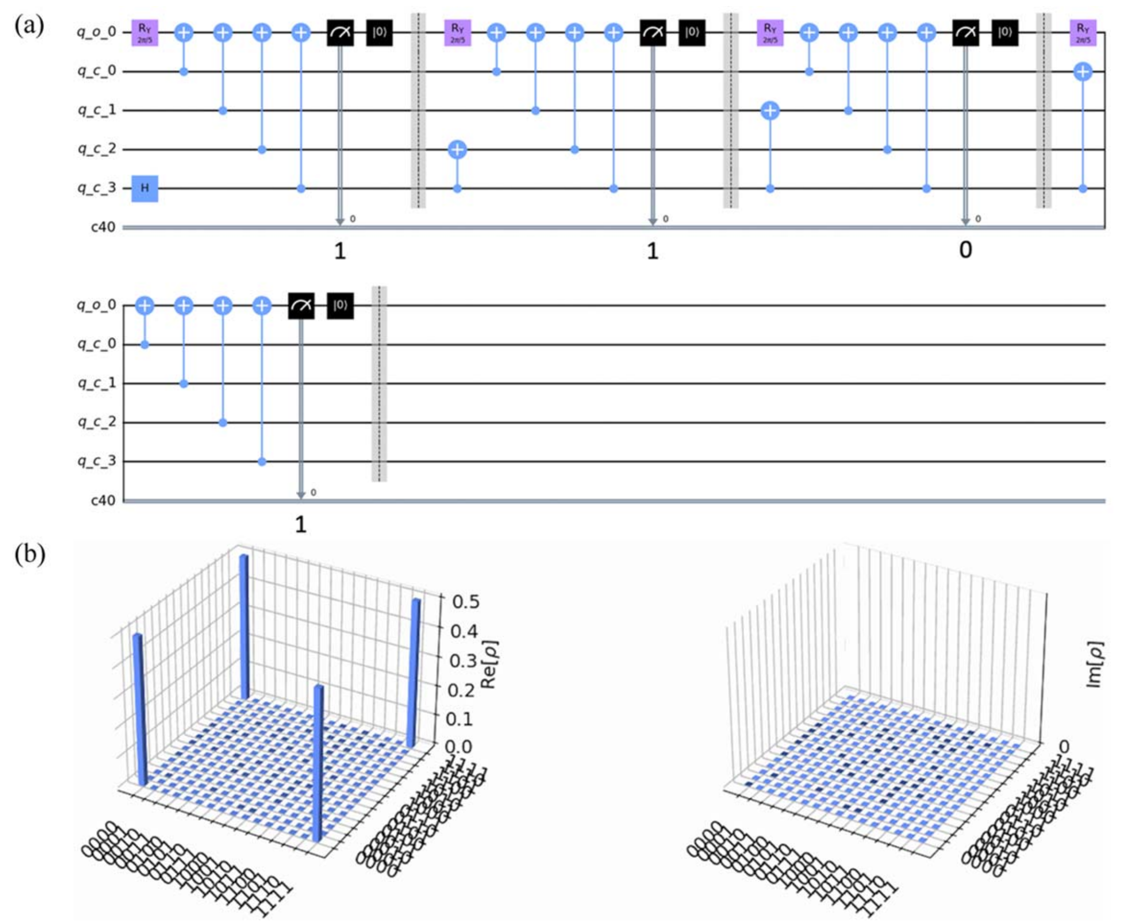


Figure 9. Execution result of state preparation for 4-qubits GHZ state. (a). The quantum circuit. (b). The cityscape diagrams for the real and the imaginary parts of the density matrix.

4.2. Lemma

We use the notations $I = (i_{n_s},\ldots,i_1)$ for a system n_s-bit string and $f(I) = i_{n_s} \bigoplus \ldots \bigoplus i_1$ for the parity function, where $\bigoplus$ is the XOR operation. $|I = |i_{n_s},\ldots,i_1$ denotes the n_s-qubit system state in the computational basis. Consider the circuit in figures 3(a) and (b). For any iteration time step t and any history sequence $h_t = \{a_0,\ o_1,\ a_1,\ o_2,\ \ldots,\ a_{t-1},\ o_t\}$, the mid-circuit observation probability is

$$\Pr(o_{t+1}) = \begin{cases} \sum_{\{I:f(I)=o_{t+1}\}} I|U_a(a_t)\rho_s(t)U_a^\dagger(a_t)|I, \text{ if } U_{ancilla}(\phi) = R_z(\phi) \\ \sum_I I|U_a(a_t)\rho_s(t)U_a^\dagger(a_t)|I\left(\cos\dfrac{\phi}{2}\right)^{2[1-(o_{t+1}\oplus f(I))]}\left(\sin\dfrac{\phi}{2}\right)^{2(o_{t+1}\oplus f(I))},\ \text{ if } U_{ancilla}(\phi) \\ = R_x(\phi) \end{cases},$$

where $\rho_s(t)$ is the system density matrix from previous time step. Notice that $\rho_s(t)$ is well-defined because of the measurement and resetting operation on the ancilla.

With the Lemma, we could explain the observed learning curves. If the ancilla rotation is $R_z(\phi)$, then the observation probability is independent of ϕ. This explains why the learning curves for $U_{ancilla}(\phi) = R_z(\phi)$ are independent of ϕ in figures 4(c), 6(c), and 8(c). We further notice that if the first action is one Hadamard gate H acting on any one of the system qubits (which is the standard first step to generate Bell-GHZ state), then $U_a(a_0)\rho_s(0)U_a^\dagger(a_0) = H_i \mid 0\rangle^{\otimes n_s}\langle 0 \mid^{\otimes n_s} H_i$ and hence $\Pr(o_{t+1}) = \dfrac{1}{2}$. Hence the observation provides no information regarding the system, and the agent can learn nothing about the system with the observation. This explains why the learning curves are constant in figures 4(c), 6(c), and 8(c).

