IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 8, 2021, accepted August 11, 2021, date of publication September 6, 2021, date of current version September 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3110652

Quantum Tree-Based Planning

ANDRÉ SEQUEIRA[1], LUIS PAULO SANTOS[1,2,3], (Associate Member, IEEE), AND LUIS SOARES BARBOSA[1,2,4]

[1]Department of Informatics, University of Minho, 4710-057 Braga, Portugal
[2]International Nanotechnology Laboratory (INL), 4715-330 Braga, Portugal
[3]CSIG, INESC TEC, 4200-465 Porto, Portugal
[4]HASLab, INESC TEC, 4200-465 Porto, Portugal

Corresponding author: André Sequeira (andresequeira401@gmail.com)

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020.

ABSTRACT Reinforcement Learning is at the core of a recent revolution in Artificial Intelligence. Simultaneously, we are witnessing the emergence of a new field: Quantum Machine Learning. In the context of these two major developments, this work addresses the interplay between Quantum Computing and Reinforcement Learning. Learning by interaction is possible in the quantum setting using the concept of oracularization of environments. The paper extends previous oracular instances to address more general stochastic environments. In this setting, we developed a novel quantum algorithm for near-optimal decision-making based on the Reinforcement Learning paradigm known as Sparse Sampling. The proposed algorithm exhibits a quadratic speedup compared to its classical counterpart. To the best of the authors' knowledge, this is the first quantum planning algorithm exhibiting a time complexity independent of the number of states of the environment, which makes it suitable for large state space environments, where planning is otherwise intractable.

INDEX TERMS Quantum computation, quantum reinforcement learning, sparse sampling.

I. INTRODUCTION

In Reinforcement Learning (RL), an agent interacts with the surrounding environment, to maximize its cumulative reward in expectation, due to the stochastic nature of the environment [1], as depicted in Figure 1.

In the context of *planning*, a RL agent has full knowledge about the dynamics of the environment, thus exploiting this information to reach the optimal policy π^*. Typically, *Dynamic Programming* methods [2] like *Policy Iteration* [3] are used to solve this problem. On the other hand, in *Model-Free RL* the agent learns purely by trial and error, typically resorting to sampling techniques. Since the environment is unknown to the agent, the latter faces a dilemma known as the *exploration-exploitation trade-off*. It must carefully balance its actions to keep learning about the environment (exploration) as well as performing increasingly more precise actions based on the gathered information (exploitation). The main advantage of model-based RL is the fact of being sample efficient, which makes it quite attractive in practice. However, there is a critical problem for the application of this method to real-world problems: the complexity in scaling planning to large state space Markov Decision Processes (MDP), which becomes intractable as it requires performing dynamic programming over exponentially large trees. One way to circumvent this problem is to construct approximate versions of the planning tree. In [4], the authors proved that iterating over a set of sampled states, i.e., covering only a vanishing fraction of the full look-ahead tree, is sufficient to compute near-optimal actions. The authors developed an algorithm, known as *Sparse Sampling*, that constructs a look-ahead tree iteratively by sampling every possible action of the MDP, at every state generated according to the underlying environment dynamics. An ϵ-approximation

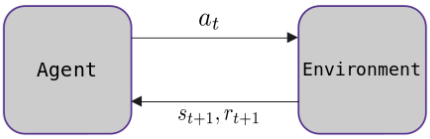

FIGURE 1. Agent-Environment Paradigm: Learning by interaction.

The associate editor coordinating the review of this manuscript and approving it for publication was Kok-Lim Alvin Yau.

(see Equation 2, below) of the optimal action to take at the initial state, can be derived from the expected reward of some policy π, also known as the *value-function*, V, compared with the optimal value function, V^*, by expanding the look-ahead tree to a given horizon, h, with bounded complexity,

$$\mathcal{O}\left(\frac{h|A|}{\epsilon}^{h\log(\frac{h}{\epsilon})}\right) \tag{1}$$

$$|V^{\pi}(s) - V^{*}(s)| \leq \epsilon, \quad \forall s \in S \tag{2}$$

where S is the state space, A is the action space, h is the horizon and ϵ is the approximation error. The algorithm exhibits a linear dependence on the number of possible actions, $|A|$, and an exponential dependence on the horizon. However, it shows no dependence on the total number of states of the environment. This constitutes the main advantage with respect to other planning algorithms since environments with large state-spaces can be addressed smoothly. Algorithms like sparse sampling which avoid a state space dependence, are suitable for exploration in a quantum framework as well.

Thus, building on the classical sparse sampling approach, this article addresses the following research question:

Is it possible to design a sample-based quantum algorithm for planning that shows no dependence on the state space of a given MDP?

This kind of classical algorithms relies on the assumption of reasonably well constructed simulated environments [5]. If the latter does not capture the real environment dynamics, then the action suggested by the agent will not correspond to the optimal action to take in the real environment. In this work, we take one step further and enforce the simulated environment to be fully quantized, a notion that first appeared in [6], [7], allowing a quantum agent to act in its environment according to the laws of quantum mechanics. Based on this interaction we prove that a quantum version of the sparse sampling algorithm produces near-optimal actions with quadratically less computational effort when compared to its classical counterpart.

The main contributions of this paper are:

- A novel quantum tree-based, sparse sampling inspired algorithm for RL within generalized stochastic environments, and exhibiting a quadratic speedup compared to its classical counterpart.
- Development of an upper bound on the size of the search space, demonstrating the algorithm independence on the environment's total number of states.
- Development of an upper bound on the sample size required to make an ϵ-optimal decision, which sustains the aforementioned quadratic speedup.

The rest of the paper is organized as follows. Section II reviews the state of the art of quantum-enhanced RL. In Section III we construct quantum oracular instances of MDP's to both deterministic and stochastic environments. In Section IV we propose a quantum version of the sparse sampling algorithm and in Section V its complexity is analyzed. Section VI empirically analyses the performance of the proposed quantum algorithm, executing a small MDP in IBM's quantum simulator. Finally, Section VII concludes and proposes topics for further research.

II. RELATED WORK

RL is based on the formal problem of MDP's, which involve evaluative feedback as well as associativity [1] i.e., selecting different actions in response to different situations. Therefore, the cornerstone of quantum algorithmic structures applied to the RL framework lies in the quantized formulation of MDP's. Some authors [8], [9] suggest a quantum MDP based on the notion of quantum superoperators, which give the dynamics of each action upon the Hilbert space describing the state of the MDP. Alternatively, Dunjko *et al.* resort to oracularized environments [6]. Given a classical environment, E, described by the MDP, a unitary quantum oracle E^q can be seen as a black-box that simulates E. The oracle allows the quantum algorithm to sample some property of E, e.g., given a state and an action (or sequence of actions) obtain some statistics related to the respective reward. E^q has to be fair, in the sense that it cannot provide more information to the quantum algorithm than what E would provide under classical access. This is guaranteed when E^q is a reversible realization of E and explains why oracles are referred to as black boxes: the quantum algorithm has no access to its inner processing. Dunjko *et al.* [6] propose an oracle that encodes the probability that a given sequence of actions will be rewarded. In this work, we improve the latter approach by directly encoding expected rewards of state-action pairs and by addressing stochastic environments, which mimic more realistic environments and generalize the former encoding. Exponential speedups in quantum RL have been demonstrated in the oracularized framework, assuming the existence of RL environments that can be directly mapped as instances of well-known quantum problems like Fourier Sampling [7], [10].

Most quantum-enhanced techniques applied to planning and RL lie in the **(CQ)** spectrum of quantum machine learning [11] i.e., classical data about an agent is encoded and further processed by a quantum device. Ref [12] propose a quantum dynamic programming algorithm with quadratic speedup with respect to the classical counterpart. In [13] the authors used a Quantum Boltzmann Machine to devise the optimal policy for a RL agent. In [14], [15] the authors proved a quantum speedup in a quantized version of the Projection Simulation algorithm [16] which was recently experimentally tested in a fully tunable integrated nanophotonic processor [17].

This work can be distinguished from the latter approaches since it is, to the best of our knowledge, the first quantum sample-based approach applied to the RL problem. Furthermore, combining a quantum-inspired sparse sampling approach with oracularized environments provides a quantum algorithm for near-optimal planning showing no dependence on the state space as opposed to previous quantum planning approaches.

