K-spin Hamiltonian for quantum-resolvable Markov decision processes

Eric B. Jones1,2 · Peter Graf1 · Eliot Kapit2 · Wesley Jones1

Received: 14 April 2020 / Accepted: 7 October 2020 / Published online: 30 October 2020
© Springer Nature Switzerland AG 2020

Abstract
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 arguing that the variational minimization of our Hamiltonian is approximately equivalent to the Bellman optimality condition for a prevalent class of environments 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.

Keywords K-spin hamiltonian · Markov decision process · Quantum optimization · Quantum annealing · Simulated annealing · Quantum approximate optimization algorithm

1 Introduction
Alongside supervised and unsupervised learning, reinforcement learning (RL) constitutes one of the main pillars of modern attempts to create artificial intelligence (Sutton and Barto 2018). 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 initially unknown optimal policy within an environment. The agent learns the unknown policy by attempting to maximize some notion of expected future rewards derived from the 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 (Golovin and Rahm 2004; Sutton et al. 1992; Silver et al. 2018).

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 (Dong et al. 2008; Dunjko et al. 2017; Lamata 2017; Briegel and...
De las Cuevas 2012; Paparo et al. 2014; Dunjko et al. 2016; Dunjko et al. 2015; Neukart et al. 2018). 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 (Papadimitriou and Tsitsiklis 1987). This means roughly that while MDPs are in principle efficiently solvable in polynomial time, they are not categorically efficiently parallelizable (Greenlaw et al. 1995). 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 (Sutton and Barto 2018).

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 as K-spin Hamiltonians of the form (Farhi et al. 2014; Denchev et al. 2016)

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

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 (Barahona 1982; Rosenberg 1975). 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 (Papadimitriou and Tsitsiklis 1987). 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 (Zintchenko et al. 2015; Barry 2000).

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 (Farhi et al. 2012; Patil et al. 2019; Bapst et al. 2013). 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 \( P = NC \)).

## 2 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 (2018). 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 \( P_{ss'rt}^{a} \), 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 \( R_{ss'rt}^{a} \), 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.

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 $\pi$: the value function

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

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)

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

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

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

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 $R_{sa}$ as possible and as few negative-valued ones over some effective time horizon dictated by the value of $\gamma$. 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 (see Supplementary Material). 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 $R^1_{1,5} = 1$ (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 $R^a_{s,s+1} = -10$ 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 (Papadimitriou and Tsitsiklis 1987). 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, $\pi$, 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 Section 3 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 (Yates 2009).

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 (Sutton and Barto 2018).

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 $\pi$. An optimal policy is one for which $Q_{sa}[\pi]$ is
maximized ∀ s, a. One of the central results of the dynamic programming treatment of MDPs is that the action-value function obeys a Bellman equation

\[ Q_{sa}[\pi] = \sum_{s'} P_{sa}^a R_{s's}^a + \sum_{s',a'} \left( \gamma \sum_{s''} P_{s's}^a P_{s'a'}^{a'} R_{s'a'}^{a''} \right) + \ldots . \tag{5} \]

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.

### 3 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. The result is

\[
Q_{sa}[\pi] = \sum_{s',a} P_{sa}^a R_{s's}^a + \sum_{s',a'} \left( \gamma \sum_{s''} P_{s's}^a P_{s'a'}^{a'} R_{s'a'}^{a''} \right) + \sum_{s',a',a''} \pi'_{s'a'} \pi''_{a''} \left( \gamma^2 \sum_{s'''} P_{s's'}^a P_{s'a''}^{a''} P_{s''a''}^{a'''} R_{s''a''}^{a'''} \right) + \ldots \tag{6} \]

\( Q \) is now a functional of the policy. A stationary point of a functional occurs whenever its variation with respect to its argument vanishes (Stuart 1965). Therefore finding the optimal policy that maximizes the Q-function everywhere is equivalent to the variational conditions

\[
\delta Q_{sa}[\pi; \eta] = 0 \quad \delta^2 Q_{sa}[\pi; \eta] < 0 \tag{7} \]

for an arbitrary variation of \( Q \) with respect to the policy, \( \pi \rightarrow \pi + \epsilon \eta \), where \( \eta \) is an arbitrary element of the policy function space and \( \epsilon \) is a very small real number. From here it is straightforward to define a Hamiltonian functional as