To explain figure 6(a), notice that if $I|U_a(a_t)\rho_s(t)U_a^\dagger(a_t)|I$ has equal weight for odd-parity sector and even-parity sector, then the observation probability is $\Pr(o_{t+1}) = \dfrac{1}{2}$ for $U_{ancilla}(\phi) = R_x(\phi)$. Furthermore, n-qubit

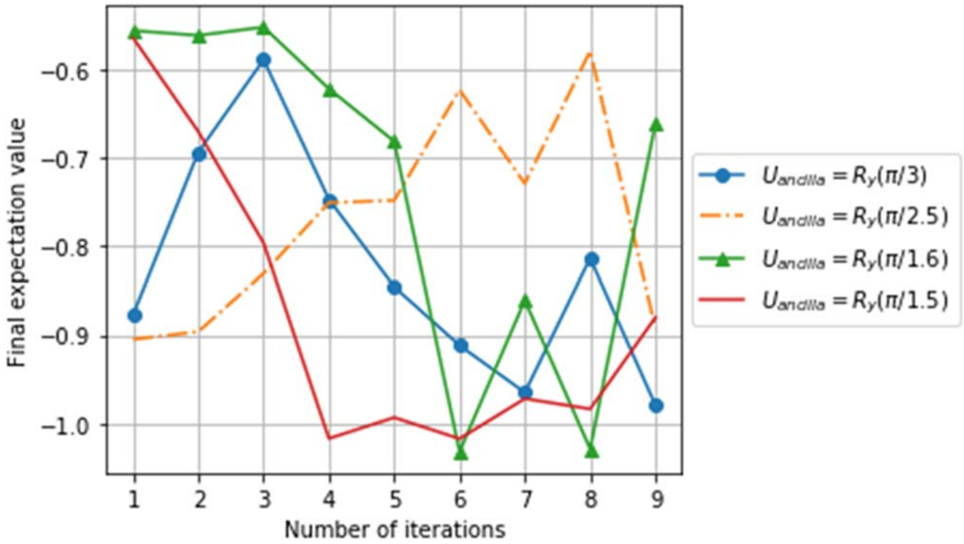


Figure 10. Planning result when the molecule H2 with bond length 1.0Å. Final energy expectation value (Hartree) is plotted with respect to the number of planning iterations. Average results of 100 executions are reported.

Bell-GHZ states are equal superpositions of all-zero state $|0\rangle^{\otimes n_s}$ and all-one state $|1\rangle^{\otimes n_s}$. All-zero state always has even parity, while all-one state has the same parity as the number of system qubit n_s. This implies that for $n_s = 3$, the target Bell-GHZ state is an equal superposition of odd-parity state and even-parity state. The agent would not be able to distinguish the target state from the system $U_a(a_0)\rho_s(0)U_a^\dagger(a_0) = H_i|0\rangle^{\otimes n_s}\langle 0|^{\otimes n_s} H_i$ after the action of one Hadamard gate when $n_s = 3$. This problem does not exist for $n_s = 2$ and $n_s = 4$ cases.

4.3. Task 2: energy minimization

We show the experimental results of H2 and H-He+. The Hamiltonians of the molecules are derived using OpenFermion [BM]. In these experiments, the orbital basis is STO-3G and the Fermion-qubit transformation is Jordan-Wigner. Since the minimum energy of the molecule is not known in advance in the energy minimization experiment, it is difficult to set the threshold value. Therefore, in this experiment, the episode ends when the number of steps reaches the maximum step.

The hyperparameters were set as follows. The maximum number of steps in circuit design is 10. The hyperparameters for the value iteration algorithm in Algorithm 2 are 10 for horizon H, 9 for number of iterations I, and 10 for the minimum initial size of the state set N. The energy unit is Hartree for all the experiments. Regarding the threshold value of the energy expectation value that is the condition for the end of the episode, as mentioned above, the minimum energy is not known in advance and it is difficult to set it. Therefore, we set this threshold value to a value that can never be reached. The episode ends only with the maximum number of steps. The set thresholds were −2 for H2 and −10 for H-He+.

To evaluate the results, the energy minimization circuit design is executed 100 times using the obtained policy after the update is completed for each iteration. The evaluation is performed by averaging the obtained energy expectation values. The smaller energy expectation value is better.

First, we describe the energy minimization circuit design experiment for H2. The planning result for H2 with bond length 1.0Å is shown in figure 10. The experiment was performed with the U_ancilla gate in figure 3(b) as Ry gate and the rotation angles are $\{\pi/3, \pi/2.5, \pi/1.6, \pi/1.5\}$. The horizontal axis shows the number of iterations of the value iteration method, and the vertical axis shows final energy expectation value when the circuit design was executed by the policy obtained from the iterations. Figure 10 demonstrate that our algorithm can solve simple quantum circuit design problem if suitable ancilla unitary is chosen. One example of the generated circuit is shown in figure 11(a). The change of energy expectation value when the circuit was executed is shown in figure 11(b). In figure 11(b), the horizontal axis shows the number of steps in the episode, and the vertical axis shows the energy expectation value.

Next experiment was conducted with the bond length spaced by 0.1 Å from 0.2 Å to 3.0 Å. The result is shown in figure 12. The experiment was performed with the U_ancilla gate in figure 3(b) as Ry gate and the rotation angles changed to $\{\pi/3, \pi/2.5, \pi/1.6, \pi/1.5\}$. The horizontal axis shows the bond length, and the vertical axis shows the energy value. Since the policy can be obtained at each rotation angle and each iteration,