III. QUANTUM AGENT-ENVIRONMENT INTERFACE

A quantum agent is an entity that has an internal representation of its current state. Let a classical MDP be given by $M = \langle S, A, P, R, \gamma \rangle$, where S is the state space, A is the action set, P is the state transition probability function, R is the reward function, and $\gamma \in [0, 1)$ is the optional discount factor. The role of an agent is to map a state, $s \in S$ to a corresponding action, $a \in A$ that it will later perform upon the environment. Therefore, the quantum agent is interpreted as a reversible function, $f : |s\rangle \otimes |0\rangle^{\otimes n_a} \mapsto |s\rangle \otimes |a\rangle$ that corresponds to any quantum circuit that prepares the mapping depicted in Figure 2, where $n_s = \mathcal{O}(log_2|S|)$ and $n_a = \mathcal{O}(log_2|A|)$ are the number of qubits required to basis encode the agents' state and action, respectively. For the sake of legibility, the superscripts $\otimes n_s$ and $\otimes n_a$ will be omitted throughout the text.

FIGURE 2. Quantum agent state preparation.

The power of a quantum agent comes from the fact that each action it takes can be a uniform superposition over the action set. The action is superposed by f conditioned on the state itself allowing the superposition to be over the set of admissible actions, A_s, at a given state, that in a general setting, is different from state to state. Therefore the action register will be the following superposition state

$$|a_s\rangle = \frac{1}{\sqrt{|A_s|}} \sum_{a \in A_s} |a\rangle \tag{3}$$

When dealing with environments that have the same number of actions for each one of its states, the uniform superposition over the action set can be efficiently implemented by f. It is simply the tensor product of n_a Hadamard gates applied to the action register. The goal of the agent is to find the optimal mapping, f^*, that maximizes the *reward*. This means that the agent also requires an internal representation of the *reward*, which will depend on the nature of the quantum environment itself. In this work we consider the environment to be generally stochastic, however, we will clarify that deterministic state transitions can still occur given the oracular formulation proposed.

An environment is formally described in RL by an MDP. A notion of a quantum MDP already exists [8]. However, we will follow the approach taken in [6], i.e., we think of environments as black-boxes with which the quantum agent interacts. When constructing *oraculizations* of classical task environments we need to guarantee that the environment does not provide more information concerning the classical counterpart, which is essentially guaranteed when there is a reversible version of the classical environment [6]. Furthermore, the oracle will be composed of two main stages, one responsible for the state transitions under the respective dynamics of a given classical environment, T, and another responsible for assigning to the agent a new reward for the accomplished transition, R.

The stochastic state transition operator can be realized as the following mapping:

$$T : |s\rangle \otimes |a\rangle \otimes |0\rangle \mapsto |s\rangle \otimes |a\rangle \otimes \sum_{s' \in S} \sqrt{P^a_{ss'}} |s'\rangle \tag{4}$$

that prepares the linear combination of possible states, given a state-action pair weighted by the corresponding state transition probability $P^a_{ss'}$, via *amplitude encoding* [18], s.t, $\sum_{s' \in S} P^a_{ss'} = 1$. The deterministic case is a special case where transition probability equals 1 for a single state s'. Therefore, a deterministic transition step derives from Equation 4 as

$$T : |s\rangle \otimes |a\rangle \otimes |0\rangle \mapsto |s\rangle \otimes |a\rangle \otimes |s'\rangle \tag{5}$$

In the context of this work, rewards depend only on the agent state and are represented by a mapping from state-space to a real number, $R_s : S \mapsto \mathbb{R}$. However, other generalizations, such as transition dependent rewards, $R_{sa} : S \times A \times S \mapsto \mathbb{R}$, can be trivially included in the proposed approach.

The reward function operator will use *angle encoding* [19] to rotate a single qubit, known as the reward qubit, accordingly to the reward $\mathcal{R}_s$ associated to the current state of the agent:

$$R_s : |s\rangle \otimes |r\rangle \mapsto |s\rangle \otimes e^{j\mathcal{R}_s \hat{\sigma}_y} |r\rangle \tag{6}$$

The angle encoding mechanism essentially provides addition "for free", therefore, ensuring that the agent can represent the cumulative reward through a sequence of actions.

$$\begin{aligned} |\psi\rangle &= |s\rangle \otimes |a\rangle \\ R_s(R_2)R_s(R_1)|\psi\rangle &= e^{jR_2\hat{\sigma}_y}.e^{jR_1\hat{\sigma}_y}|\psi\rangle = e^{j(R_1+R_2)\hat{\sigma}_y}|\psi\rangle \end{aligned} \tag{7}$$

However, such free addition comes with a trade-off, i.e., the total reward the agent can receive is restricted to the interval $[0, \pi/2]$. Rotation on a qubit is a periodic function, therefore, outside the interval, we lose information about the true reward and the agent will not be able to distinguish one reward sequence from another. Furthermore, rotations are treated as y-rotations, thus, assuming that the reward qubit is initialized in the ground state, after a transition step, it will become

$$\begin{aligned} e^{j\mathcal{R}_s\hat{\sigma}_y}|r\rangle &= \begin{bmatrix} cos(\mathcal{R}_s) & -sin(\mathcal{R}_s) \\ sin(\mathcal{R}_s) & cos(\mathcal{R}_s) \end{bmatrix} |0\rangle \\ &= \begin{bmatrix} cos(\mathcal{R}_s) & -sin(\mathcal{R}_s) \\ sin(\mathcal{R}_s) & cos(\mathcal{R}_s) \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} cos(R_s) \\ sin(R_s) \end{bmatrix} \\ &= cos(\mathcal{R}_s)|0\rangle + sin(\mathcal{R}_s)|1\rangle \end{aligned} \tag{8}$$

The use of the discount factor $\gamma \in [0, 1)$ helps to normalize the reward achieved.

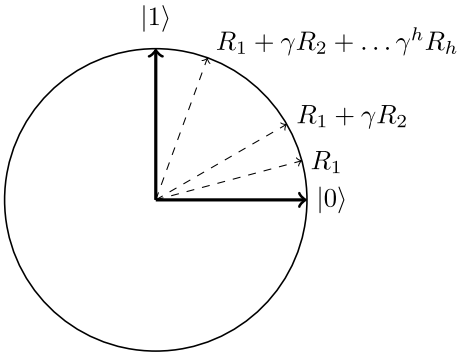

FIGURE 3. Evolution of the reward qubit.

Assuming that the environment has access to the maximum reward possible R_{max}, in a single transition step, we know that with $\gamma \in [0, 1)$:

$$\sum_{t=0}^{h-1} \frac{\gamma^t R_t}{R_{max}} \leq \sum_{t=0}^{h-1} \frac{\gamma^t R_{max}}{R_{max}} \leq \sum_{t=0}^{h-1} \gamma^t \leq \frac{\gamma^h - 1}{\gamma - 1} \tag{9}$$

Thus, normalizing the reward to

$$\frac{\pi}{2} \frac{(\gamma - 1)}{(\gamma^h - 1)} \frac{\gamma^t R_t}{R_{max}} \tag{10}$$

ensures that for horizon h, the maximum reward collected is at most $\frac{\pi}{2}$ as in Equation (11)

$$\sum_{t=0}^{h-1} \frac{\pi}{2} \frac{(\gamma - 1)}{(\gamma^h - 1)} \frac{\gamma^t R_t}{R_{max}} \leq \sum_{t=0}^{h-1} \frac{(\gamma - 1)}{(\gamma^h - 1)} \gamma^t \leq \frac{\pi}{2} \tag{11}$$

At this point, we already have all the ingredients for composing the quantum oracular environment O. Given that the oracle will interact with different quantum registers at every time step, that is, will act according to transition step quantum registers, the oracular environment will be composed of the product of the two stages mentioned above:

$$O = \prod_{t=0}^{h-1} R_t T_t \tag{12}$$

IV. QUANTUM TREE-BASED PLANNING

The goal of a RL agent is to learn how to exploit the environment in order to maximize the *expected* reward. Expectation comes from the stochasticity the environment, traced back to the recursive relationship of the *Bellman expectation equation* [1] following a policy π:

$$\begin{aligned} & q_\pi(s, a) \\ & = \mathbb{E}_\pi \left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | S_t = s, A_t = a \right] \\ & = \mathbb{E}_\pi \left[R_{t+1} + \gamma q(S_{t+1}, A_{t+1}) | S_t = s, A_t = a \right] \\ & = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a'|s') q_\pi(s', a') \end{aligned} \tag{13}$$

in which the expectation is revealed by the subsequent steps rewards weighted by both state transition probabilities and the policy π.

