K-spin Hamiltonian for quantum-resolvable Markov decision processes

Eric B. Jones,[1, 2, *] Peter Graf,[1] Eliot Kapit,[2] and Wesley Jones[1]

[1] *National Renewable Energy Laboratory, Golden, CO 80401, USA*
[2] *Department of Physics, Colorado School of Mines, Golden, CO 80401, USA*

(Dated: April 14, 2020)

The Markov decision process is the mathematical formalization underlying the modern field of reinforcement learning when transition and reward functions are unknown. We derive a pseudo-Boolean cost function that is equivalent to a K-spin Hamiltonian representation of the discrete, finite, discounted Markov decision process with infinite horizon. This K-spin Hamiltonian furnishes a starting point from which to solve for an optimal policy using heuristic quantum algorithms such as adiabatic quantum annealing and the quantum approximate optimization algorithm on near-term quantum hardware. In proving that the variational minimization of our Hamiltonian is equivalent to the Bellman optimality condition we establish an interesting analogy with classical field theory. Along with proof-of-concept calculations to corroborate our formulation by simulated and quantum annealing against classical Q-Learning, we analyze the scaling of physical resources required to solve our Hamiltonian on quantum hardware.

I. INTRODUCTION

Alongside supervised and unsupervised learning, reinforcement learning (RL) constitutes one of the main pillars of modern attempts to create artificial intelligence [1]. RL describes a set of techniques, anchored in the mathematical formalism of the Markov decision process (MDP), whereby an autonomous agent learns how to behave according to an optimal policy within an environment in order to maximize some notion of expected future rewards derived from said environment by repeatedly interacting with it. When the agent finds itself in some state of the environment, it chooses an action that causes it to transition to a subsequent state of the environment and simultaneously be given a reward (or penalty). The agent's behavioral policy is therefore typically regarded as a probability distribution over state-action pairs, which tells the agent which action(s) is (are) best to take in each state. RL has begun to see use in a wide variety of applications including recommendation systems and optimal control, and perhaps the most stark demonstration of the computational utility of RL has been its role in enabling a computer to beat expert human players at the game of Go [2–4].

While RL is a novel and powerful algorithmic paradigm, its implementation currently remains largely tied to computational resources that are dictated by classical physics at the hardware level. Given that the space of possible policies for an agent to enumerate is often combinatorially large, such as in the instance of Go, there is good reason to suspect that the particular type of parallelism afforded by quantum hardware might help an agent in finding an optimal policy more quickly. In fact, an existing body of literature already explores quantum formulations of RL in situations where either the agent or environment, or both, are regarded as quantum mechanical, and in some instances a quantum speedup has been established (see for example [5–12]). To our knowledge however, there does not yet exist a formulation of a quantum MDP with input from a classical environment that is amenable to solution directly on near-term, noisy intermediate-scale quantum hardware via existing quantum optimization algorithms.

In this work, we therefore establish what we regard to be the simplest and most straightforward connection between the mathematical formalism of RL and that of quantum computing. Namely, we consider a "quantum-accelerated" agent interacting with a classical environment. Moreover, we suppose that the agent possesses a model of its environment such that the quantum MDP consists only of the agent solving for an optimal policy using quantum hardware. Strictly speaking, this formulation in and of itself does not constitute "quantum RL" because the agent already has a model of its environment. However, paired with a classical routine where the agent learns the transition and reward rules for its environment, our quantum-resolved MDP may be thought of as a (potentially hybrid) quantum subroutine embedded in a larger hybrid quantum-classical RL algorithm.

Note that all three variants of the general MDP (finite horizon, infinite horizon discounted, and infinite horizon averaged cost) have been shown to be P-complete [13]. This means roughly that while MDPs are in principle efficiently solvable in polynomial time, they are not categorically efficiently parallelizable [14]. Moreover, a general MDP suffers from the "curse of dimensionality", meaning that its attendant memory requirements grow exponentially in the dimension of the state space. In practice, a conventional computer can solve MDPs involving millions of states [1].

Our approach is motivated by the observation that a majority of near-term quantum algorithms such as adiabatic quantum annealing and the quantum approximate optimization algorithm are concerned with the solution of discrete, combinatorial optimization problems recast

[*] ebjones@mymail.mines.edu; Eric.Jones@nrel.gov

as K-spin Hamiltonians of the form [15, 16]

$$\hat{H} = -\sum_{k=1}^{K} \sum_{j_1 \ldots j_k = 1}^{N} J_{j_1 \ldots j_k} \hat{Z}_{j_1} \ldots \hat{Z}_{j_k}. \tag{1}$$

As such, we identify the policy with discrete variables to be mapped to spins (and thus qubits) and a sum over Q-functions with a K-spin Hamiltonian. Since it is well-known that finding the ground state of a K-spin Hamiltonian is in the complexity class NP-complete through its polynomial-time order reduction to the Ising spin Hamiltonian, it may at first glance look odd to map a problem from P-complete into a formulation, which ostensibly resides in NP-complete [17, 18]. Our primary justification is that certain MDP generalizations, such as the partially observable MDP (POMDP), are harder than the MDP- for example POMDP is PSPACE-complete [13]. The present work provides a platform from which to explore K-spin Hamiltonians as an approximation scheme for problems such as POMDP. In addition, while *arbitrary* instances of the K-spin problem are NP-complete there can be *specific* instances of the K-spin problem, which are soluble in polynomial time [19, 20]. It is possible that the K-spin Hamiltonian derived in the present work inherits its polynomial-time solubility from the dynamic programming formulation from which it is derived. Suggesting otherwise however, is the fact that when reformulated into Ising spin form, the k-regular k-XORSAT problem has been shown to take exponential time to solve by adiabatic quantum annealing when $k = 3$ while Gaussian elimination allows polynomial-time solubility classically [21–23]. We therefore do not consider our quantum MDP a panacea for all large-order polynomial-time MDPs. Rather, it furnishes an interesting connection between two disparate fields in computational science while allowing for quantum parallelization of a problem, which is not efficiently parallelizable on a classical computer (unless $P = NC$).