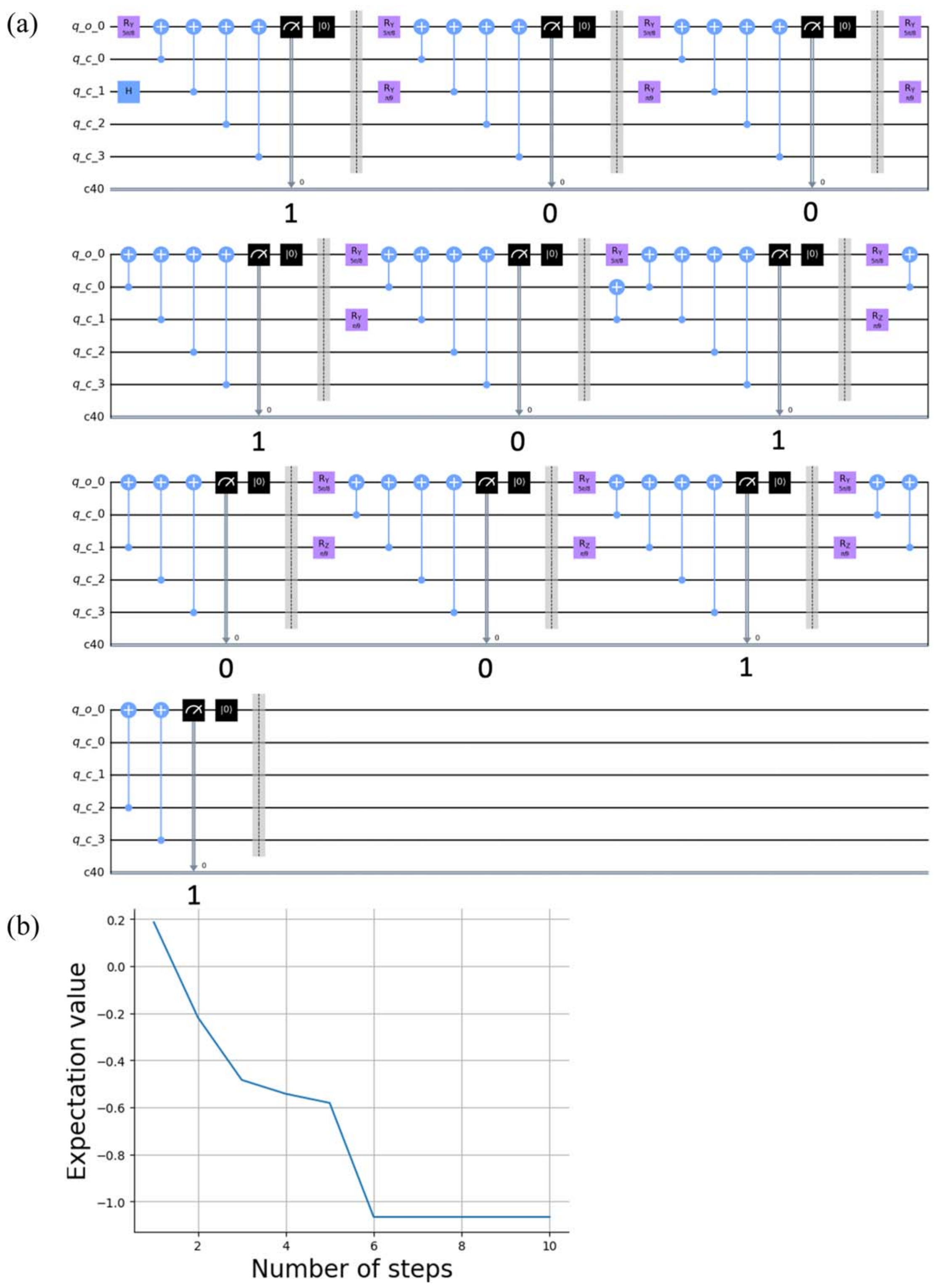


Figure 11. Single execution result of energy minimization for H2 bond length 1.0Å. (a). The quantum circuit. (b). Energy expectation value (Hartree) is plotted with respect to the number of steps.

the best policy is the one with the smallest average energy expectation value over four possible angles and nine iterations. In figure 12, the average and minimum energy expectation value when the circuit design is executed 100 times using the best policy are plotted. The exact minimum energy obtained by diagonalizing the Hamiltonian of H2 is also plotted in figure 12. In figure 12, the minimum energy obtained by our method is represented by black dots, and the exact minimum energy is represented by orange line. A kink is observed around 1.5 Å of QOMDP curve. Similar phenomena appear in VQE calculation for LiH molecule [KMT17].

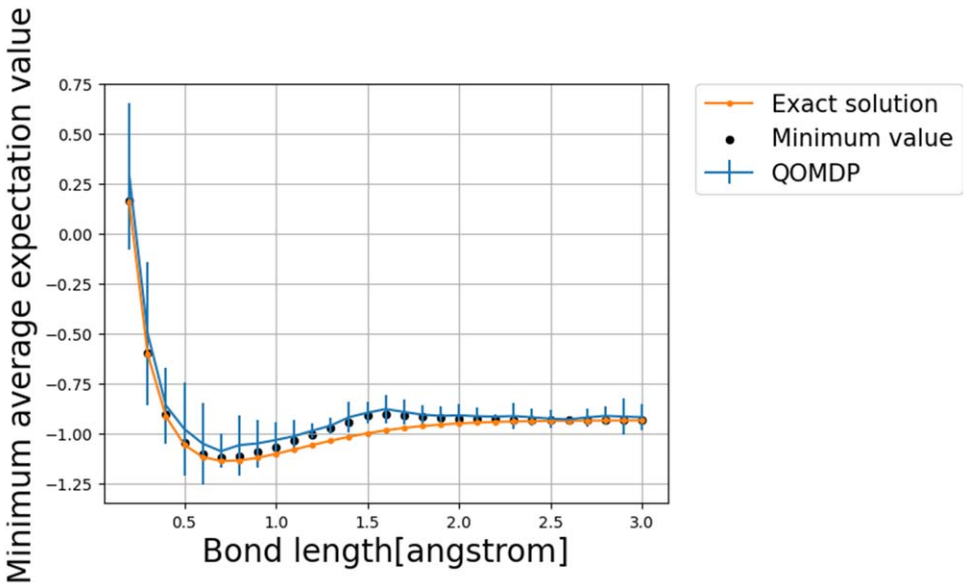


Figure 12. Minimum energy of H2. Energy value (Hartree) is plotted with respect to the bond length. Blue line is energy expectation value gotten by the best policy. Black dot is the minimum value of executions. Orange line is the exact minimum value. Error bar denotes one standard deviation over 100 executions.