Using the angle encoding formalism to represent rewards allows the expected reward associated with a given sequence of h actions to be encoded as an amplitude. Actually, let the initial quantum state, including the uniform superposition over the action set A, be given by the following tensor product:

$$|\psi_0\rangle = |s\rangle \otimes \frac{1}{\sqrt{|A|}} \sum_{a \in A} |a\rangle \otimes |0\rangle \otimes |r\rangle \tag{14}$$

Then, a single interaction between the agent and the quantum environment yields

$$\begin{aligned} T|\psi_0\rangle &= |s\rangle \otimes \frac{1}{\sqrt{|A|}} \sum_{a \in A} |a\rangle \otimes \sum_{s' \in S} \sqrt{P_{ss'}^a} |s'\rangle \otimes |r\rangle = |\psi_0'\rangle \\ R|\psi_0'\rangle &= |s\rangle \otimes \frac{1}{\sqrt{|A|}} \sum_{a \in A} |a\rangle \otimes \sum_{s' \in S} \sqrt{P_{ss'}^a} |s'\rangle \otimes e^{j\mathcal{R}_{s'} \hat{\sigma}_y} |r\rangle \\ &= \sum_{a \in A} \sum_{s' \in S} \frac{1}{\sqrt{|A|}} \sqrt{P_{ss'}^a} e^{j\mathcal{R}_{s'} \hat{\sigma}_y} |s\rangle \otimes |a\rangle \otimes |s'\rangle \otimes |r\rangle \end{aligned} \tag{15}$$

which prepares a linear combination between all possible transition states, weighted by the product of the state transition probabilities and the respective outcome state rewards. This reasoning can be extended to allow for h interactions, i.e., sequences of h actions, by resorting to the quantum oracular environment O, as given by Equation (12). This is equivalent to compute a lookahead tree of depth h in superposition. Figure 4 represents the one-step lookahead tree.

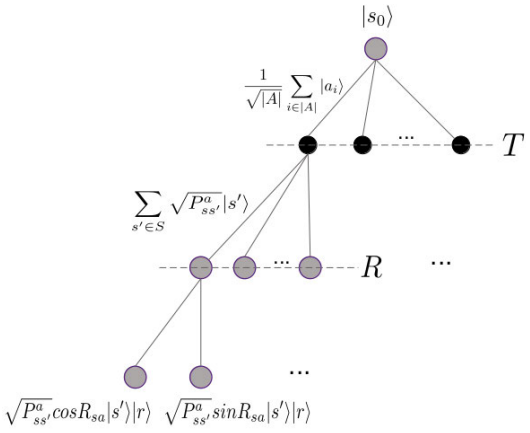

FIGURE 4. One step lookahead tree computed in superposition, created by oracle calls of T and R.

Let $|\psi\rangle = O|\psi_0\rangle$ be the quantum superposition resulting from the evaluation of sequences of h actions. The amplitude of each term of $|\psi\rangle$ with $|r\rangle = |1\rangle$ represents an approximation [1] to the expected reward received by the agent for the corresponding sequence of h actions. In fact, combining

[1]It corresponds to an approximation since rewards are encoded as a y-rotation on a qubit, therefore the expectation will be measured with respect to a trigonometric parametrization of the reward rather than the actual true value. However, without loss of generality, we can still distinguish the expectation of different actions.

Equation (8) and Equation (15), and thus expanding the representation of $e^{j\mathcal{R}_{s'}\hat{\sigma}_y}|r\rangle$, yields

$$R|\psi_0'\rangle = \sum_{a\in A}\sum_{s'\in S}\frac{1}{\sqrt{|A|}}\sqrt{P^a_{ss'}}|s\rangle\otimes|a\rangle\otimes|s'\rangle\otimes \otimes[cos(\mathcal{R}_{s'})|0\rangle + sin(\mathcal{R}_{s'})|1\rangle] \quad (16)$$

Equation (16) shows that the expected reward associated with each sequence of h actions is encoded in the sine term of the reward qubit, i.e., in the superposition terms with $|r\rangle = 1$.

Note that for an MDP with initial state $s_0 \in S$ and action space A, there is a corresponding quantum state as in Equation (14) representing the agent, which interacts with the quantum environment for a given *horizon, h*. $O|\psi_0\rangle$ acts in the corresponding transition step sub-registers, preparing a superposition state $|\psi\rangle$ in which the term with $|r\rangle = |1\rangle$ and with the highest amplitude represents the maximum expected reward. That term is associated with the optimal action to take at s_0.

Therefore the goal is to measure the quantum register holding the identifier of the first action in the sequence of actions. Ideally, the quantum state $|\psi\rangle$ would collapse into the term with larger amplitude and $|r\rangle = |1\rangle$, thus giving access to the optimal action to take at s_0. However, this is far from guaranteed. If some sequences of actions in the superposition lead, with high probability, to small or even null rewarded states, the cosine term of the reward qubit can be significantly larger than the associated sine term. In other words, it is possible that some terms with $|r\rangle = |0\rangle$ in $|\psi\rangle$ have large amplitudes and thus a significant probability of being measured. These terms do not carry any relevant information concerning the optimal action and thus, these measurements should be discarded.

If the environment was deterministic, i.e., each action possessed a single possible outcome, there wouldn't be any expectation and the rewards could be represented using basis encoding. The Quantum Maximum Finding (QMF) algorithm [20] could thus be applied and the optimal action would be measured with high probability [6]. Stochastic environments, however, explicitly require expectation, i.e., weighting rewards by their corresponding state transition probabilities. Resorting to angle encoding to represent the expected reward precludes the use of QMF.

Let p be the probability of measuring a "good" term of the $|\psi\rangle$ superposition, i.e., a term with $|r\rangle = |1\rangle$. p can be maximized by applying the amplitude amplification algorithm [21], a generalization of Grover's algorithm [22] to arbitrary non-uniform superpositions. $|\psi\rangle$ can be decomposed into the two orthogonal states, $|\psi_{good}\rangle$ and $|\psi_{bad}\rangle$, each representing $|\psi\rangle$'s projection onto the subspace with reward qubit $|r\rangle = |1\rangle$ and $|r\rangle = |0\rangle$, respectively. Let the angle θ between $|\psi\rangle$ and $|\psi_{bad}\rangle$ be related to the aggregated amplitude of good states in $|\psi\rangle$, such that $\theta = \arcsin(\sqrt{p})$. Then, $|\psi\rangle$ is given by

$$|\psi\rangle = sin(\theta)|\psi_{good}\rangle + cos(\theta)|\psi_{bad}\rangle \quad (17)$$

The amplitude amplification algorithm is based on successive applications of *Grover's operator*, $\mathcal{G}$, to the superposition $|\psi\rangle$. Each application of $\mathcal{G}$ increases the amplitude of states in $|\psi_{good}\rangle$. After j applications of $\mathcal{G}$ the quantum state will be

$$|\psi^{(j)}\rangle = sin((2j+1)\theta)|\psi_{good}\rangle + cos((2j+1)\theta)|\psi_{bad}\rangle \quad (18)$$

with the probability of measuring a good state changing from $p = \sin^2(\theta)$ to $p^{(j)} = sin^2((2j+1)\theta)$. If p is known *a priori*, or equivalently if θ is known, then j can be set such that $p^{(j)} \approx 1$; This requires $j \propto \mathcal{O}(\frac{1}{\sqrt{p}})$ applications of $\mathcal{G}$, resulting on the expected quadratic advantage over a classical algorithm, which would require $\mathcal{O}(\frac{1}{p})$ computational steps.