II. MARKOV DECISION PROCESS

The basic mathematical formalism that describes an autonomous agent interacting with its environment is the MDP. Our treatment follows from the standard text by Sutton and Barto [1]. We take the discrete, finite, discounted MDP with infinite horizon to be a 5-tuple (S, A, P, R, γ) where S is a finite set of discrete states in which the agent may find itself, A is a finite set of discrete actions that the agent may carry out, P is a set of transition probabilities of the form $P^a_{ss'}$, which describe the probability that an agent will transition to state s' from state s given that it has taken action a, R is a set of rewards of the form $R^a_{ss'}$, which describe the reward that an agent receives upon moving from state s to state s' under the action a, and γ is the discount factor, which gauges the degree to which rewards farther into the future are prioritized less than more immediate ones.

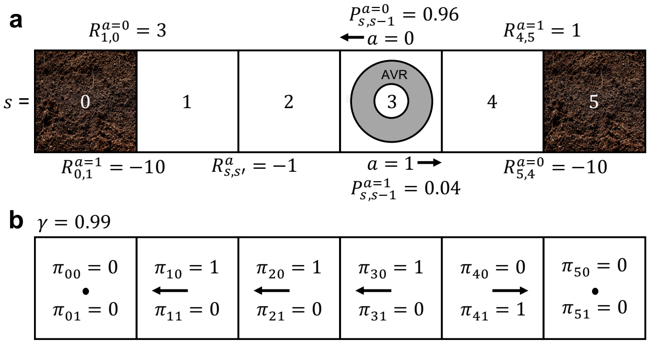

FIG. 1. The Markov decision process. **a** AVR in a hallway trying to find the larger dirt pile to clean formulated as an MDP. **b** Optimal policy at discount factor $\gamma = 0.99$.

The goal of an agent moving around in an environment is to learn an optimal policy π_{sa} such that when it finds itself in the state s, π_{sa} is the probability that the agent should take the action a in order to maximize its expected future discounted reward. Formally, there are two quantities that represent an expected future discounted reward under the policy π: the value function

$$V_s[\pi] = \mathbb{E}_\pi[G_t | S_t = s], \tag{2}$$

which describes the future discounted reward expected by virtue of being in state s at time t, and the action-value function (also known as the Q-function)

$$Q_{sa}[\pi] = \mathbb{E}_\pi[G_t | S_t = s, A_t = a], \tag{3}$$

which describes the future discounted reward expected by choosing action a when starting in state s at time t. In either case, the quantity

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{4}$$

is the discounted return expected by the agent from time t onward. Intuitively, the agent wants to collect as many of the largest, positive-valued $R^a_{ss'}$ as possible and as few negative-valued ones over some effective time horizon dictated by the value of γ. If the sets P and R are known a priori the agent is said to have a model of its environment. If not, the agent must learn features of its environment implicitly or through a training simulation, which are situations encompassed by the MDP. To make these concepts concrete consider the particular example of an autonomous vacuuming robot (AVR) in a hallway trying to find the larger of two dirt piles to clean as displayed in Fig. 1a. The states accessible to the AVR, $s \in S = \{0, \ldots, 5\}$, are the six floor tiles, and the action space is $A = \{0, 1\}$ with $a = 0$ corresponding to moving left and $a = 1$, right. The terminal states, $s = 0$ and $s = 5$, correspond to dirt piles that the AVR wants to find and clean. As such, if it encounters a dirt pile, the AVR receives the rewards $R^1_{4,5} = 1$ (the smaller pile) and

$R^0_{1,0} = 3$ (the larger pile). If we suppose that each AVR is responsible for finding and cleaning only one dirt pile, then we can entice the AVR to stay at the dirt pile it has found by introducing a steep penalty for moving away $R^0_{5,4} = R^1_{5,5} = R^1_{0,1} = R^0_{0,0} = -10$. Note that one could also immobilize the AVR with $P^a_{0,0} = P^a_{5,5} \sim 1$. Each step in the interior of the hallway, say between $s = 2$ and 3, incurs a penalty $R^a_{s,s' \in interior} = -1$ so that the AVR does not dawdle. We specify that this process is slightly stochastic in that $P^0_{s,s-1} = P^1_{s,s+1} = 0.96$ and $P^1_{s,s-1} = P^0_{s,s+1} = 0.04 \; \forall s$. In this context, stochasticity could reflect a tendency of the AVR to occasionally misfire and move in the wrong direction. We make this designation because the deterministic MDP is even easier (NC complexity class) than the general, stochastic MDP [13]. At the edges of the hallway the AVR sees relatively hard boundary conditions via $P^0_{0,0} = P^1_{5,5} = 0.96$. We also regard the policy, π, as a binary indicator variable such that $\pi_{sa} = 1$ if the agent takes action a in state s and $\pi_{sa} = 0$ if it does not. Such a policy is called a deterministic policy. However, the K-spin Hamiltonian derived in Sec. III is valid for non-deterministic policies as well since any probabilistic policy expressed as a fixed-precision or floating-point number has a linear expansion in binary variables and thus can be mapped linearly to qubit degrees of freedom [24]. The use of deterministic policy variables simply allows a more economical use of qubit resources. We remark that this environment is equivalent to a 1-dimensional version of the classic "Gridworld" environment encountered in the RL literature [1].

As established, one way for an agent to understand the quality of its current policy is by calculating the action-value function $Q_{sa}[\pi]$ in Eq. 3. For a given policy, the action-value function returns the expected future discounted reward for taking action a in state s and then subsequently following π. An optimal policy is one for which $Q_{sa}[\pi]$ is maximized $\forall\ s, a$. One of the central results of the dynamic programming treatment of MDPs is that the action-value function obeys a Bellman equation

$$Q_{sa}[\pi] = \sum_{s'} P^a_{ss'} \left(R^a_{ss'} + \gamma \sum_{a'} \pi_{s'a'} Q_{s'a'}[\pi] \right). \tag{5}$$

We use Eq. 5 as the starting point from which to derive a K-spin Hamiltonian for a general discrete, finite, discounted MDP over an infinite horizon.

III. MDP K-SPIN HAMILTONIAN

One can express the action-value function purely as a functional of the policy by repeatedly substituting the left-hand side of Eq. 5 into the right-hand side of Eq. 5.

TABLE I. Analogy between classical field theory and the MDP.