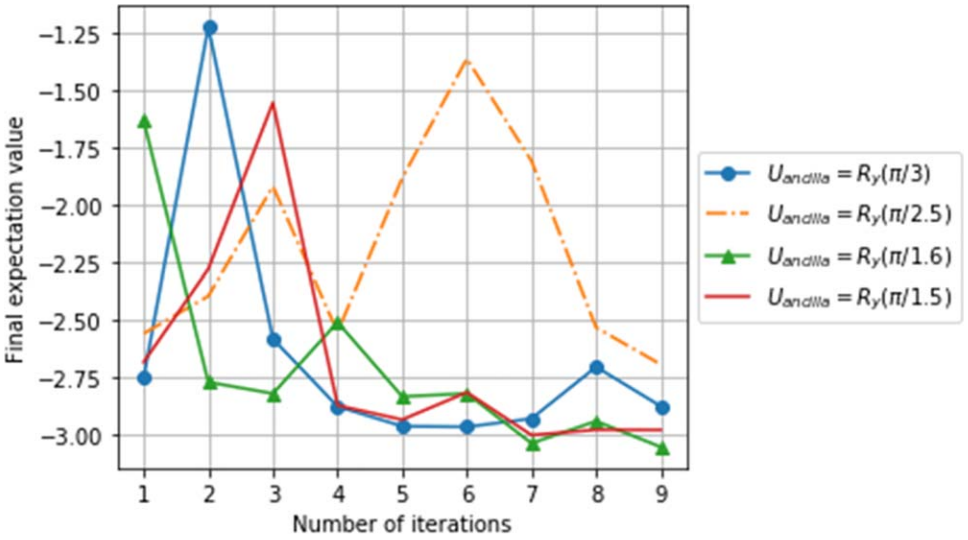


Figure 13. Planning result when the molecule H-He + with bond length 1.0Å. Final energy expectation value (Hartree) is plotted with respect to the number of planning iterations. Average results of 100 executions are reported.

This might due to incorrectness of the spin wavefunction [STS20], which might be improved by modifying the QOMDP action space. Notice that VQE algorithm has polynomial complexity with respect to number of qubits, while our algorithm requires exponentially large classical planning. VQE simulation achieving chemical accuracy (0.0016 Hartree) for H2 molecules has been reported [KMT17] with <10 circuit depth and <10000 function calls. Our algorithm fails to reach high accuracy around bond length 1.5 Å. The potential advantage of our algorithm is that the agent could automatically search for the ansatz instead of human-design ansatz based on prior knowledge. However, there is no guarantee that the QOMDP agent can always find the global minimum.

Second, we describe the case where the molecule is H-He+. The presentation is similar to that of the H2. The planning result and the execution result are depicted in figures 13 and 14. The minimum energy is depicted in figure 15. Figure 15 shows that for all bond lengths, the minimum energy gotten by QOMDP, represented by the black dots, is almost the same as the exact minimum energy represented by orange line.