Within the context of this work $|\psi\rangle$ is some arbitrary non-uniform superposition and p is thus unknown. The optimal number of applications of $\mathcal{G}$ cannot be precomputed. Thus, the exponential adaptive search version of the amplitude amplification algorithm, as described in [23], has to be used. This entails the need for multiple executions of the fundamental amplitude amplification circuit, each with a randomly selected number j of applications of $\mathcal{G}$. Such j's are randomly selected from a range of possible values that grows exponentially as the algorithm iterates. This algorithm, referred to as *QSearch*, has been shown to exhibit the same computational complexity as the original algorithm, i.e., $\mathcal{O}(\frac{1}{\sqrt{p}})$, although with larger constants.

On average half of the states in $|\psi\rangle$ are "good" states, with the other half being "bad" states, since each rotation of the $|r\rangle$ qubit results on both a sine and a cosine term (the only exceptions occurring when the reward is either 0 or R_{max}). If $|\psi\rangle$ was a uniform superposition, the probability of measuring a good state would be $p = \frac{1}{2}$, and applying amplitude amplification would not make sense. However, $|\psi\rangle$ is an arbitrary superposition with unknown amplitudes associated with "good" states; in the general case, it is expected that $p \ll \frac{1}{2}$, and amplitude amplification is crucial to make the probability of measuring a good state close to 1.

The procedure described above results in the measurement of an action, which corresponds to the first action in a sequence of h actions. The measurement of the quantum register corresponds to sampling that action from a distribution proportional to each actions' sequence squared expected reward. To find the action with maximum expected reward multiple sampling is performed: the entire procedure is repeated (sampled) m times. A distribution $\mathcal{A}$ over the set A of possible actions is built, by counting how many times each action is measured. The optimal action is then selected as

$$a^* = \underset{a}{\text{argmax}}\,\mathcal{A} \quad (19)$$

If we set the horizon to be the effective horizon $\mathcal{O}(\frac{1}{1-\gamma})$ i.e., the look-ahead at which the γ-discounted reward is approximately null, then in the limit, m samples will generate the true distribution for $\mathcal{A}$. Setting m appropriately we can reach an ϵ-approximation of the optimal action to take at the initial state.

The full quantum tree-based planning is presented in Algorithm 1.

Algorithm 1 Quantum Tree-Based Planning

horizon h, R_{max}, *samples* m;
$s \leftarrow 0$, $\mathcal{A} \leftarrow [0, \ldots, 0]$;
while $s < m$ **do**
  $i \leftarrow 0$;
  $|s_i\rangle \leftarrow |0\rangle^{\otimes log_2|S|}$, $|r\rangle \leftarrow |0\rangle$;
  $|\psi_i\rangle \leftarrow |s_i\rangle \otimes |r\rangle$;
  while $i < h$ **do**
    $|a_i\rangle \leftarrow |0\rangle^{\otimes log_2|A|}$, $|s_{i+1}\rangle \leftarrow |0\rangle^{\otimes log_2|S|}$;
    $|a_i\rangle \leftarrow H^{\otimes log_2|A|}|a_i\rangle$;
    $|\psi_i'\rangle \leftarrow |\psi_i\rangle \otimes |a_i\rangle \otimes |s_{i+1}\rangle$;
    $|\psi_i''\rangle \leftarrow T\,(|s_i\rangle \otimes |a_i\rangle \otimes |s_{i+1}\rangle)$;
    $|\psi_{i+1}\rangle \leftarrow R_s\,(|s_{i+1}\rangle \otimes |r\rangle)$;
    $i \leftarrow i + 1$;
  end
  $action \leftarrow QSearch(|\psi_{h-1}\rangle)$;
  $\mathcal{A}[action] \leftarrow \mathcal{A}[action] + 1$;
  $s \leftarrow s + 1$;
end
Return $\underset{a}{\text{argmax}}\,\mathcal{A}$;

V. COMPLEXITY ANALYSIS

The computational complexity of the proposed algorithm is determined by the complexity of the exponential search algorithm and by the number of samples taken, assuming that the oracles are efficiently built. Let N be the size of the search space and n the number of marked states. The QSearch algorithm will find a good state in time complexity $\mathcal{O}(\sqrt{\frac{N}{n}})$ [23]. The original Grover's algorithm assumes a uniform initial distribution. However, as outlined in [24], the amplitude amplification subroutine applies to arbitrary superpositions, whose complexity depends on the average and variance of the initial amplitude distribution of the marked and unmarked states. It still requires $\mathcal{O}(\sqrt{\frac{N}{n}})$ iterations, although the success probability can be small for certain unfavourable initial amplitude distributions. We use QSearch as a way to amplify the probability of measuring a high expected reward action, given the unknown initial balance between marked/unmarked states in the case of model-free RL. Moreover, as mentioned in Section IV to have a certain degree of confidence in the suggested action, this procedure is repeated m times. Therefore the complexity of the algorithm becomes

$$\mathcal{O}(m\sqrt{\frac{N}{n}}) \tag{20}$$

The search space and the number of marked states will be both dictated by the dynamics of the MDP, which is completely unknown to the agent. However, without loss of generality, we can say that in the worst case the agent will mark a single state, thus maximising $\sqrt{\frac{N}{n}}$. Characterizing both the size of the search space and the number of required samples, m, calls for a more fine-grained analysis which we will do in Subsection V-A and Subsection V-B respectively.

A. BOUNDING THE SEARCH SPACE

In a purely deterministic setting, we know that the branching factor associated with each transition step will be the number of actions that an agent can take, and so, for a given tree depth, we have $\mathcal{O}(|A|^h)$ superposition terms. In a general stochastic environment, this is not true, given that in the quantum setting, the branching factor will be a function of the number of actions and the respective number of reachable states, as can be seen in Figure 4. Let K be a random variable that captures the branching factor of the environment. We can say that for a given tree depth, we have bounded search space $\mathcal{O}(K)$. The state transition probability matrix will vary according to the problem, therefore we give a probabilistic bound for K that fits generalized environments.

Let k_i be a random variable that quantifies the outcome of some action i.e., the number of reachable states given a state-action pair. In the simplest case, a deterministic transition step leads to a single fixed state. On the other side of the spectrum, a transition step could in principle lead to every possible state of the environment. However, very few environments have this feature. An example of this arises in the well known *bandits problem* [1]. In this case, a transition step leads always to every possible state of the environment. However, interestingly, bandits correspond to a single state environment. Moreover, in a typical, or realistic stochastic environment, a transition step will lead to a small subset of all possible states. This effect can be modelled by a Beta distribution (see Figure 5) that for a state-action pair generates an increasingly larger subset of states with exponentially decaying probability. Let k_i be sampled from a Beta distribution Beta(α, β) with $\alpha < \beta$,

$$k_i \sim Beta(\alpha, \beta) \tag{21}$$

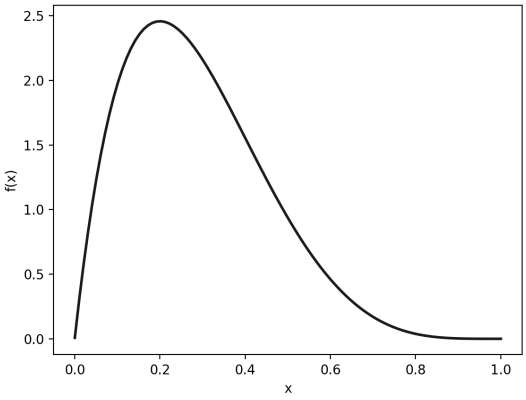

FIGURE 5. Beta distribution with $\alpha = 2$ and $\beta = 5$.

The Beta distribution of Figure 5 has expected value

$$\mu = \mathbb{E}(x) = 0.2857 \tag{22}$$

In general, $\forall_{\alpha,\beta} \in \mathbb{R}$, the expected value of $Beta(\alpha, \beta)$ is

$$\mu = \frac{1}{1 + \frac{\beta}{\alpha}} \tag{23}$$

In an MDP with $|A|$ possible actions, the one-step lookahead tree (see Figure 6) generates at most $|A|$ random variables, k_i, $i \in 1 \ldots |A|$, each one generating a variable number of next states accordingly to the Beta distribution.