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

Recall here that \( \pi_{sa} \in [0,1] \) and therefore has a simple relationship to single-qubit spin operators via \( \pi_{sa} \rightarrow (\hat{1}_{sa} - \hat{Z}_{sa})/2 \). So, Eq. 8 is equivalent to a K-spin Hamiltonian where \( \gamma^K \) 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 \( \pi \) is equivalent to maximizing \( Q_{sa} \) for each \( s \) and \( a \). For this purpose, consider an arbitrary variational derivative of the Hamiltonian with respect to the policy. Finding a stationary point of the Hamiltonian, with Eq. 6 substituted into Eq. 8, corresponds to

\[
\frac{\delta H[\pi]}{\delta \pi_{sa}} = 0 = -\sum_{s,a} \frac{\delta Q_{sa}[\pi]}{\delta \pi_{sa}} . \tag{9} \]

But since the variation is arbitrary and since each Q function only depends on the state-action coordinate \((s, a)\) through its local transition and reward environment, each term in the sum should approximately vanish leading to the first condition in Eq. 7 for all \( s, a, \tilde{s}, \tilde{a} \) (for a more detailed discussion, please see Supplemental Material).

There is an interesting analogy to classical field theory in the present discussion shown in Table 1. Each state-action pair \( \mu = (s, a) \) plays the role of a spacetime coordinate over which a field \( \pi_{\mu} \) 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 (Kappen 2005; Theodorou et al. 2010; Dynkin 1983). 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.

### Table 1 Analogy between classical field theory (CFT) and the Markov decision process (MDP)

| Object            | CFT                | MDP                |
|-------------------|--------------------|--------------------|
| Coordinate        | \( x = (x^0, x^1, x^2, x^3) \) | \( \mu = (s, a) \) |
| Field             | \( \Phi(x) \)     | \( \pi_{\mu} \)   |
| Density           | \( \mathcal{L}(\Phi(x), x) \) | \( Q_{\mu}[\pi] \) |
| Functional        | \( S[\Phi] = \int dx \mathcal{L}(\Phi(x), x) \) | \( H[\pi] = \sum_{\mu} Q_{\mu}[\pi] \) |
| Stationarity      | \( \delta S = 0 \) | \( \delta H = 0 \) |
| E-L Eqn.          | \( \delta \mathcal{L}/\delta \Phi = 0 \) | \( \delta Q/\delta \pi = 0 \) |

Springer
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, by substituting Eq. 6, which is a formally exact expansion of the Q-function in policy variables, into the Hamiltonian definition in Eq. 8, whose minimization was proved equivalent to the Bellman optimality condition in Eq. 9, we obtain the first line in the following Hamiltonian, the full expression of which (Eq. 10) we may use to solve an MPD for an optimal policy

\[ H[\pi] = -\sum_{k=0}^{\infty} \sum_{\mu_1,..,\mu_k=1}^{S \times A} J_{\mu_1,..,\mu_k} \pi_{\mu_1} \cdots \pi_{\mu_k} \]

\[ + M \sum_{s_1}^{|S|} \left( \sum_{a_1=1}^{|A|} \pi_{s_1,a_1} - 1 \right)^2 \]

where we have made the identifications \( s(a) \rightarrow s_0(a_0), s'(a') \rightarrow s_1(a_1), \text{etc.} \) and

\[ J_{\mu_1,..,\mu_k} = J_{(s_{\mu_1},a_{\mu_1})..(s_{\mu_k},a_{\mu_k})} \]

\[ = \gamma^k \sum_{s_{00},a_{00},s_{k+1},a_{k+1}} P_{s_{00}s_{1}a_{1}}^{a_{0}a_{1}} \cdots P_{s_{k}s_{k+1}a_{k+1}}^{a_{k}a_{k+1}} \]