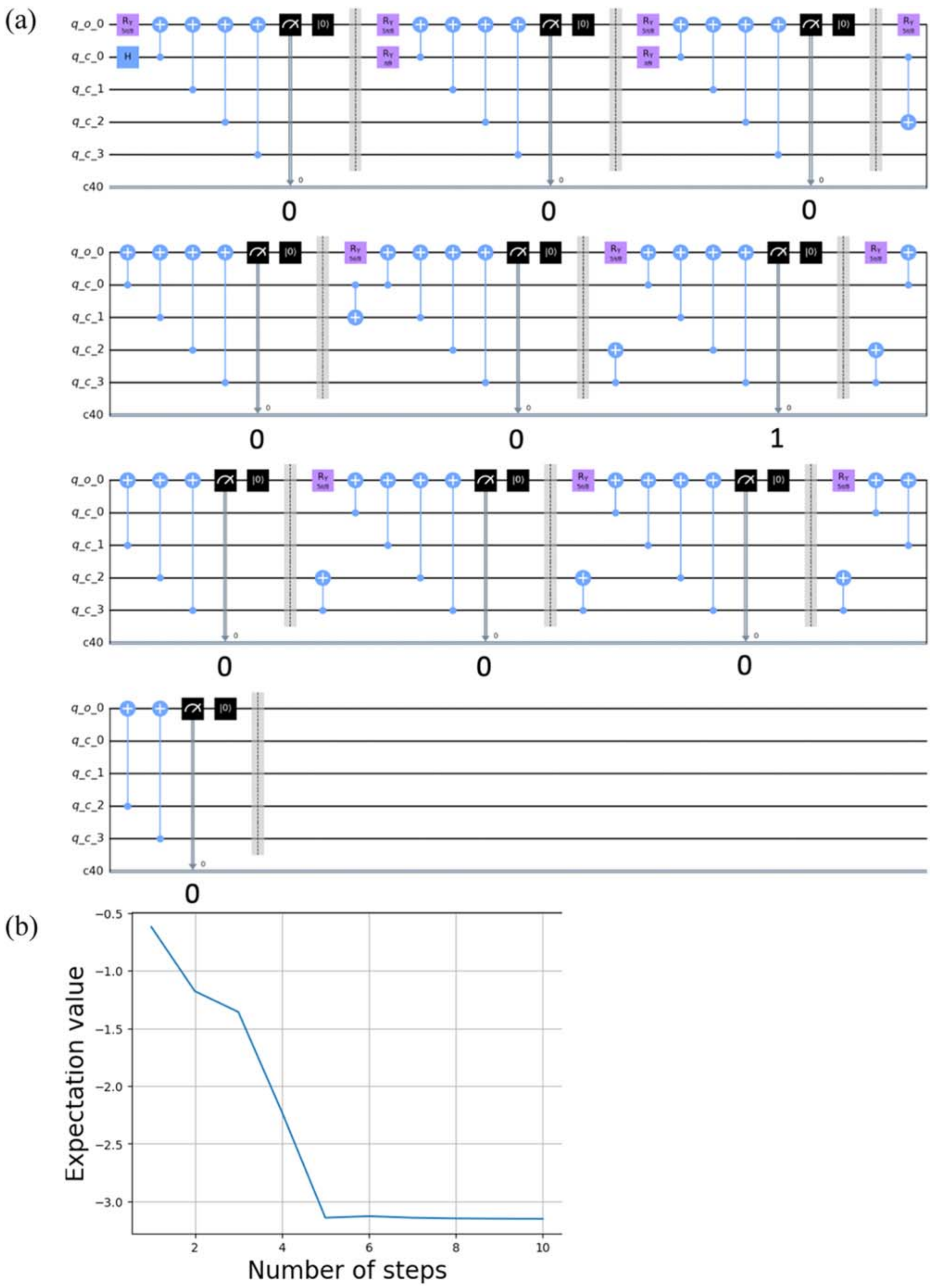


Figure 14. Single execution result of energy minimization for H-He + bond length 1.0Å. (a) The quantum circuit. (b) Energy expectation value (Hartree) is plotted with respect to the number of steps.

5. Conclusion

In this work, a QOMDP based planning algorithm is designed to solve for the problem of quantum circuit architecture search. Point-based approximation is used to resolve the intractability due to the planning history and the continuous Hilbert space. We implement the algorithm, and the simulation results suggest that the algorithm can successfully find circuits to produce entangled states and to minimize energy functionals for simple molecules. Our algorithm only requires small number of readouts from quantum circuits for online decision making. However, it costs exponentially large classical resources in the planning stage of the algorithm. One possible approach to scale up our method is to equip the classical agent with a tensor network simulator [CHHGK21] to tackle the exponentially scaling with respect to the circuit width. Future investigations are required to make the method suitable for large scale quantum computations.

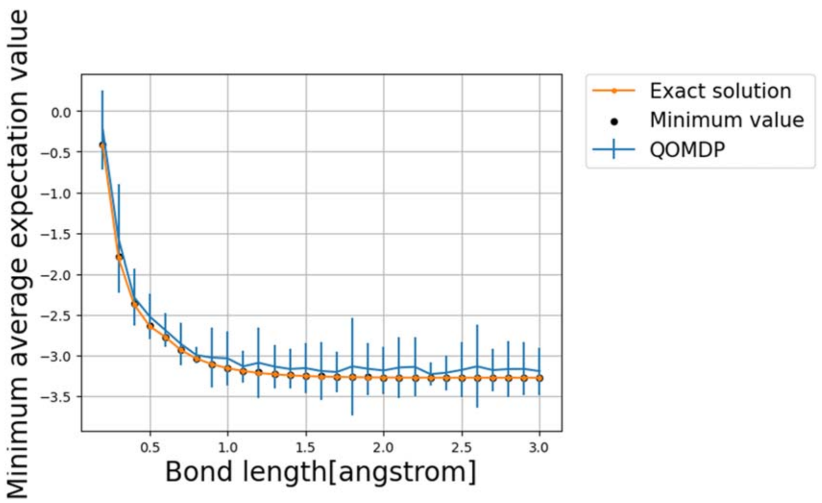


Figure 15. Minimum energy of H-He+. Energy value (Hartree) is plotted with respect to the bond length. Blue line is energy expectation value gotten by the best policy. Black dot is the minimum value of executions. Orange line is the exact minimum value. Error bar denotes one standard deviation over 100 executions.

Acknowledgments

The authors gratefully acknowledge the funding from Information-technology Promotion Agency (IPA) under MITOU TARGET program 2021. We thank Naoki Yamamoto for valuable discussions.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/tomo920/QCArchitectrue-QOMDP.

Appendix

The derivation for value iteration equation (9) is provided here.

$$V_q^{\pi}(|s\rangle) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(|\, s_t\rangle,\, a_t)||\, s_0\rangle = |s\rangle]$$
$$= \sum_t \gamma^t \mathbb{E}[r(|\, s_t\rangle,\, a_t)||\, s_0\rangle = |s\rangle]$$
$$= \sum_t \gamma^t \sum_{a_t} \sum_{h_t} \Pr(h_t,\, a_t||\, s_0\rangle = |s\rangle) r(S(h_t,\, |s\rangle),\, a_t)$$
$$= \sum_t \gamma^t \sum_{a_t} \sum_{h_t} \Pr(a_t|h_t,\, |\, s_0\rangle = |s\rangle)\Pr(h_t||s\rangle) r(S(h_t,\, |s\rangle),\, a_t)$$
$$= \sum_t \gamma^t \sum_{a_t} \sum_{h_t} \pi(a_t|S(h_t,\, |s\rangle))$$
$$\times \prod_{i=1}^{t} \Pr(o_{t-i+1}|S(h_{t-i},\, |s\rangle),\, a_{t-i})\pi(a_{t-i}|S(h_{t-i},\, |s\rangle))r$$
$$\times\left(\frac{\prod_{i=1}^{t} A_{o_{t-i+1}}^{a_{t-i}}}{\sqrt{\prod_{i=1}^{t} \Pr(o_{t-i+1}\mid S(h_{t-i}, |s\rangle), a_{t-i})}}|s\rangle,\, a_t\right) = \sum_t \gamma^t \sum_{a_t} \sum_{h_t} \pi(a_t|S(h_t,\, |s\rangle)) \prod_{i=1}^{t} \Pr(o_{t-i+1}|S(h_{t-i},\, |s\rangle),\, a_{t-i})\pi(a_{t-i}|S(h_{t-i},\, |s\rangle))$$
$$\times \frac{\langle s\mid (\prod_{i=1}^{t} A_{o_{t-i+1}}^{a_{t-i}})^{\dagger} R_{a_t} \prod_{i=1}^{t} A_{o_{t-i+1}}^{a_{t-i}}|s\rangle}{\prod_{i=1}^{t} \Pr(o_{t-i+1}\mid S(h_{t-i}, |s\rangle), a_{t-i})}$$
$$= \sum_t \gamma^t \sum_{a_t} \sum_{h_t} \pi(a_t|S(h_t,\, |s\rangle)) \prod_{i=1}^{t} \pi(a_{t-i}|S(h_{t-i},\, |s\rangle))$$
$$\times\langle s|\, (\prod_{i=1}^{t} A_{o_{t-i+1}}^{a_{t-i}})^{\dagger} R_{a_t} \prod_{i=1}^{t} A_{o_{t-i+1}}^{a_{t-i}}|s\rangle = \langle s|\sum_t \sum_{a_t} \sum_{h_t} \gamma^t \prod_{i=0}^{t} \pi(a_{t-i}|S(h_{t-i},\, |s\rangle))(\prod_{i=1}^{t} A_{o_{t-i+1}}^{a_{t-i}})^{\dagger} R_{a_t} \prod_{i=1}^{t} A_{o_{t-i+1}}^{a_{t-i}}|s\rangle$$
$$= \langle s|\, \Upsilon(\pi) \mid s\rangle. \tag{A1}$$