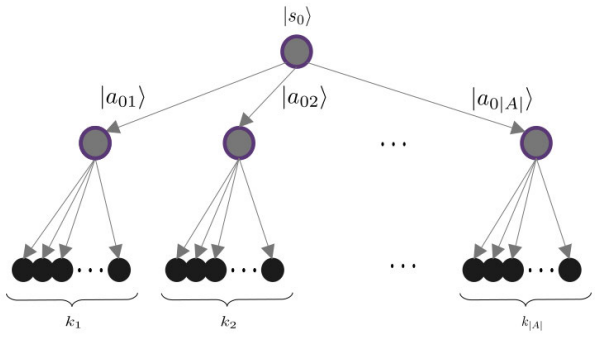

FIGURE 6. One-step lookahead tree. Nodes in black correspond to states generated from each random variable k_i.

Let K be the sum of all random variables corresponding to the total number of generated states after a single transition step, i.e.,

$$K = k_1 + \cdots + k_{|A|} = \sum_{i=1}^{|A|} k_i \tag{24}$$

Let μ be the expected value of K

$$\mu = \mathbb{E}[K] = \sum_{i=1}^{|A|} \mathbb{E}[k_i] = |A| * \mathbb{E}[k_i] = |A| \frac{1}{1 + \frac{\beta}{\alpha}} \tag{25}$$

From the Chernoff bound, the probability of K deviating from the expected value decays exponentially as

$$\mathbb{P}[K \geq (1+\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{2+\delta}}, \quad \delta \geq 0 \tag{26}$$

Setting $\delta = 1 + \frac{2\beta}{\alpha} \geq 0$, the probability of K being larger than the action space itself decreases exponentially with the dimension of the action space.

$$\mathbb{P}[K \geq 2|A|] \leq e^{-\frac{(1+\frac{2\beta}{\alpha})^2 |A| (1+\frac{2\beta}{\alpha})}{2+(1+\frac{2\beta}{\alpha})}} \tag{27}$$

From Equation (27) we conclude that the one-step look-ahead tree will have more than $2|A|$ generated states with a small probability. This decays exponentially with the size of the action space. Therefore, for a full look-ahead tree of horizon h, the number of states is bounded by

$$\mathcal{O}\left(2|A|^h\right) \tag{28}$$

The bound in Equation (28) asserts that the size of the search space in typical environments will be larger than twice the size of the action space with exponentially decreasing probability. Specifically, if $\beta \geq 2\alpha$, we can say that the Beta distribution constitutes a fair model for the number of states generated given a state-action pair. If the presence of a binary action MDP, the probability of the search space being larger than twice the size of the action space will be less than 10%. However, the probability decreases exponentially with the cardinality of the action space. For $|A| = 4$, the probability decreases to less than 1% which fits in the narrative of large state-action space MDP's.

B. BOUNDING THE SAMPLE SIZE

The important question now is to define the optimal number of samples required for an ϵ-approximation of the action that maximizes the expected reward at the initial state. The *Wilson interval* [25] gives us an estimate of the number of samples needed from an arbitrary qubit to get an ϵ-approximation to the probability of measuring that qubit in any of the basis states $\{|0\rangle, |1\rangle\}$. From [18], we know that the number of samples necessary for a single qubit is

$$m \leq \mathcal{O}\left(\frac{z^2}{8\epsilon^2}(\sqrt{16\epsilon^2 + 1} + 1)\right) \tag{29}$$

where z is the estimate confidence level.[2] We want to find the approximate probability of measuring an action that leads to the highest expected reward, which typically is encoded in more than a single qubit. A single qubit may refer to a simple binary action MDP. In general, this entails the need for measuring $log|A|$ qubits. So we can use the single qubit case to suggest an approximation for the multiple qubit case. The number of samples needed is bounded by

$$m \leq \mathcal{O}\left(\frac{z^2 log|A|}{8\epsilon^2}(\sqrt{16\epsilon^2 + 1} + 1)\right) \tag{30}$$

C. DISCUSSION

From Equation (20) it is known that the complexity of the proposed algorithm is $\mathcal{O}(m\sqrt{\frac{N}{n}})$.

Combining the previous expression with the bound in the search space size computed in Subsection V-A, equation 28, we get

$$\mathcal{O}(m\sqrt{\frac{N}{n}}) \sim \mathcal{O}\left(m\sqrt{\frac{2|A|^h}{n}}\right)$$

thus establishing the independence on the size of the MDP state space.

By further considering applying the bound on m developed in Subsection V-B, Equation (30), the overall complexity of the algorithm is bounded by

$$\mathcal{O}(m\sqrt{\frac{N}{n}}) \leq \mathcal{O}\left(\frac{z^2 log|A|}{8\epsilon^2}(\sqrt{16\epsilon^2 + 1} + 1)\sqrt{\frac{2|A|^h}{n}}\right) \tag{31}$$

[2]Tabulated value [18]. A z-value of 2.58 corresponds to 99% confidence.

with probability

$$e^{-\frac{(1+\frac{2\beta}{\alpha})^2|A|\mu}{2+\mu}} \tag{32}$$

This demonstrates that the quantum algorithm proposed suggests an ϵ-optimal action to be taken in any initial state of a given MDP with quadratically less computational effort with respect to the original classical Sparse Sampling algorithm. The complexity of the latter classical algorithm is $\mathcal{O}\left(\frac{h|A|}{\epsilon}^{h\log(\frac{h}{\epsilon})}\right)$, as given by equation 1 and repeated here for convenience.

VI. NUMERICAL EXPERIMENTS AND RESULTS

The convergence rates of the quantum algorithm proposed here and its classical sparse sampling counterpart (as described in Figure 1, page 198 of [4]) were empirically compared, based on the respective query complexities. The figure of merit used in the comparison is the respective frequency of selection of the best action as a function of the number of queries. In the quantum setting, one query corresponds to one oracle call, as given by equation 20. In the classical setting, one query corresponds to the evaluation of the Q-function for a sequence of h actions, where h is the horizon. In practice, the total number of queries performed by the classical algorithm is equal to how many times the condition presented in line 1 of the method **EstimateQ** () evaluates to True (see Figure 1, page 198 of [4]). The frequency of selection of the best action, denoted by a^* in the figures below, is the ratio between how many times the best action was selected and the total number of experiments performed. It is therefore a value in the interval [0, 1]. To identify the best action, i.e. the action with the larger expected reward for the given initial state, a brute force algorithm was executed, which evaluates all possible sequences of actions and respective outcomes; this is only possible because the evaluated MDPs (see below) are of moderate size concerning the number of states and actions. An algorithm is said to exhibit a better convergence rate if it requires fewer queries for the same frequency or, conversely, if a larger frequency is achieved for the same number of queries.

The procedure used to obtain the experimental data for a given number of queries is described by Algorithm 2. Note that $\mathcal{A}$ is the distribution over the measured actions for *#queries* of the MDP (note that *execute(#queries)* applies both to the quantum and the classical cases). The action is then selected from $\mathcal{A}$; if this corresponds to the best action, then the respective histogram bin is updated. The (normalized) histograms are presented in the next subsections, allowing for a comparison of the quantum and classical algorithms' query complexities.

Experimental results are presented for three MDPs, selected to represent problems with different degrees of difficulty in terms of finding the action with the highest expected reward. Subsection VI-A presents a 2 × 2 stochastic grid world, whose optimal action exhibits a significantly larger expected reward than alternative actions, therefore being easy to find. Subsection VI-B presents a randomly generated sparse MDP: from any state, only a small subset of the full state space can be reached. Subsection VI-C presents a randomly generated dense MDP: from any state-action pair every other state of the environment can be reached. Both the sparse and dense randomly generated MDP's have 4 states and two alternative actions.

Algorithm 2 Best Action Frequency for *#queries*

```
input: The number of queries: #queries
for #queries do
    #BestA = 0;
    for #experiments in [1, ..., NExperiments] do
        A = execute (#queries) ;
        a = selectAction (A);
        if a == BestAction then
            #BestA++;
        end
    end
    Hist[#queries] = #BestA/NExperiments;
end
```

All experimental results were obtained using the IBM Qiskit simulator [26]. All experiments were executed taking $s_0 = 0$ as the initial state; A discount factor $\gamma = 0.9$ was used and the rewards were normalized as described in Equation (11). For the sake of clarity and simplicity, the evolution of the quantum state for a single interaction of the agent with the oracularized environment is presented only for the grid world, i.e. the simplest MDP.