Object	Classical Field Theory	MDP
Coordinate	$x = (x^0, x^1, x^2, x^3)$	$\mu = (s, a)$
Field	$\Phi(x)$	π_μ
Density	$\mathcal{L}(\Phi(x), x)$	$Q_\mu[\pi]$
Functional	$S[\Phi] = \int d^4x \mathcal{L}(\Phi(x), x)$	$H[\pi] = \sum_\mu Q_\mu[\pi]$
Stationarity	$\delta S = 0$	$\delta H = 0$
Euler-Lagrange	$\delta \mathcal{L}/\delta \Phi = 0$	$\delta Q/\delta \pi = 0$

The result is

$$\begin{aligned} Q_{sa}[\pi] &= \sum_{s'} P^a_{ss'} R^a_{ss'} + \sum_{s',a'} \pi_{s'a'} \left(\gamma \sum_{s''} P^a_{ss'} P^{a'}_{s's''} R^{a'}_{s's''} \right) \\ &+ \sum_{s',a',s'',a''} \pi_{s'a'} \pi_{s''a''} \left(\gamma^2 \sum_{s'''} P^a_{ss'} P^{a'}_{s's''} P^{a''}_{s''s'''} R^{a''}_{s''s'''} \right) \\ &+ \ldots . \end{aligned} \tag{6}$$

Q is now a functional of the policy. A stationary point of a functional occurs whenever its variation with respect to its argument vanishes [25]. Therefore finding the optimal policy that maximizes the Q-function everywhere is equivalent to the variational condition

$$\frac{\delta Q_{sa}[\pi]}{\delta \pi_{\bar{s}\bar{a}}} = 0 \tag{7}$$

for an arbitrary variation of Q with respect to the policy. From here it is straightforward to define a Hamiltonian functional as

$$H[\pi] \equiv -\sum_{s,a} Q_{s,a}[\pi]. \tag{8}$$

Recall here that $\pi_{sa} \in \{0, 1\}$ and therefore has a simple relationship to single-qubit spin operators via $\pi_{sa} \rightarrow (\hat{\mathbb{1}}_{sa} - \hat{Z}_{sa})/2$. So, Eq. 8 is equivalent to a K-spin Hamiltonian where γ^K multiplies the largest order in the infinite expansion that needs to be taken into account in order to find the optimal policy.

The Hamiltonian in Eq. 8 is amenable to simulation on a quantum computer either via adiabatic or variational preparation. What one needs to check is whether minimizing H with respect to π is equivalent to maximizing Q_{sa} for each s and a. For this purpose, consider an arbitrary variation of the Hamiltonian with respect to the policy. Minimization of the Hamiltonian corresponds to

$$\begin{aligned} \frac{\delta H[\pi]}{\delta \pi_{\bar{s}\bar{a}}} &= 0 \\ &= -\sum_{s,a} \frac{\delta Q_{sa}[\pi]}{\delta \pi_{\bar{s}\bar{a}}}. \end{aligned} \tag{9}$$

But since the variation is arbitrary, each term in the sum needs to vanish identically leading to Eq. 7 for all $s, a, \bar{s}, \bar{a}$.

There is an interesting analogy to classical field theory in the present discussion shown in Table I. Each state-action pair $\mu = (s, a)$ plays the role of a space-time coordinate over which a field π_μ is defined. The action-value function can loosely be thought of similar to a Lagrangian density. The Hamiltonian functional plays a role analogous to the action functional of field theory and its variation leads to an "Euler-Lagrange" equation for Q. We note that there exist path integral approaches to both optimal control and reinforcement learning, and Markov chains have been explored as tools in field theory [26–28]. It would be an interesting direction for future work to explore the possible connections between our K-spin theory and these previous approaches. In addition, we emphasize that by expressing the MDP in the form of Eq. 8 (along with the conditional probability normalization below in the second line of Eq. 10), a quantum optimization heuristic will attempt to produce the optimal policy for the entire state-action space at once by virtue of the fact that the Hamiltonian contains in its parameters a model for the entire environment. In the absence of a known model, such as in the instance where the state-action space is too large to allow tabulation of all transition and reward functions or where such functions simply are not known a priori, we expect that one should still be able to write down a Hamiltonian whose ground state infers a good policy by sampling transition and reward functions from across the state-action space. We leave a demonstration of this concept for future work as well, since it is beyond the scope of the present paper.

Explicitly then, the Hamiltonian we may use to solve an MPD for an optimal policy is

$$\begin{aligned} H[\pi] = &-\sum_{k=0}^{\infty} \sum_{\mu_1 \ldots \mu_k = 1}^{|S \times A|} J_{\mu_1 \ldots \mu_k} \pi_{\mu_1} \ldots \pi_{\mu_k} \\ &+ M \sum_{s_1=1}^{|S|} \left(\sum_{a_1=1}^{|A|} \pi_{s_1 a_1} - 1 \right)^2 \end{aligned} \tag{10}$$

where we have made the identifications $s(a) \to s_0(a_0)$, $s'(a') \to s_1(a_1)$, etc. and

$$\begin{aligned} J_{\mu_1 \ldots \mu_k} &\equiv J_{(s_1,a_1)\ldots(s_k,a_k)} \\ &\equiv \gamma^k \sum_{s_0, a_0, s_{k+1}} P^{a_0}_{s_0 s_1} \ldots P^{a_k}_{s_k s_{k+1}} R^{a_k}_{s_k s_{k+1}}. \end{aligned} \tag{11}$$

Eq. 10 includes a penalty function with strength M in the second line due to the fact that the policy π_{sa} is generally a probability distribution conditioned on the state variable s and so must equal unity when summed over all actions ($\sum_{a_1} \pi_{s_1 a_1} = 1 \ \forall \ s_1$). Also, note that the $k = 0$ term involves a slight abuse of notation and is actually just a constant offset to the Hamiltonian, which otherwise displays the same form as Eq. 1 with $K \to \infty$. The quantities defined in Eq. 11 are $k-$qubit coupling coefficients, which gauge how many terms in the Hamiltonian need to be retained in order to solve for an optimal policy. Even though γ^k monotonically decreases with k since $0 < \gamma < 1$, it is not clear a priori that Eq. 1 is convergent for a general MDP problem instance. However, even if Eq. 1 is an asymptotic series in some instances, it may still provide useful solutions as is well known within the context of perturbative quantum field theory [29]. We take $k = K$ as the order at which one is practically able to truncate the series.