The Lemma is presented and proved here.

Lemma

We use the notations $I=(i_{n_s},\ldots,i_1)$ for a system n_s-bit string and $f(I)=i_{n_s}\bigoplus\ldots\bigoplus i_1$ for the parity function, where $\bigoplus$ is the XOR operation. $|I\rangle=|\,i_{n_s},\ldots,i_1\rangle$ denotes the n_s-qubit system state in the computational basis. Consider the circuit in figures 3*(a) and (b). For any iteration time step t and any history sequence $h_t=\{a_0,\,o_1,\,a_1,\,o_2,\,\ldots\,,a_{t-1},\,o_t\}$, the mid-circuit observation probability is*

$$\Pr(o_{t+1})=\begin{cases}\displaystyle\sum_{\{I:f(I)=o_{t+1}\}}\langle I|U_a(a_t)\rho_s(t)U_a^{\dagger}(a_t)|I\rangle,\ \text{if}\quad U_{ancilla}(\phi)=R_z(\phi)\\ \displaystyle\sum_I\langle I|U_a(a_t)\rho_s(t)U_a^{\dagger}(a_t)|I\rangle\left(cos\frac{\phi}{2}\right)^{2[1-(o_{t+1}\bigoplus f(I))]}\left(sin\frac{\phi}{2}\right)^{2(o_{t+1}\bigoplus f(I))},\ \text{if}\quad U_{ancilla}(\phi)=R_x(\phi)\end{cases}$$

where $\rho_s(t)$ is the system density matrix from previous time step. Notice that $\rho_s(t)$ is well-defined because of the measurement and resetting operation on the ancilla.

Proof:

We use the convention that the Hilbert space is the tensor product *system*$\bigotimes$*ancilla*, where the system part has n_s qubits and the ancillary part has one qubit. The total density matrix from the previous time step is $\rho_s(t)\bigotimes|\,0\rangle\langle 0\,|$. The system and ancillary unitary is $U_a(a_t)\bigotimes U_{ancilla}(\phi)=U_a(a_t)\bigotimes R_\alpha(\phi)$, where $\alpha\in\{x,\,y,\,z\}$. The entangler is $\sum_I|I\rangle\langle I|\bigotimes X^{f(I)}$. The measured density matrix can be calculated to be

$$\rho'=\left(\sum_I|\,I\rangle\langle I\,|\bigotimes X^{f(I)}\right)\left(U_a(a_t)\rho_s(t)U_a^{\dagger}(a_t)\bigotimes R_\alpha(\phi)\,|\,0\rangle\langle 0\,|\,R_\alpha^{\dagger}(\phi)\right)\left(\sum_J|\,J\rangle\langle J\,|\bigotimes X^{f(J)}\right)$$
$$=\sum_{I,J}|\,I\rangle\langle I\,|\,U_a(a_t)\rho_s(t)U_a^{\dagger}(a_t)\,|\,J\rangle\langle J\,|\bigotimes X^{f(I)}R_\alpha(\phi)\,|\,0\rangle\langle 0\,|\,R_\alpha^{\dagger}(\phi)X^{f(J)}. \tag{A2}$$

Hence the observation probability is

$$\Pr(o_{t+1})=Tr[\mathbb{I}_{n_s}\bigotimes|\,o_{t+1}\rangle\langle o_{t+1}\,|\,\rho']$$
$$=\sum_I\langle I\,|\,U_a(a_t)\rho_s(t)U_a^{\dagger}(a_t)\,|\,I\rangle\,|\,\langle o_{t+1}\,|\,X^{f(I)}R_\alpha(\phi)\,|\,0\rangle\,|^2. \tag{A3}$$

If $R_\alpha(\phi)=R_z(\phi)$, then $|\,\langle o_{t+1}\,|\,X^{f(I)}R_z(\phi)\,|\,0\rangle\,|^2=\delta_{o_{t+1},f(I)}$

If $R_\alpha(\phi)=R_x(\phi)$,

then $|\,\langle o_{t+1}\,|\,X^{f(I)}R_x(\phi)\,|\,0\rangle\,|^2=|\,\langle o_{t+1}\,|\,R_x(\phi)|f(I)\rangle\,|^2=\left(\cos\frac{\phi}{2}\right)^{2[1-(o_{t+1}\bigoplus f(I))]}\left(\sin\frac{\phi}{2}\right)^{2(o_{t+1}\bigoplus f(I))}.$ Q.E.D.