A. STOCHASTIC GRIDWORLD

The simulated environment corresponds to a 2 × 2 stochastic gridworld MDP, $A = \{up, down, left, right\} \mapsto \{0, 1, 2, 3\}$, illustrated in Figure 7. The state space is $S = \{s_0, s_1, s_2, s_3\} \mapsto \{0, 1, 2, 3\}$ and the per-state reward function is $R(s) \in \mathbb{R}, \quad \forall s \in S$ as given by Equation (33). The environment stochasticity comes from the fact that every action has a probability equal to 0.3 of moving to an adjacent state, different from the state implied by that action, which is

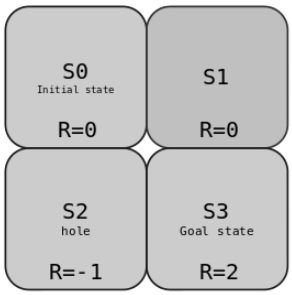

FIGURE 7. The 2 × 2 stochastic gridworld.

reached with a probability equal to 0.7.

$$R(s) = \begin{cases} 0 & s = s_0 \\ 0 & s = s_1 \\ -1 & s = s_2 \\ 2 & s = s_3 \end{cases} \tag{33}$$

The initial state, after preparing an uniform superposition over the four possible actions, is as given by Equation (34).

$$|\psi_0\rangle = |s_0\rangle \otimes \frac{1}{2}\left[|up\rangle + |down\rangle + |left\rangle + |right\rangle\right] \otimes|0\rangle \otimes |r\rangle \tag{34}$$

Applying the stochastic state transition operator entails:

$$\begin{aligned} T|\psi_0\rangle = |s0\rangle \\ \otimes[\frac{1}{2}|up\rangle \otimes (\sqrt{0.7}|s_0\rangle + \sqrt{0.15}|s_1\rangle + \sqrt{0.15}|s_2\rangle) \\ +\frac{1}{2}|down\rangle \otimes (\sqrt{0.7}|s_2\rangle + \sqrt{0.15}|s_0\rangle + \sqrt{0.15}|s_1\rangle) \\ +\frac{1}{2}|left\rangle \otimes (\sqrt{0.7}|s_0\rangle + \sqrt{0.15}|s_1\rangle + \sqrt{0.15}|s_2\rangle) \\ +\frac{1}{2}|right\rangle \otimes (\sqrt{0.7}|s_1\rangle + \sqrt{0.15}|s_0\rangle + \sqrt{0.15}|s_2\rangle)] \\ \otimes|r\rangle = |\psi_0'\rangle \end{aligned} \tag{35}$$

The action of the reward operator is given by Equation (36), where η is the reward normalization factor (see Equation (11)).

$$\begin{aligned} R|\psi_0'\rangle = |s0\rangle \\ \otimes[\frac{1}{2}|up\rangle \otimes (\sqrt{0.7}|s_0\rangle + \sqrt{0.15}|s_1\rangle + \sqrt{0.15}e^{-j\eta}|s_2\rangle) \\ +\frac{1}{2}|down\rangle \otimes (\sqrt{0.7}e^{-j\eta}|s_2\rangle + \sqrt{0.15}|s_0\rangle + \sqrt{0.15}|s_1\rangle) \\ +\frac{1}{2}|left\rangle \otimes (\sqrt{0.7}|s_0\rangle + \sqrt{0.15}|s_1\rangle + \sqrt{0.15}e^{-j\eta}|s_2\rangle) \\ +\frac{1}{2}|right\rangle \otimes (\sqrt{0.7}|s_1\rangle + \sqrt{0.15}|s_0\rangle + \sqrt{0.15}e^{-j\eta}|s_2\rangle)] \\ \otimes|r\rangle = |\psi_1\rangle \end{aligned} \tag{36}$$

The resulting quantum state can evolve further to deeper levels of the decision tree, appending the respective time step quantum registers, up to the pre-established look-ahead horizon. The amplitude of valid states is then amplified, increasing the probability of measuring the optimal action to take at the initial state, i.e., the action with maximum expected reward. The probability of finding this action is increased by sampling multiple times, i.e., executing and measuring the quantum state repeatedly, as discussed in Section V.

Figure 8 depicts the best action selection frequency as a function of the number of queries for the grid world – note that the horizontal axis, the number of queries, is on a logarithmic scale. The quantum algorithm requires much fewer queries than the classical one and quickly reaches a threshold on the number of queries upon which the best action is always selected. The green dashed line presents the frequency obtained with the classical algorithm as if obtained

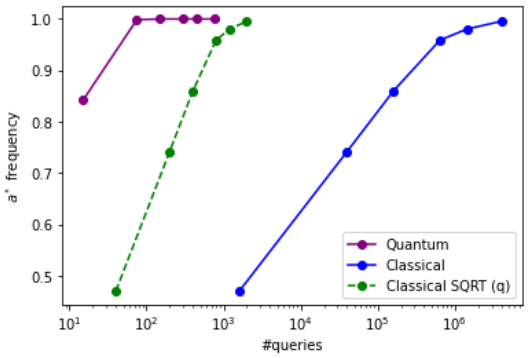

FIGURE 8. Gridworld: best action selection frequency as a function of the number of queries (the horizontal axis is in a log scale).

with the square root of the number of queries that were required. The similarity between the green dashed line and the quantum curve is a strong hint that the quantum algorithm presents a quantum advantage proportional to the square root of the classical algorithm in terms of the number of queries.

To further demonstrate that the frequency of selection of the best action for the quantum algorithm increases with the square root of the number of queries, rather than linearly, the quantum experimental data were fitted into both a linear ($a^* = c_0 * \#q + c_1$) and a square root ($a^* = c_0 * \sqrt{\#q} + c_1$) model. Figure 9 depicts the quantum experimental data as a purple solid line, the fitted linear curve as a dashed yellow line and the fitted SQRT curve as a dashed green line – note that the horizontal axis follows a linear scale. The SQRT model is a better fit, as further quantitatively demonstrated by the Root Mean Squared Error (RMSE) computed for both models. Note that the fitting suffers from a perturbation introduced by the fact that the frequency is capped

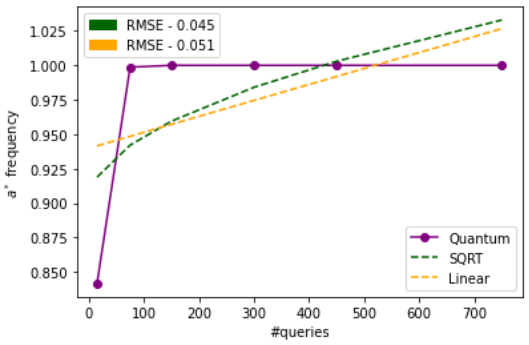

FIGURE 9. Gridworld: quantum data curve fitting with $a^* = c_0 * \#q + c_1$ and $a^* = c_0 * \sqrt{\#q} + c_1$ models.

This clearly demonstrates that the plot of Figure 9 clearly shows that the quantum algorithm selects the best action more frequently than the classical one for the same number of queries, up to extreme query rates where both algorithms select the best action with 100% frequency. This stems from the fact that the MDP in the study has an optimal action that is easily distinguished from all other actions. In the

quantum setting, the choice of normalized reward encoding of Equation (11) jointly with amplitude amplification enabled, in this case, a clearer distinction in the expected reward of all actions. From Figure 9 we can also infer from an analytical point of view given by the dashed green curve, that a model with the quadratic number of samples of the classical algorithm, behaves similarly to the quantum algorithm. Additionally, resorting to curve fitting, in Figure 9, it's demonstrated that a quadratic model fits the quantum data with a smaller mean squared error compared to a linear model. This clearly shows that the quantum data entails a quadratic reduction in the number of queries, with constant terms, compared to the classical case.