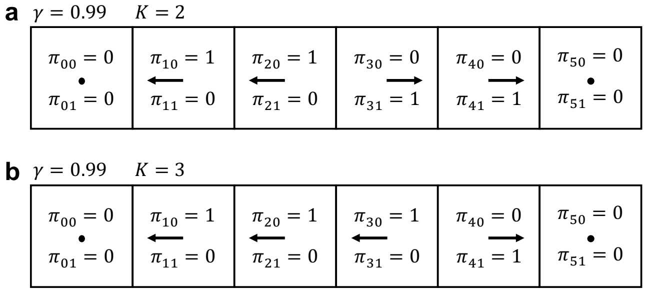
a $\gamma = 0.99$ $K = 2$
$\pi_{00} = 0$ $\pi_{01} = 0$
$\pi_{10} = 1$ $\pi_{11} = 0$
$\pi_{20} = 1$ $\pi_{21} = 0$
$\pi_{30} = 0$ $\pi_{31} = 1$
$\pi_{40} = 0$ $\pi_{41} = 1$
$\pi_{50} = 0$ $\pi_{51} = 0$
b $\gamma = 0.99$ $K = 3$
$\pi_{00} = 0$ $\pi_{01} = 0$
$\pi_{10} = 1$ $\pi_{11} = 0$
$\pi_{20} = 1$ $\pi_{21} = 0$
$\pi_{30} = 1$ $\pi_{31} = 0$
$\pi_{40} = 0$ $\pi_{41} = 1$
$\pi_{50} = 0$ $\pi_{51} = 0$

FIG. 2. Effects of prematurely truncating the K-spin Hamiltonian. **a** At $\gamma = 0.99$ and $K = 2$ for a 6-tile state space and asymmetric terminal rewards specified in Sec. II, the optimal policy is erroneously symmetric. Note however, that this policy is the true ground state for $\gamma = 0.8$ and $K = 2$. **b** When K is raised to 3, the AVR accurately identifies longer-term rewards and moves left at the fourth tile from left instead of moving right.

In terms of hardware requirements, a larger K will require higher gate complexity and depth for gate-model machines and larger numbers of ancillary qubits for annealer-model machines in order to perform order-reduction [30, 31]. Gate-model machines may also use ancillary variables in order to reduce gate complexity. In addition, if K is chosen to be too small with respect to the decay of γ^k then the ground state of the Hamiltonian will fail to exhibit some long-range correlations that might be essential to finding an optimal policy at every state-action pair. We will discuss hardware resource requirements for accurate policy determination in more detail in Sec. IV. Fig. 1b shows the optimal policy found by solving for the ground state of H for the AVR-dirt scenario described in Sec. II on the D-Wave 2000Q processor and validated by simulated annealing and classical Q-Learning run in an OpenAI Gym environment [32]. For the Q-Learning hyperparameters, we use $\alpha = 0.1$ (learning rate), $\epsilon = 0.1$ (convergence threshold), and $N_{ep} = 20,000$ (number of training episodes). A more detailed discussion of quantum and simulated annealing and their hyperparameters will follow in Sec. V. The strength of the penalty function in Eq. 10 is set to $M = 3$, the discount factor is set to $\gamma = 0.99$, and the Hamiltonian is truncated at $K = 3$ and then reduced to quadratic order, which corresponds to a quadratic unconstrained binary optimization (QUBO) or Ising model. The procedure for Hamiltonian order reduction will be discussed in Sec. IV, but we remark here that we set the penalty strength required for order reduction to $M_{OR} = 5$. Fig. 2 shows how premature truncation adversely affects solu-

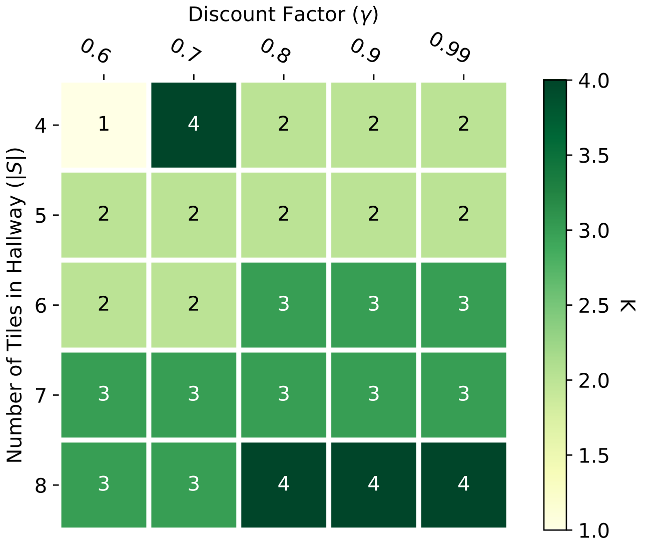

FIG. 3. Lowest order (K) at which Eq. 10 can be truncated and still find the unique, optimal policy as a function of system size ($|S|$) and discount factor (γ) for the environment described in Sec. II. The truncation order scales roughly as $K \sim |S|/2$.