ORCID iDs

Chih-Chieh Chen https://orcid.org/0000-0003-3092-4346
Tomah Sogabe https://orcid.org/0000-0001-9258-6130

References

[A18] Aaronson S 2018 Shadow tomography of quantum states *Proc. of the 50th Annual ACM SIGACT Symp. on Theory of Computing (STOC 2018) (New York, NY, USA)* pp. 325–38

[A21] ANIS M S *et al* 2021 Qiskit: An Open- source Framework for Quantum Computing https://qiskit.org/

[AAB19] Arute F *et al* 2019 Quantum supremacy using a programmable superconducting processor *Nature* **574** 505–10

[AAH16] Alfailakawi M G, Ahmad I and Hamdan S 2016 Harmony-search algorithm for 2D nearest neighbor quantum circuits realization *Expert Syst. Appl.* **61** C (November 2016) 16–27

[SJD22] Skolik A, Jerbi S and Dunjko V 2022 Quantum agents in the Gym: a variational quantum algorithm for deep Q-learning *Quantum* **6** 720

[AU22] Aho A and Ullman. J 2022 Abstractions, their algorithms, and their compilers *Commun. ACM* **65** 76–91

[B18] Bukov M, Day A G R, Sels D, Weinberg P, Polkovnikov A and Mehta P 2018 Reinforcement learning in different phases of quantum control *Phys. Rev. X* **8** 031086

[BAHH21] Baum Y, Amico M, Howell S, Hush M, Liuzzi M, Mundada P, Merkh T, Carvalho A R R and Biercuk M J 2021 Experimental deep reinforcement learning for error-robust gateset design on a superconducting quantum computer *PRX Quantum* **2** 040324

[BBA14] Barry J, Barry D T and Aaronson S 2014 *Phys. Rev. A* **90** 032311

[BM] Babbush R and McClean J Announcing OpenFermion: The Open Source Package for Quantum Computers', Google AI Blog, (https://ai.googleblog.com/2017/10/announcing-openfermion-open-source.html)

[BSK21] Borah S, Sarma B, Kewming M, Milburn G J and Twamley J 2021 *Phys. Rev. Lett.* **127** 190403 Published 2 November

[C16] Cidre G A 2016 *Planning in a Quantum System.* (PA: Carnegie Mellon University Pittsburgh)

[C20] Chen S Y-C, Yang C-H H, Qi J, Chen P-Y, Ma X and Goan H-S 2020 Variational quantum circuits for deep reinforcement learning *IEEE Access* **8** 141007–24

[C21] Cerezo M *et al* 2021 Variational quantum algorithms *Nat Rev Phys* **3** 625–44

[CHHGK21] Chen S Y-C, Huang C-M, Hsing C-W, Goan H-S and Kao Y-J 2022 Variational Quantum Reinforcement Learning Via Evolutionary Optimization *Mach. Learn.: Sci. Technol.* **3** 015025

[DB18] Dunjko V and Briegel. H J 2018 Machine learning & artificial intelligence in the quantum domain: a review of recent progress *Rep. Prog. Phys.* **81** 074001

[FM17] Farghadan A and Mohammadzadeh N 2017 Quantum circuit physical design flow for 2D nearest-neighbor architectures *Int. J. Circuit Theory Appl.* **45** 989–1000

[G96] Grover L K 1996 A fast quantum mechanical algorithm for database search *Proc. of the twenty-eighth annual ACM symposium on Theory of Computing (STOC '96) (New York, NY, USA)* pp. 212–9

[GHZ89] Greenberger D M, Horne M A and Zeilinger A Going Beyond Bell's Theorem *arXiv* 0712.0921[quant-ph]

[HCT19] Havlíček V *et al* 2019 Supervised learning with quantum-enhanced feature spaces *Nature* **567** 209–12

[HNYN11] Hirata Y, Nakanishi M, Yamashita S and Nakashima Y 2011 An efficient conversion of quantum circuits to a linear nearest neighbor architecture *Quantum Info. Comput.* **11** 142–66

[HWN21] He R H *et al* 2021 Deep reinforcement learning for universal quantum state preparation via dynamic pulse control *EPJ Quantum Technol.* **8** 29

[J21] Jerbi S, Gyurik C, Marshall S, Briegel H J and Dunjko V 2021 Parametrized quantum policies for reinforcement learning *NeurIPS 2021 Poster* (https://doi.org/10.48550/arXiv.2103.05577)

[K21] Kwak Y, Yun W J, Jung S, Kim J-K and Kim J 2021 Introduction to quantum reinforcement learning: theory and pennylane-based implementation 2108.06849 arXiv e-prints, p. arXiv

[K22] Kimura T 2022 https://github.com/tomo920/QCArchitectrue-QOMDP

[KCC13] Kormushev P, Calinon S and Caldwell D G 2013 Reinforcement learning in robotics: applications and real-world challenges *Robotics* **2** 122–48

[KFC21] Kuo E-J, Fang Y-L L and Chen S Y-C Quantum architecture search via deep reinforcement learning 2104.07715arXiv [quant-ph]

[KLC98] Kaelbling L P, Littman M L and Cassandra A R 1998 Planning and acting in partially observable stochastic domains *Artif. Intell.* **101** 99–134

[KSCS21] Kimura T, Shiba K, Chen C-C, Sogabe M, Sakamoto K and Sogabe T 2021 Variational quantum circuit-based reinforcement learning for pomdp and experimental implementation *Mathematical Problems in Engineering* Article ID 3511029 **11** 2021

[KMT17] Kandala A *et al* 2017 Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets *Nature* **549** 242–6

[LS20] Lockwood O and Si M 2020 Reinforcement learning with quantum variational circuits *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 16https://ojs.aaai.org/index.php/AIIDE/article/view/7437

[LS21] Lockwood O and Si M 2021 Playing atari with hybrid quantum–classical reinforcement learning *NeurIPS 2020 Workshop on Pre-registration in Machine Learning* in Proceedings of Machine Learning Research 148:285-301 https://proceedings.mlr.press/v148/lockwood21a.html

[LSJ15] Lin C-C, Sur-Kolay S and Jha N K 2015 PAQCS: physical design-aware fault-tolerant quantum circuit synthesis *IEEE Trans. Very Large Scale Integr. VLSI Syst.* **23** 1221–34

[MDW21] Mackeprang J, Dasari D B R and Wrachtrup J 2020 A reinforcement learning approach for quantum state engineering *Quantum Mach. Intell.* **2** 5