B. RANDOM SPARSE MDP

The simulated environment corresponds to a randomized sparse MDP, with 2 possible actions $A = \{0, 1\}$, illustrated in Figure 10. The state space is $S = \{0, 1, 2, 3\}$. The sparsity of the model comes from the fact that from each state-action pair, the agent can only reach a subset of the state space. This makes the model somewhat close to the grid world above, however, additionally, the expected reward is similar for both actions, making it more difficult to understand compared to the previous case.

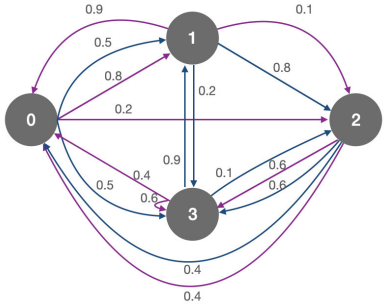

FIGURE 10. Randomized Sparse MDP with 2 possible actions. Action 0 indicated with purple color and action 1 illustrated with blue color.

From Figure 11, it can be recognized that the optimal action frequency decreased compared to the grid world model, which in fact, shows an increase in the level of difficulty when solving for the optimal action. Moreover, the quantum algorithm differs from the classical one, reaching the optimal action frequency of 1. However, asymptotically, the classical algorithm eventually reaches the same frequency, but with a larger number of queries. Additionally, it can be seen that the quadratic analytical curve seems more close to the quantum curve. The latter observation is aligned with the hypothesis of a quadratic reduction in the number of queries.

Once curve fitting is applied as in Figure 12, what can be observed is a reduction for the mean squared error for a quadratic model compared to the grid world case. The quadratic model is closer to the quantum data. The latter observation is also aligned with the expected behaviour of the quantum algorithm i.e., given the environment increasing

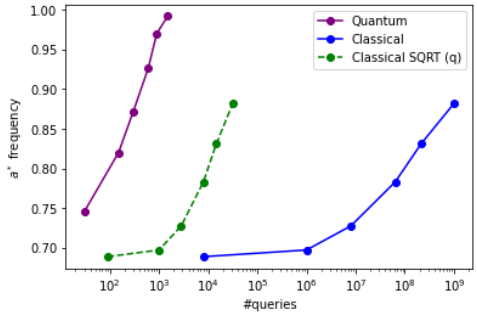

FIGURE 11. Sparse MDP!: best action selection frequency as a function of the number of queries – the horizontal axis is in a log scale.

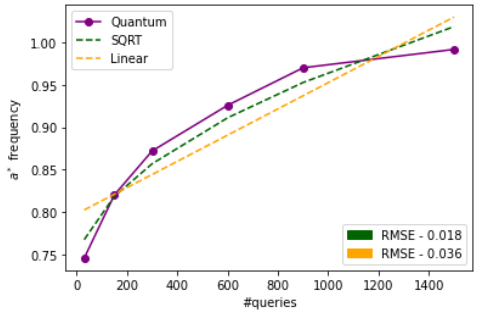

FIGURE 12. Sparse MDP!: quantum data curve fitting with $a^* = c_0 * \#q + c_1$ and $a^* = c_0 * \sqrt{\#q} + c_1$ models.

difficulty, the number of queries converges to the quadratic number of queries needed by the classical algorithm.

C. RANDOM DENSE MDP

The simulated environment corresponds to a randomized dense MDP, with 2 possible actions $A = \{0, 1\}$, as illustrated in Figure 13. The state space is $S = \{s_0, s_1, s_2, s_3\}$. The main difference with respect to the previous model, is the dense connectivity between all possible states-action pairs, indicating highly stochastic behaviour within the environment and the hardness in solving it. Additionally, the expected reward is similar for both actions, as before.

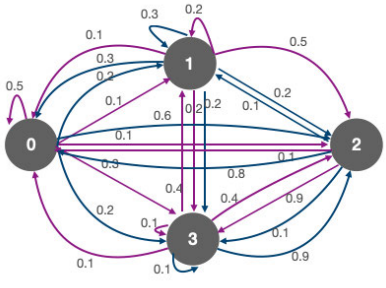

FIGURE 13. Randomized dense MDP with 2 possible actions. Action 0 indicated with purple color and action 1 illustrated with blue color.

From Figure 14, it can be seen that both classical and quantum algorithms, as opposed to the previous two cases

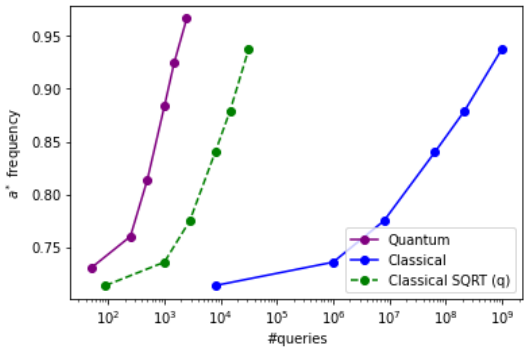

FIGURE 14. Dense MDP!: best action selection frequency as a function of the number of queries – the horizontal axis is in a log scale.

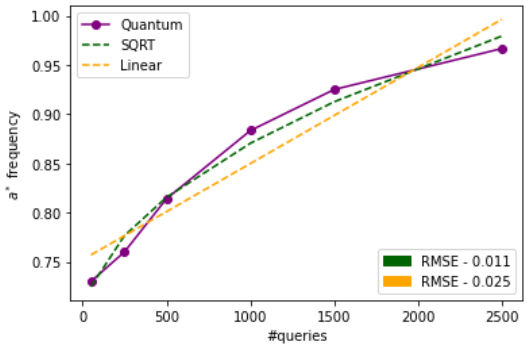

FIGURE 15. Dense MDP!: quantum data curve fitting with $a^* = c_0 * \#q + c_1$ and $a^* = c_0 * \sqrt{\#q} + c_1$ models.

studied, fail to converge to the optimal action frequency of 1, at least in the same interval of queries. This is explained not only by the stochastic nature of the environment but as well by the similarity of actions rewards. Eventually, given enough queries, both models would achieve an optimal action frequency of 1. However, as before, the quantum algorithm still shows fewer queries needed for the same frequency. From the quadratic analytical curve, it can be seen even greater proximity to the quantum curve.

The curve fitting of Figure 15 clearly shows greater proximity between a quadratic model fitting quantum data. Moreover, it can be seen a reduction in the mean squared error compared to previous models. The latter observation reinforces the hypothesis posed in Subsection VI-B i.e., increasing the difficulty of the environment, the number of queries needed by the quantum algorithm will converge to the quadratic number of queries needed by the classical algorithm, with constant terms.

VII. CONCLUSION

This paper proposes a sparse sampling inspired quantum algorithm that allows an RL-based agent to compute an ϵ-optimal action on any given state of a complex stochastic environment. This is, to the best of the authors' knowledge, the first sample-based contribution to quantum Reinforcement Learning.

Moreover, it was shown that the proposed algorithm:

- reaches an ϵ-optimal action for any state of the MDP with quadratically less computational effort than its classical counterpart [4];
- shows no dependence on the size of the MDP's state space, enabling it to efficiently deal with large RL environments.

The latter observation draws the line when comparing with other planning algorithms, specifically dynamic programming algorithms, which assume complete knowledge of the environment. The proposed quantum algorithm operates in a model-free context resorting to a sampling-based approach, thus dispensing with such complete knowledge.

The optimal number of samples required to compute ϵ-optimal actions was derived using a novel statistical approach. This approach assumes that real-world environments exhibit some locality, in the sense that the number of reachable states from any given state-action pair is much smaller than the total number of states in the MDP. If the locality assumption does not hold, that is, if the state transition graph is densely connected, then the quantum algorithm's independence on the MDP's number of states no longer holds as well.

Additionally, the characterization of the complexity of the quantum algorithm assumes that the quantum oracles that model the environment are themselves efficient. This corresponds to the relativized complexity analysis, where quantum algorithms have access to powerful oracles, whose internal structure is not examined and assumed to be $\mathcal{O}(1)$ [27]. The oracle complexity might in some cases become relevant, especially due to the cost of loading data from a classical MDP into a quantum state. This may reduce the algorithms' advantage.