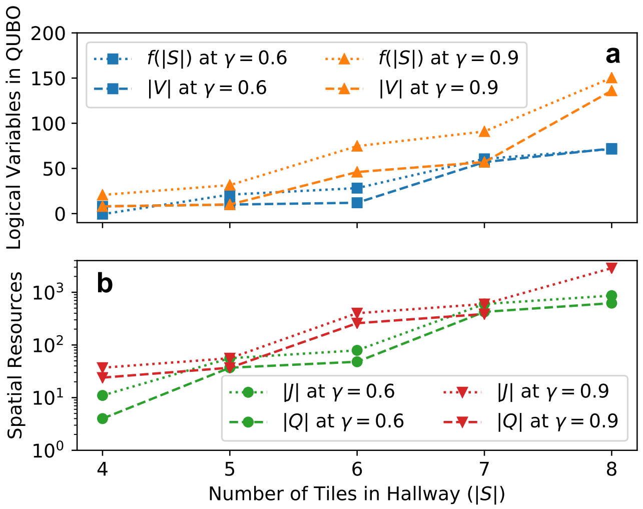

FIG. 4. Scaling of spatial resources for quantum annealing. **a** Number of logical qubits ($|V|$, dashed lines for overall trend) needed to encode the truncated Hamiltonian at quadratic order as a function of system size ($|S|$). Dotted lines represent the fitted theoretical scaling function for order reduction by substitution, $f(|S|) = 3\gamma|S \times A|K - 25\gamma$. Orange curves with triangles denote $\gamma = 0.9$ while blue curves with squares denote $\gamma = 0.6$. **b** Dotted lines represent the number of coefficients ($|J|$) needed to represent the truncated, order-reduced Hamiltonian. Dashed lines represent the number of physical qubits ($|Q|$) required to minor embed the truncated, order-reduced Hamiltonian on the D-Wave 2000Q (16,4)-Chimera hardware graph. Red curves with inverted triangles are at $\gamma = 0.9$ while green curves with circles are at $\gamma = 0.6$.

tion quality for the same environment. In Fig. 2a, truncation at $K = 2$ erroneously leads to a policy that is symmetric in the state space due to the fact that terms, which couple policy variables in the fourth tile from left to the left-most tile are neglected. By truncating instead at $K = 3$, the AVR becomes aware of longer-term rewards and therefore identifies the correct policy. Due to this observation, we turn to an analysis of the scaling of spatial resources (i.e. numbers of qubits and Hamiltonian parameters) in Sec. IV before assessing the scaling in time complexity of the K-spin MDP in Sec. V.

IV. HARDWARE REQUIREMENTS FOR THE K-SPIN MDP

Given that the order ($k = K$) at which Eq. 10 is truncated affects whether or not the ground-state solution equates to the optimal policy, a natural question to ask is: how many terms in the Hamiltonian need to be retained? This question is also important since either the number of ancilla qubits or gate complexity or both will grow as a function of K. Fig. 3 shows the lowest truncation order (K) at which the optimal policy was found as the unique ground state of Eq. 10 as a function of system size ($|S|$) and discount factor (γ) by simulated annealing. For our hallway environment, we find that the truncation order scales roughly as $K \sim |S|/2$ for typical values of the discount factor such as $\gamma = 0.9$, but for a general environment we expect that this scaling will depend upon the form of the transition and reward probabilities P and R as well as the dimensions of the state and action spaces S and A. The aberrant $K = 4$ result observed at ($|S| = 4, \gamma = 0.7$) is a consequence of the small size of the state space combined with a small discrepancy in how the boundary conditions are implemented classically. Boundary conditions matter less as the state space becomes larger. Otherwise, Fig. 3 is encouraging for two reasons. First, while Eq. 10 formally involves an infinite number of terms, Fig. 3 confirms that in practice, only a finite number of terms need to be retained in order to find an optimal policy. This situation is similar to the dynamic programming treatment for MDPs wherein only a finite number of policy iterations are needed in order to find an optimal policy [1]. And second, we see that as the discount factor (γ) is varied, our Hamiltonian's ground state is able to track the optimal policy by simply including higher-order interactions. For example, the $K = 2 \rightarrow 3$ phase transition observed between $\gamma = 0.7$ and $\gamma = 0.8$ for $|S| = 6$ corresponds to the ground state policy switching from the one in Fig. 2a to the policy in Fig. 2b.

We now turn to an accounting of physical resources required to simulate our Hamiltonian on quantum hardware. In its current form, i.e. before the substitution to qubit operators $\pi_{sa} \rightarrow (\hat{\mathbb{1}}_{sa} - \hat{Z}_{sa})/2$, Eq. 10 is in the multilinear polynomial form of a pseudo-Boolean classical cost function. There is a useful result from combinatorial optimization that an arbitrary pseudo-Boolean function with order (degree) K can be reduced to a quadratic-order polynomial in polynomial time [18]. There are a variety of approaches to polynomial order reduction, which typically require the introduction of ancillary variables [33]. Here we discuss the most general but worst-scaling technique for order reduction in order to place upper bounds on the quantum hardware required to solve a general MDP. This technique is referred to as substitution.

Substitution for order reduction was first introduced by Rosenberg, and it is predicated on the observation that for three binary variables $x, y, z \in \{0, 1\}$, relationships such as

$$xy = (\neq)\, z \quad \text{iff} \quad xy - 2xz - 2yz + 3z = (>)\, 0 \tag{12}$$

hold [18, 31]. Therefore, one may rewrite a product of two variables xy as a single variable z while at the same time introducing the penalty function

$$P = M_{OR}(xy - 2xz - 2yz + 3z) \tag{13}$$

into the problem Hamiltonian to enforce $xy = z$, where M_{OR} is the strength of the penalty function. If each of $|S \times A|$ variables occurs in Hamiltonian terms with at most $K - 1$ other binary variables, then the number of additional qubits needed to quadratize the problem scales as $O(|S \times A|K)$, where $|S \times A|$ is the magnitude of the MDP state-action space and K is the order at which Eq. 10 is truncated [34]. Fig. 4a shows the number of logical qubits ($|V|$) required to simulate the truncated, order-reduced Hamiltonian as a function of system size for two different discount factors, $\gamma = 0.6$ (blue squares) and $\gamma = 0.9$ (orange triangles). Dashed lines connect points calculated by using the D-Wave *make_quadratic* utility while dotted lines connect points calculated by the function $f(|S|) = 3\gamma|S \times A|K - 25\gamma$, which in the asymptotic limit corroborates the $O(|S \times A|K)$ scaling of order reduction by substitution. This scaling is general for a fixed γ and independent of the hardware platform used. The dependence of $|V|$ on γ is a result of the fact that, on average, for larger discount factors more terms in Eq. 10 need to be taken into account in order to find the correct policy. Once order-reduced, these terms manifest as additional logical variables.