Equation 10 includes a penalty function with strength \( M \) in the second line due to the fact that the policy \( \pi_{s_{\mu}} \) is generally a probability distribution conditioned on the state variable \( s \) and 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 \( K \rightarrow \infty \). 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 (Pedersen et al. 2019; Boros and Hammer 2002). 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 \( \gamma^k \), 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 Section 4. Figure 1b shows the optimal policy found by solving for the ground state of \( H \) for the AVR-dirt scenario described in Section 2 on the D-Wave 2000Q processor and validated by simulated annealing and classical Q-Learning run in an OpenAI Gym environment (Brockman et al. 2016). 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 Section 5. 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 Section 4, but we remark here that we set the penalty strength required for order reduction to \( M_{OR} = 5 \). Figure 2 shows how premature truncation adversely affects solution 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 Section 4 before assessing the scaling in time complexity of the K-spin MDP in Section 5.

4 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 \( K \). Figure 3 shows the lowest truncation order \( (K) \) at which
Effects of prematurely truncating the K-spin Hamiltonian. \textbf{a} At $\gamma = 0.99$ and $K = 2$ for a 6-tile state space and asymmetric terminal rewards specified in Section 2, the optimal policy is erroneously symmetric. Note however, that this policy is the true ground state for $\gamma = 0.8$ and $K = 2$. \textbf{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.

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 ($\gamma$) for the environment described in Section 2. The truncation order scales roughly as $K \sim |S|/2$. 

Quantum Machine Intelligence (2020) 2:12 Page 6 of 11
In its current form, i.e., before the substitution to qubit operators $\pi_{sa} \rightarrow (\hat{I}_{sa} - \hat{Z}_{sa})/2$, 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 (Rosenberg 1975). There are a variety of approaches to polynomial order reduction, which typically require the introduction of ancillary variables (Dattani 2019). 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

$$xy = (\neq)z \iff xy - 2xz - 2yz + 3z = (>0)$$

(12)

hold (Rosenberg 1975; Boros and Hammer 2002). Therefore, one may rewrite a product of two variables $xy$ as a single variable $z$ while at the same time introducing the penalty function

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

(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 (Fix et al. 2011). Figure 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 $\gamma$ and independent of the hardware platform used. The dependence of $|V|$ on $\gamma$ 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.

Figure 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 (Jones et al. 2020). 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 (Lucas 2019). Figure 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 (see Supplementary Material regarding parameter choices in Fig. 4).

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 (Boothby et al. 2019). By contrast, gate-model quantum computers can decompose higher-order interactions via gate compilation into products of single and two-qubit gates (Nielsen and Chuang 2002). 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 $\beta$, we have

$$e^{-i\beta \hat{H}} = \prod_{k=0}^{K} \prod_{\mu_1,\ldots,\mu_k=1}^{|S \times A|} \left[ \hat{I}_{\mu_1} \otimes \cdots \otimes \hat{I}_{\mu_k} + \left( e^{i\beta J_{\mu_1,\ldots,\mu_k}} - 1 \right) \hat{A}_{\mu_1} \otimes \cdots \otimes \hat{A}_{\mu_k} \right].$$

Equation 14 is equivalent to a product of $(k - 1)$-qubit controlled-PHASE ($e^{i\beta J_{\mu_1,\ldots,\mu_k}}$) gates. Without introducing any additional ancillary qubits, a general $(N - 1)$-qubit controlled-unitary may be decomposed with $O(N^2)$ single- and two-qubit gates (Barenco et al. 1995). 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 $O(pK^2|S \times A|^k)$. 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 (Kumar 2013). 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 (Kadowaki and Nishimori 1998).

5 Time complexity of K-spin MDP

5.1 Quantum annealing

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

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

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 (Denchev et al. 2016; Albash and Lidar 2018). 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 $M \approx M_{OR} \geq 2$ 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.

Figure 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 $\gamma$. As such, Fig. 5b shows $TTS(t^*_f)$ as a function of |S| for all values of $\gamma$. 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
Section 4, $TTS(t^*_f)$ 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 $\gamma$ is a direct result of the dependence of $|V|$ and therefore $|Q|$ on $\gamma$. 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 (Farhi et al. 2012; Patil et al. 2019; Bapst et al. 2013).