The search tree is uniformly expanded over the set of possible actions, corresponding to the agent trying every possible action in every state in superposition. One interesting topic for future work would be to further reduce the search space by exploiting a non-uniform tree expansion. By using a priori knowledge from the environment, the transition operator could potentially expand the tree over a subset of prioritized actions, thus reducing the size of Grover's algorithm search space or allowing for larger look-ahead horizons.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.
[2] R. Bellman, "The theory of dynamic programming," *Bull. Amer. Math. Soc.*, vol. 60, no. 6, pp. 503–515, 1954.
[3] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2009.
[4] M. Kearns, Y. Mansour, and A. Y. Ng, "A sparse sampling algorithm for near-optimal planning in large Markov decision processes," in *Proc. IJCAI Int. Joint Conf. Artif. Intell.*, 1999, pp. 1324–1331.
[5] S. M. Kakade, "On the sample complexity of reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 1–133.
[6] V. Dunjko, J. M. Taylor, and H. J. Briegel, "Quantum-enhanced machine learning," *Phys. Rev. Lett.*, vol. 117, no. 13, pp. 1–19, Sep. 2016.
[7] V. Dunjko, Y.-K. Liu, X. Wu, and J. M. Taylor, "Exponential improvements for quantum-accessible reinforcement learning," 2017, *arXiv:1710.11160*. [Online]. Available: http://arxiv.org/abs/1710.11160

[8] S. Ying and M. Ying, "Reachability analysis of quantum Markov decision processes," *Inf. Comput.*, vol. 263, pp. 31–51, Dec. 2018.
[9] J. Barry, D. T. Barry, and S. Aaronson, "Quantum partially observable Markov decision processes," *Phys. Rev. A, Gen. Phys.*, vol. 90, no. 3, Sep. 2014, Art. no. 032311, doi: 10.1103/PhysRevA.90.032311.
[10] V. Dunjko, J. M. Taylor, and H. J. Briegel, "Advances in quantum reinforcement learning," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 282–287.
[11] E. Aïmeur, G. Brassard, and S. Gambs, "Machine learning in a quantum world," in *Proc. Can. Conf. AI*, 2006, pp. 431–442.
[12] P. Ronagh, "The problem of dynamic programming on a quantum computer," 2019, *arXiv:1906.02229*. [Online]. Available: http://arxiv.org/abs/1906.02229
[13] D. Crawford, A. Levit, N. Ghadermarzy, J. S. Oberoi, and P. Ronagh, "Reinforcement learning using quantum Boltzmann machines," *Quantum Inf. Comput.*, vol. 18, nos. 1–2, pp. 51–74, Feb. 2018.
[14] A. A. Melnikov, A. Makmal, V. Dunjko, and H. J. Briegel, "Projective simulation with generalization," *Sci. Rep.*, vol. 7, no. 1, pp. 1–14, Dec. 2017, doi: 10.1038/s41598-017-14740-y.
[15] T. Sriarunothai, S. Wölk, G. S. Giri, N. Friis, V. Dunjko, H. J. Briegel, and C. Wunderlich, "Speeding-up the decision making of a learning agent using an ion trap quantum processor," *Quantum Sci. Technol.*, vol. 4, no. 1, pp. 1–11, 2019.
[16] H. J. Briegel and G. De las Cuevas, "Projective simulation for artificial intelligence," *Sci. Rep.*, vol. 2, no. 1, pp. 1–16, Dec. 2012.
[17] V. Saggio, B. E. Asenbeck, A. Hamann, T. Strömberg, P. Schiansky, V. Dunjko, N. Friis, N. C. Harris, M. Hochberg, D. Englund, S. Wölk, H. J. Briegel, and P. Walther, "Experimental quantum speed-up in reinforcement learning agents," *Nature*, vol. 591, no. 7849, pp. 229–233, Mar. 2021, doi: 10.1038/s41586-021-03242-7.
[18] M. Schuld and F. Petruccione, *Supervised Learning With Quantum Computers*, 1st ed. Cham, Switzerland: Springer, 2018.
[19] R. LaRose and B. Coyle, "Robust data encodings for quantum classifiers," *Phys. Rev. A, Gen. Phys.*, vol. 102, no. 3, Sep. 2020, Art. no. 032420, doi: 10.1103/PhysRevA.102.032420.
[20] A. Ahuja and S. Kapoor. (1999). *A Quantum Algorithm for finding the Maximum*. [Online]. Available: http://arxiv.org/abs/quant-ph/9911082
[21] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, "Quantum amplitude amplification and estimation," *Quantum Comput. Inf.*, vol. 305, pp. 53–74, Oct. 2002, doi: 10.1090/conm/305/05215.
[22] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput. (STOC)*, 1996, pp. 212–219, doi: 10.1145/237814.237866.
[23] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, "Tight bounds on quantum searching," *Prog. Phys.*, vol. 46, nos. 4–5, pp. 493–505, 1998.
[24] E. Biham, O. Biham, D. Biron, M. Grassl, D. A. Lidar, and D. Shapira, "Analysis of generalized Grover quantum search algorithms using recursion equations," *Phys. Rev. A, Gen. Phys.*, vol. 63, no. 1, Dec. 2000, Art. no. 012310.
[25] E. B. Wilson, "Probable inference, the law of succession, and statistical inference," *J. Amer. Stat. Assoc.*, vol. 22, no. 158, pp. 209–212, 1927.
[26] C. et al Zoufal, "Qiskit: An open-source framework for quantum computing," IBM Quantum, Tech. Rep., 2021. [Online]. Available: https://quantum-computing.ibm.com/, doi: 10.5281/zenodo.2562111#.X7Meb9QFHJI.mendeley.
[27] S. Aaronson. (Jul. 2010). *Doing My Oracle Duty*. [Online]. Available: https://www.scottaaronson.com/blog/?p=451

ANDRÉ SEQUEIRA received the B.S. and M.Sc. degrees in physics engineering from the University of Minho, Braga, Portugal, in 2021, where he is currently pursuing the Ph.D. degree in computer science. His research interests include quantum information and quantum machine learning.

LUIS PAULO SANTOS (Associate Member, IEEE) received the bachelor's and M.Sc. degrees in computer science from the Department of Informatics. He is currently an Assistant Professor with the Department of Informatics, Universidade do Minho, Portugal. He lectures computer architecture and computer graphics. In 2019, he joined the International Iberian Nanotechnology Laboratory, Quantum Software Engineering Group, Braga, Portugal, as a Research Associate. He has spent several periods as an Invited Researcher on a few international institutions, such as the University of Bristol, U.K.; Warwick Manufacturing Group, University of Warwick, U.K.; the Université de Rennes I, France; and Texas Advanced Computing Center, The University of Texas at Austin, USA. He has been the Vice-Director of the Department of Informatics. He has been the Director of the Doctoral Program on informatics. He is a Senior Researcher of INESC TEC, Portugal. More recently, he became interested in quantum computing and its applications to global illumination, and graphics and numerical integration, in general. He participated in several research projects and supervised seven Ph.D. students. His research interests include global illumination and parallel processing. He has published several articles on conferences and journals within these areas of knowledge. He was the Chair of the Eurographics Portuguese Chapter, from 2016 to 2020. He acted as an Associate Editor of *Computers and Graphics* (Elsevier), from 2011 to 2019.

LUIS SOARES BARBOSA is currently a Full Professor with the Computer Science Department, Universidade do Minho, and a Senior Researcher with the High Assurance Software Laboratory, HASLab INESC TEC. He has a second academic affiliation with the United Nations University, where he serving as the Deputy Head for its Operational Unit on Policy-Driven Electronic Governance. UNU-EGOV is an international think-tank devoted to multidisciplinary research on how digital transformation may contribute to empowered democratic citizenship, trustworthy public infrastructures, more inclusive societies and, in broad terms, sustainable development. His main research interests include program semantics, logics and calculi applied to rigorous software analysis, design, and construction. He is particularly interested in the architectural dimension, such as interaction, composition, and reconfiguration of different sorts of software components, namely nondeterministic, probabilistic, quantum, continuous, or hybrid. Most of his work is framed on Coalgebra and Modal Logic. More recently, he became interested in exploring connections between physics and computation at two levels, such as the discrete-continuous frontier and the classic-quantum interaction. In this context, he joined INL, the International Iberian Nanotechnology Laboratory, a new research group in Quantum Software Engineering Group, in 2019.