Fig. 4b shows the scaling of computational resources required to simulate the truncated, order-reduced Hamiltonian as a function of system size on the D-Wave 2000Q processor in particular for $\gamma = 0.6$ (green circles) and $\gamma = 0.9$ (red inverted triangles). Dotted lines connect points that denote the number of parameters ($|J|$) of the form in Eq. 11 needed to specify the Hamiltonian after order reduction while dashed lines connect points that denote the number of physical qubits ($|Q|$) required to simulate the order-reduced Hamiltonian after being minor-embedded on the D-Wave (16,4)-Chimera hardware graph using the *find_embedding* utility. The *find_embedding* utility was run ten times with the best resultant embedding taken. Note that the point $|Q|(|S| = 8, \gamma = 0.9)$ is missing since $|J|(|S| = 8, \gamma = 0.9) = 2,740$ and so the order-reduced Hamiltonian cannot be embedded on the $2,048$ qubit processor.

As observed in previous work, $|J|$ and $|Q|$ track closely to one another over multiple orders of magnitude [35]. In addition, at $\gamma = 0.6$ and $\gamma = 0.9$ as shown, $|Q|$ and $|V|$ both increase by roughly two orders of magnitude between $|S| = 4$ and $|S| = 8$. This scaling similarity sits well within the worst-case asymptotic $O(|V|^2)$ scaling for embedding a general complete graph ($K_{|V|}$) into a Chimera graph using logical chains [36]. Fig. 4b therefore shows that in addition to being amenable to the type of quantum parallelism exhibited by conventional quantum optimization hardware and heuristics, Eq. 10 may suffer from a less severe curse of dimensionality when compared to the classical MDP.

The process of order reduction is necessary in order to solve for the ground state of Eq. 10 on conventional quantum annealers since execution of an annealing schedule is beholden to representing qubit-qubit interactions on the native two-qubit coupling in hardware [37]. By contrast, gate-model quantum computers can decompose higher-order interactions via gate compilation into products of single and two-qubit gates [38]. Specifically, after truncating and elevating Eq. 10 to an operator and exponentiating it, we find that for a quantum approximate optimization algorithm (QAOA)-like unitary with variational parameter β, we have

$$\begin{aligned} e^{-i\beta\hat{H}} = \prod_{k=0}^{K} \prod_{\mu_1\ldots\mu_k=1}^{|S\times A|} \Bigg[& \hat{\mathbb{1}}_{\mu_1} \otimes \ldots \otimes \hat{\mathbb{1}}_{\mu_k} \\ & + \left(e^{i\beta J_{\mu_1\cdots\mu_k}} - 1\right) \hat{\pi}_{\mu_1} \otimes \ldots \otimes \hat{\pi}_{\mu_k} \Bigg]. \end{aligned} \tag{14}$$

Eq. 14 is equivalent to a product of $(k - 1)$-qubit controlled-PHASE ($e^{i\beta J_{\mu_1\cdots\mu_k}}$) gates. Without introducing any additional ancillary qubits, a general N-qubit controlled-unitary may be decomposed with $O(2^N)$ two-qubit gates [39]. Therefore, without exploiting any hardware-specific implementations, we estimate that for a QAOA of depth p, the contribution of Eq. 14 to the gate volume scales at worst as $O(p2^{K|S\times A|^K})$. However, either by introducing ancillary qubits or by hardware specific implementations, scaling of the gate volume can be reduced to $O(pK|S \times A|)$ or better [40]. Keeping in mind that the hardware resources required to solve Eq. 10 are very much architecture dependent, we now turn to a particular experimental assessment of the time complexity of solving the K-spin Hamiltonian by classical simulated annealing and the variant of quantum annealing (transverse field Ising model) offered by the D-Wave 2000Q processor [41].

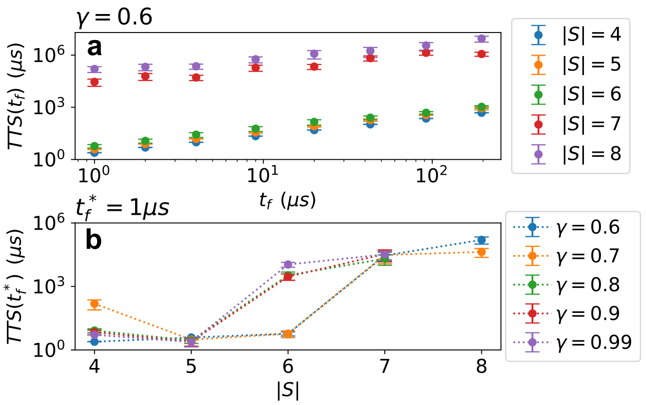

FIG. 5. Time to solution (TTS) for quantum annealing. **a** TTS as a function of annealing time (t_f) for different system sizes ($|S|$) at $\gamma = 0.6$. Optimal TTS occurs at (or below) $t_f^* = 1\mu s$ for all system sizes, which is representative of all other values for gamma. Error bars represent one standard deviation from the mean. **b** Optimal time-to-solution ($TTS(t_f^*)$) as a function of system size ($|S|$) for different values of γ. Error bars represent one standard deviation from the mean.

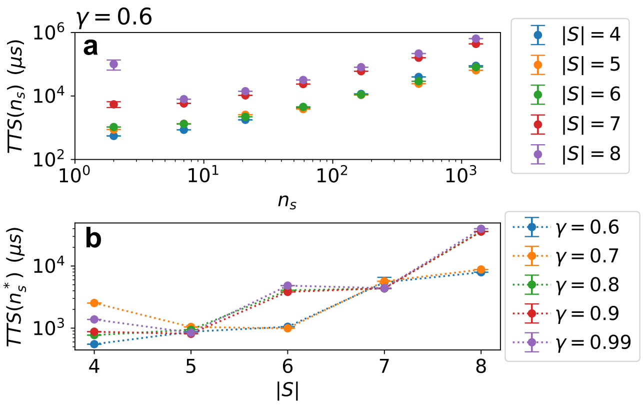

FIG. 6. Time to solution (TTS) for simulated annealing. **a** TTS as a function of number of Monte Carlo sweeps (n_s) for different system sizes ($|S|$) at $\gamma = 0.6$. Optimal TTS now generally depends upon both $|S|$ and γ. Error bars represent one standard deviation from the mean. **b** Optimal time-to-solution ($TTS(n_s^*)$) as a function of system size ($|S|$) for different values of γ. Error bars represent one standard deviation from the mean with most being smaller than the marker size.