5.2 Simulated annealing

An analog to Eq. 15 for simulated annealing is

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

where for large $n_s \cdot N$, $t_a = n_s \cdot N/f$, but in general $t_a = t_o(n_s)$ (Isakov et al. 2015). 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 $\gamma = 0.6$, $n^*_s \approx 7$ 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 $\gamma$. 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 D-Wave Systems Inc. (2018)).
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. Given that it has recently been shown that optimal quantum control problems can possess glassy energy landscapes (Day et al. 2019), it appears likely that the exponential time complexity to solve an MDP as a spin Hamiltonian may result from frustrated long-range and many-body policy interactions introduced in the definition of Eq. 10. In the type of environment depicted in Fig. 1, for example, frustration could occur among two- and three-body terms deep in the interior of a long hallway where all reward functions are $-1$. For two-body terms, the resulting couplings are antiferromagnetic whereas for three-body terms, the couplings are ferromagnetic, inducing frustration. This is perhaps unsurprising since the $k$th order of Eq. 10 has the same form as the Random Energy Model commonly used to model spin glasses in the limit where $k \to \infty$ (Derrida 1980).

## 6 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 research used Ising@Los Alamos National Laboratory’s D-Wave quantum annealer. 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.

## Funding

This work was supported by the Laboratory Directed Research and Development (LDRD) Program at NREL. This material is based in part upon work supported by the National Science Foundation under Grant No. PHY-1653820.

## Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## Code availability

The code that supports the findings of this study are available from the corresponding author upon reasonable request.

## Compliance with ethical standards

## Conflict of interest

The authors declare that they have no competing interests.

## Disclaimer

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.

## References

Albash T, Lidar DA (2018) Demonstration of a scaling advantage for a quantum annealer over simulated annealing. Phys Rev X 8(3):031016

Bapst V, Fioni L, Krzakala F, Semerjian G, Zamponi F (2013) The quantum adiabatic algorithm applied to random optimization problems: the quantum spin glass perspective. Phys Rep 523(3):127–205

Barabona F (1982) On the computational complexity of ising spin glass models. J Phys A Math Gen 15(10):3241

Barenco A, Bennett CH, Cleve R, DiVincenzo DP, Margolus N, Shor P, Sleator T, Smolin JA, Weinfurter H (1995) Elementary gates for quantum computation. Phys Rev A 52(5):3457

Barry AC (2000) The ising model is np-complete. SIAM News 33(6):1–3

Boothby K, Bunyk P, Raymond J, Roy A (2019) Next-generation topology of d-wave quantum processors. Technical report, Technical report

Boros E, Hammer PL (2002) Pseudo-boolean optimization. Discret Appl Math 123(1-3):155–225

Briegel HJ, De las Cuevas G (2012) Projective simulation for artificial intelligence. Sci Rep 2:400

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

Dattani N (2019) Quadratization in discrete optimization and quantum mechanics. arXiv:1901.04405

Day AGR, Bukov M, Weinberg P, Mehta P, Dries S (2019) Glassy phase of optimal quantum control. Phys Rev Lett 122(2):020601

Denchev VS, Boixo S, Isakov SV, Ding N, Babbush R, Smelyanskiy V, Martinis J, Neven H (2016) What is the computational value of finite-range tunneling? Phys Rev X 6(3):031015

Derrida B (1980) Random-energy model: limit of a family of disordered models. Phys Rev Lett 45(2):79

Dong D, Chen C, Li H, Tarn TJ (2008) Quantum reinforcement learning. IEEE Trans Sys Man Cybern Part B Cybern 38(5):1207–1220

Dunjko V, Friis N, Hans JB (2015) Quantum-enhanced deliberation of learning agents using trapped ions. New J Phys 17(2):023006

Dunjko V, Taylor JM, Hans JB (2016) Quantum-enhanced machine learning. Phys Rev Lett 117(13):130501