[MFM08] Maslov D, Falconer S M and Mosca. M 2008 Quantum circuit placement. trans *Comp.-Aided Des. Integ. Cir. Sys.* **27** 752–63

[MK21] Moll M and Kunczik L 2021 Comparing quantum hybrid reinforcement learning to classical methods *Hum.-Intell. Syst. Integr.* **3** 15–23

[MKS15] Mnih V *et al* 2015 Human-level control through deep reinforcement learning *Nature* **518** 529–33

[MLWEV21] Ostaszewski M, Trenkwalder L M, Masarczyk W, Scerri E and Dunjko V 2021 Reinforcement learning for optimization of variational quantum circuit architectures *arXiv* 2103.16089 [quant-ph] https://doi.org/10.48550/arXiv.2103.16089 2103.16089

[MNK18] Mitarai K, Negoro M, Kitagawa M and Fujii K 2018 Quantum circuit learning *Phys. Rev. A* **98** 032309

[MRB16] McClean J R, Romero J, Babbush R and Aspuru-Guzik A 2016 The theory of variational hybrid Quantum–classical algorithms *New J. Phys.* **18** 023023 New

[NBS19] Niu M Y *et al* 2019 Universal quantum control through deep reinforcement learning *npj Quantum Inf* **5** 33

[NC11] Nielsen M A and Chuang. I L 2011 *Quantum computation and quantum information (10th Anniversary Edition).* (New York: Cambridge University Press)

[NMM18] Negnevitsky V *et al* 2018 Repeated multi-qubit readout and feedback with a mixed-species trapped-ion register *Nature* **563** 527–31

[NY17] Nurdin H I and Yamamoto N 2017 Linear dynamical quantum systems *In Analysis, Synthesis, and Control.* (Cham, Switzerland: Springer International Publishing AG)

[PT87] Papadimitriou C H and Tsitsiklis J N 1987 The complexity of markov decision processes *Math. Oper. Res.* **12** 441–50

[P18] Preskill J 2018 Quantum computing in the NISQ era and beyond *Quantum* **2** 79

[PMS14] Peruzzo A *et al* 2014 A variational eigenvalue solver on a photonic quantum processor *Nat. Commun.* **5** 4213

[PT20] Pirhooshyaran M and Terlaky T 2021 *Quantum Machine Intelligence* **3** 25

[PGT03] Pineau J, Gordon G and Thrun. S 2003 Point-based value iteration: an anytime algorithm for POMDPs *Proc. of the 18th international joint conference on Artificial intelligence (IJCAI'03) (San Francisco, CA, USA)* (Morgan Kaufmann Publishers Inc.) 1025–30

[RN21] Russell S and Norvig P 2021 *Artificial Intelligence: A Modern Approach* (London, UK: Pearson Education) 4th

[SK19] Schuld M and Killoran. N 2019 Quantum machine learning in feature Hilbert spaces *Phys. Rev. Lett.* **122** 040504

[S94] Shor P W 1994 Algorithms for quantum computation: discrete logarithms and factoring *Proc. 35th Annual Symp. on Foundations of Computer Science, pp* 124–34

[SEL21] Sivak V V, Eickbusch A, Liu H, Royer B, Tsioutsios I and Devoret M H 2022 Model-free quantum control with reinforcement learning *Phys. Rev. X* **12** 011059

[SS73] Smallwood R D and Sondik E J 1973 The optimal control of partially observable markov processes over a finite horizon *Oper. Res.* **21** 5, 1071–88 INFORMS

[SSS17] Silver D *et al* 2017 Mastering the game of Go without human knowledge *Nature* **550** 354–9

[SV10] Silver D and Veness J 2010 Monte-carlo planning in large pomdps *Proc. of the 23rd Int. Conf. on Neural Information Processing Systems ser. NIPS* **10** 2164–72 Red Hook, NY, USA: Curran Associates Inc..

[SB18] Sutton R S and Barto A G 2018 *Reinforcement Learning: An Introduction* (Cambridge, MA, USA: The MIT Press)

[SBM06] Shende V V, Bullock S S and Markov I L 2006 Synthesis of quantum-logic circuits *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **25** 1000–10

[SSP14] Shafaei A, Saeedi M and Pedram M 2014 Qubit placement to minimize communication overhead in 2D quantum architectures *2014 19th Asia and South Pacific Design Automation Conf. (ASP-DAC)* 495–500

[STS20] Sugisaki K, Toyota K, Sato K, Shiomia D and Takui T 2020 A probabilistic spin annihi- lation method for quantum chemical calculations on quantum computers *Phys. Chem. Chem. Phys.* **22** 20990–4

[TBF05] Thrun S, Burgard W and Fox D 2005 *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)* (Cambridge, MA, USA: The MIT Press)

[WUZ17] Wang Y *et al* 2017 Single-qubit quantum memory exceeding ten-minute coherence time *Nature Photon* **11** 646–50

[WLQ21] Wang P *et al* 2021 Single ion qubit with estimated coherence time exceeding one hour *Nat. Commun.* **12** 233

[YC21] Ye E and Chen S Y-C Quantum architecture search via continual reinforcement learning *arXiv* 2112.05779[quant-ph]

[YFY21] Ying M S, Feng Y and Ying S G 2021 Optimal policies for quantum markov decision processes *Int. J. Autom. Comput.* **18** 410–21

[YT21] Yoshioka T and Tsai J S 2021 Fast unconditional initialization for superconducting qubit and resonator using quantum-circuit refrigerator *Appl. Phys. Lett.* **119** 124003

[YY18] Ying S G and Ying M S 2018 Reachability analysis of quantum Markov decision processes *Inf. Comput.* **263** 31–51

[ZHZY20] Zhang S-X, Hsieh C-Y, Zhang S and Yao H 2020 Differentiable quantum architecture searcharXiv:2010.08561

[ZWA19] Zhang X M *et al* 2019 When does reinforcement learning stand out in quantum control? a comparative study on state preparation *npj Quantum Inf* **5** 85