V. TIME COMPLEXITY OF K-SPIN MDP

A. Quantum Annealing

A common and well-defined time-to-solution (TTS) metric for the D-Wave 2000Q quantum annealer is

$$TTS(t_f) = t_f \frac{\ln(1-p_d)}{\ln(1-p_s(t_f))}, \tag{15}$$

where here $t_f \in [1\mu s, 200\mu s]$ is the time chosen over which each annealing schedule is carried out, p_d is a desired success probability often taken to be 0.99, and for a given annealing time, $p_s(t_f)$ is the probability of success for a single run of the annealer [16, 42]. For both quantum annealing calculations and simulated annealing calculations, we set $M = 3$ and $M_{OR} = 5$. These values are chosen such that the penalty functions for conditional probability conservation of the policy (second line in Eq. 10) and consistency terms from order-reduction (Eq. 13), respectively, are costly to violate. We note that the particular values of $M = 3$ and $M_{OR} = 5$ are somewhat arbitrary in that any $M \approx M_{OR} \gtrsim 2$ will make the attendant penalty functions sufficiently costly to violate. In each problem instance, the chain strength to enforce the logical consistency of embedded variables was set to $J_{chain} = 1.5|J|_M$, where $|J|_M$ is the largest magnitude coupling term in the order-reduced QUBO. This value was chosen so that the fraction of broken logical chains was less than 0.001.

Fig. 5a shows a representative set of curves for $TTS(t_f)$ as a function of annealing time at $\gamma = 0.6$. At each system size, it is clear that the optimal time-to-solution $TTS(t_f^*)$ occurs at (or below) $t_f^* = 1\mu s$, which is the shortest annealing time available on the processor used. This is also the case for all other values of γ. As such, Fig. 5b shows $TTS(t_f^*)$ as a function of $|S|$ for all values of γ. While $TTS(t_f^*)$ actually appears to decrease between $|S| = 4$ and 5 for $\gamma \geq 0.7$, likely due to the boundary condition effect for small $|S|$ mentioned in Sec. IV, $TTS(t_f^*)$ between the rest of the values of $|S|$ appears to follow a sort piecewise polynomial scaling where large jumps in $TTS(t_f^*)$ are concurrent with large jumps in the number of physical qubits ($|Q|$) in Fig. 4b. Indeed, the dependence of $TTS(t_f^*)$ on γ is a direct result of the dependence of $|V|$ and therefore $|Q|$ on γ. However, that $TTS(t_f^*)$ ranges over five orders of magnitude while stepping through just a twofold increase in $|S|$ speaks to asymptotic exponential scaling of the solution of Eq. 10 by quantum annealing on the D-Wave 2000Q, in clear analogy with solving k-regular k-XORSAT– another polynomial-time solvable problem– by quantum annealing [21–23].

B. Simulated Annealing

An analog to Eq. 15 for simulated annealing is

$$TTS(t_a) = t_a \frac{\ln(1-p_d)}{\ln(1-p_s(t_a))}, \tag{16}$$

where for large $n_s \cdot N$, $t_a = n_s \cdot N/f$, but in general $t_a = t_a(n_s)$ [43]. Here, n_s is the total number of Monte Carlo sweeps (that is, the number of times all spins see an attempted update at a given temperature times the number of temperature steps) involved in a particular annealing run, N is the number of logical spins needed to represent the system under consideration, and f is the attempted spin update frequency. Note that since f is a fixed property of the implementation and classical hardware being used across all problem instances and since $N = |V|$ is fixed by the problem instance, we consider $TTS(n_s)$ in Fig. 6a. In this plot, for $\gamma = 0.6$ and $|S| = 8$, $n_s^* \approx 7$ while for all other system sizes at this discount factor $n_s \approx 2$. In general for the simulated annealing algorithm presented here n_s^* will depend upon both $|S|$ and γ. The simulated annealing implementation used here was the D-Wave *SimulatedAnnealingSampler* with default, linearly interpolated $\beta = 1/k_BT$ values run on a single 2.5 GHz Intel Core i5 processor with 16 GB of 1600 MHz DDR3 memory (source code at [44]). Taken together with the results in Fig. 5b, the results in Fig. 6b support the likely asymptotically exponential time complexity for solving the MDP as a spin Hamiltonian.

VI. CONCLUSION

Here we have established a direct link between the Markov decision process formalism of reinforcement learning and a quantum Hamiltonian amenable to solution on current-generation and near-term quantum hardware. Our derivation demonstrates that some problems that resist parallelization classically may become manifestly parallel when recast into the quantum model of computation. Despite this parallelization, the K-spin Hamiltonian likely scales exponentially in time when solved using conventional simulated and quantum annealing resources. Moreover, spatial fully-connected quantum computational resources scale polynomially in the size of the state space, while limited hardware connectivity induces a polynomial growth in spatial resources that is at most quadratically worse than the case for fully-connected resources. Despite these limitations, our K-spin Hamiltonian provides another previously unexplored inroad to quantum formulations of full reinforcement learning as well as more difficult variants of the MDP such as POMDP.

ACKNOWLEDGMENTS

This work was authored in part by the National Renewable Energy Laboratory (NREL), operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. This work was supported by the Laboratory Directed Research and Development (LDRD) Program at NREL. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes. This research used Ising, Los Alamos National Laboratory's D-Wave quantum annealer. Ising is supported by NNSA's Advanced Simulation and Computing program. The authors would like to thank Scott Pakin and Denny Dahl. This material is based in-part upon work supported by the National Science Foundation under Grant No. PHY-1653820.

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction* (MIT press, 2018).

[2] N. Golovin and E. Rahm, Reinforcement learning architecture for web recommendations, in *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.*, Vol. 1 (IEEE, 2004) pp. 398–402.

[3] R. S. Sutton, A. G. Barto, and R. J. Williams, Reinforcement learning is direct adaptive optimal control, IEEE Control Systems Magazine **12**, 19 (1992).

[4] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, A general reinforcement learning algorithm that masters chess, shogi, and go through self-play, Science **362**, 1140 (2018).

[5] D. Dong, C. Chen, H. Li, and T.-J. Tarn, Quantum reinforcement learning, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **38**, 1207 (2008).