Dunjko V, Taylor JM, Hans JB (2017) Advances in quantum reinforcement learning. In: IEEE international conference on systems, man, and cybernetics (SMC), pp 282–287. IEEE
Dynkin EB (1983) Markov processes as a tool in field theory. J Funct Anal 50(2):167–187
Farhi E, Gosset D, Hen I, Sandvik AW, Shor P, Young AP, Francesco Z (2012) Performance of the quantum adiabatic algorithm on random instances of two optimization problems on regular hypergraphs. Phys Rev A 86(5):052334
Farhi E, Goldstone J, Gutmann S (2014) A quantum approximate optimization algorithm. arXiv:1411.4028
Fix A, Gruber A, Boros E, Ramin Z (2011) A graph cut algorithm for higher-order markov random fields. In: International conference on computer vision, pp. 1020–1027. IEEE, p 2011
Golovin N, Rahm E (2004) Reinforcement learning architecture for web recommendations. In: International conference on information technology: coding and computing, 2004. Proceedings. ITCC 2004. vol 1, pp 398–402. IEEE
Greenlaw R, Hoover JH, Ruzzo WL et al (1995) Limits to parallel computation: p-completeness theory. Oxford University Press on Demand, Oxford
Isakov SV, Zintchenko IN, Rønnow TF, Troyer M (2015) Optimised simulated annealing for ising spin glasses. Comput Phys Commun 192:265–271
Jones EB, Kapit E, Chang CY, Biagioni D, Vaidhynathan D, Graf P, Jones W (2020) On the computational viability of quantum optimization for pmu placement. arXiv:2001.04489
Kadowaki T, Nishimori H (1998) Quantum annealing in the transverse ising model. Phys Rev E 58(5):5355
Kappen HJ (2005) Path integrals and symmetry breaking for optimal control theory. J Stat Mech Theory Exp 2005(11):P11011
Kumar P (2013) Direct implementation of an n-qubit controlled-unitary gate in a single step. Quantum Inf Process 12(2):1201–1223
Lamata L (2017) Basic protocols in quantum reinforcement learning with superconducting circuits. Sci Rep 7(1):1609
Lucas A (2019) Hard combinatorial problems and minor embeddings on lattice graphs. Quantum Inf Process 18(7):203
Neukart F, Dollen DV, Seidel C, Compostella G (2018) Quantum-enhanced reinforcement learning for finite-episode games with discrete state spaces. Front Phys 5:71
Nielsen MA, Chuang I (2002) Quantum computation and quantum information
Papadimitriou CH, Tsitsiklis JN (1987) The complexity of markov decision processes. Math Oper Res 12(3):441–450
Paparo GD, Dunjko V, Makmal A, Martin-Delgado MA, Hans JB (2014) Quantum speedup for active learning agents. Phys Rev X 4(3):031002
Patil P, Kourtis S, Chamon C, Mucciolo ER, Andrei ER (2019) Obstacles to quantum annealing in a planar embedding of xorsat. Phys Rev B 100(5):054435
Pedersen SP, Christensen KS, Nikolaj TZ (2019) Native three-body interaction in superconducting circuits. Phys Rev Res 1(3):033123
Peskin ME (2018) An introduction to quantum field theory. CRC Press, Boca Raton
Rosenberg IG (1975) Reduction of bivalent maximization to the quadratic case. Cahiers du Centre d’etudes de Recherche Operationnelle 17:71–74
Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T, et al. (2018) A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. Science 362(6419):1140–1144
Stuart ED (1965) Dynamic programming and the calculus of variations. Technical report, RAND CORP SANTA MONICA CA
Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT Press, Cambridge
Sutton RS, Barto AG, Williams RJ (1992) Reinforcement learning is direct adaptive optimal control. IEEE Control Syst Mag 12(2):19–22
D-Wave Systems Inc. (2018) Source code for neal.sampler. https://docs.ocean.dwavesys.com/projects/neal/en/latest/_modules/neal/sampler.html#SimulatedAnnealingSampler.sample. Accessed: 2020-03-21
Theodorou E, Buchli J, Schaal S (2010) A generalized path integral control approach to reinforcement learning. J Mach Learn Res 11(Nov):3137–3181
Yates R (2009) Fixed-point arithmetic: an introduction. Digital Signal Labs 81(83):198
Zintchenko I, Hastings MB, Troyer M (2015) From local to global ground states in ising spin glasses. Phys Rev B 91(2):024201

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.