[6] V. Dunjko, J. M. Taylor, and H. J. Briegel, Advances in quantum reinforcement learning, in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (IEEE, 2017) pp. 282–287.

[7] L. Lamata, Basic protocols in quantum reinforcement learning with superconducting circuits, Scientific reports **7**, 1609 (2017).

[8] H. J. Briegel and G. De las Cuevas, Projective simulation for artificial intelligence, Scientific reports **2**, 400 (2012).

[9] G. D. Paparo, V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel, Quantum speedup for active learning agents, Physical Review X **4**, 031002 (2014).

[10] V. Dunjko, J. M. Taylor, and H. J. Briegel, Quantum-enhanced machine learning, Physical review letters **117**, 130501 (2016).

[11] V. Dunjko, N. Friis, and H. J. Briegel, Quantum-enhanced deliberation of learning agents using trapped ions, New Journal of Physics **17**, 023006 (2015).

[12] F. Neukart, D. Von Dollen, C. Seidel, and G. Compostella, Quantum-enhanced reinforcement learning for finite-episode games with discrete state spaces, Frontiers in Physics **5**, 71 (2018).

[13] C. H. Papadimitriou and J. N. Tsitsiklis, The complexity of markov decision processes, Mathematics of operations research **12**, 441 (1987).

[14] R. Greenlaw, H. J. Hoover, W. L. Ruzzo, *et al.*, *Limits to parallel computation: P-completeness theory* (Oxford

University Press on Demand, 1995).

[15] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv preprint arXiv:1411.4028 (2014).

[16] V. S. Denchev, S. Boixo, S. V. Isakov, N. Ding, R. Babbush, V. Smelyanskiy, J. Martinis, and H. Neven, What is the computational value of finite-range tunneling?, Physical Review X **6**, 031015 (2016).

[17] F. Barahona, On the computational complexity of ising spin glass models, Journal of Physics A: Mathematical and General **15**, 3241 (1982).

[18] I. G. Rosenberg, Reduction of bivalent maximization to the quadratic case, Cahiers du Centre d'etudes de Recherche Operationnelle **17**, 71 (1975).

[19] I. Zintchenko, M. B. Hastings, and M. Troyer, From local to global ground states in ising spin glasses, Physical Review B **91**, 024201 (2015).

[20] B. A. Cipra, The ising model is np-complete, SIAM News **33**, 1 (2000).

[21] E. Farhi, D. Gosset, I. Hen, A. Sandvik, P. Shor, A. Young, and F. Zamponi, Performance of the quantum adiabatic algorithm on random instances of two optimization problems on regular hypergraphs, Physical Review A **86**, 052334 (2012).

[22] P. Patil, S. Kourtis, C. Chamon, E. R. Mucciolo, and A. E. Ruckenstein, Obstacles to quantum annealing in a planar embedding of xorsat, Physical Review B **100**, 054435 (2019).

[23] V. Bapst, L. Foini, F. Krzakala, G. Semerjian, and F. Zamponi, The quantum adiabatic algorithm applied to random optimization problems: The quantum spin glass perspective, Physics Reports **523**, 127 (2013).

[24] R. Yates, Fixed-point arithmetic: An introduction, Digital Signal Labs **81**, 198 (2009).

[25] S. E. Dreyfus, *Dynamic programming and the calculus of variations*, Tech. Rep. (RAND CORP SANTA MONICA CA, 1965).

[26] H. J. Kappen, Path integrals and symmetry breaking for optimal control theory, Journal of statistical mechanics: theory and experiment **2005**, P11011 (2005).

[27] E. Theodorou, J. Buchli, and S. Schaal, A generalized path integral control approach to reinforcement learning, journal of machine learning research **11**, 3137 (2010).

[28] E. Dynkin, Markov processes as a tool in field theory, Journal of Functional Analysis **50**, 167 (1983).

[29] M. E. Peskin, *An introduction to quantum field theory* (CRC press, 2018).

[30] S. P. Pedersen, K. S. Christensen, and N. T. Zinner, Native three-body interaction in superconducting circuits, Physical Review Research **1**, 033123 (2019).

[31] E. Boros and P. L. Hammer, Pseudo-boolean optimization, Discrete applied mathematics **123**, 155 (2002).

[32] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, Openai gym, arXiv preprint arXiv:1606.01540 (2016).

[33] N. Dattani, Quadratization in discrete optimization and quantum mechanics, arXiv preprint arXiv:1901.04405 (2019).

[34] A. Fix, A. Gruber, E. Boros, and R. Zabih, A graph cut algorithm for higher-order markov random fields, in *2011 International Conference on Computer Vision* (IEEE, 2011) pp. 1020–1027.

[35] E. B. Jones, E. Kapit, C.-Y. Chang, D. Biagioni, D. Vaidhynathan, P. Graf, and W. Jones, On the computational viability of quantum optimization for pmu placement, arXiv preprint arXiv:2001.04489 (2020).

[36] A. Lucas, Hard combinatorial problems and minor embeddings on lattice graphs, Quantum Information Processing **18**, 203 (2019).

[37] K. Boothby, P. Bunyk, J. Raymond, and A. Roy, *Next-generation topology of D-wave quantum processors*, Tech. Rep. (Technical report, 2019).

[38] M. A. Nielsen and I. Chuang, Quantum computation and quantum information (2002).

[39] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, Physical review A **52**, 3457 (1995).

[40] P. Kumar, Direct implementation of an n-qubit controlled-unitary gate in a single step, Quantum information processing **12**, 1201 (2013).

[41] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse ising model, Physical Review E **58**, 5355 (1998).

[42] T. Albash and D. A. Lidar, Demonstration of a scaling advantage for a quantum annealer over simulated annealing, Physical Review X **8**, 031016 (2018).

[43] S. V. Isakov, I. N. Zintchenko, T. F. Rønnow, and M. Troyer, Optimised simulated annealing for ising spin glasses, Computer Physics Communications **192**, 265 (2015).

[44] Source code for neal.sampler, `https://docs.ocean.dwavesys.com/projects/neal/en/latest/_modules/neal/sampler.html#SimulatedAnnealingSampler.sample`, accessed: 2020-03-21.