BALANCING EXPLORATION AND EXPLOITATION IN REINFORCEMENT LEARNING USING A VALUE OF INFORMATION CRITERION

Isaac J. Sledge[1] *José C. Príncipe*[1,2]

[1] Department of Electrical and Computer Engineering, University of Florida
[2] Department of Biomedical Engineering, University of Florida

ABSTRACT

In this paper, we consider an information-theoretic approach for addressing the exploration-exploitation dilemma in reinforcement learning. We employ the value of information, a criterion that provides the optimal trade-off between the expected returns and a policy's degrees of freedom. As the degrees of freedom are reduced, an agent will exploit more than explore. As the policy degrees of freedom increase, an agent will explore more than exploit.

We provide an efficient computational procedure for constructing policies using the value of information. The performance is demonstrated on a standard reinforcement learning benchmark problem.

***Index Terms*—**Reinforcement learning, exploration, exploitation

1. INTRODUCTION

The problem of optimal decision making under uncertainty is crucial for intelligent agents. Reinforcement learning [1, 2] addresses this problem by proposing that agents should maximize an expected long-term return provided by the environment. This provides the basis for trial-and-error-based learning of an action-selection policy. Such a policy is a mapping between states in the environment and suitable actions that permit an agent to achieve its objectives.

There are a variety of reinforcement learning techniques that can be applied for uncovering action-selection policies. Popular examples include temporal-difference learning proposed by Sutton [3], Q-learning pioneered by Watkins [4], and SARSA by Wiering and Schmidhuber [5].

For these methods, it is helpful if an agent investigates its environment and simultaneously leverages some or all of its past experiences. Phrased another way, an agent has to both explore and exploit so that the returns for a given policy do not stagnate during learning [6].

The trade-off between agent exploration and exploitation has been previously considered. An overview of classical exploration-exploitation schemes is given by Kaelbling, Littman, and Moore [7]. The former conduct principled explorations of the environment, but mainly either for special classes of problems or problems with limited state-action state complexity [8–10]. As the number of agent states grows, many of these formally justified approaches become computationally intractable. The latter type of procedures are often heuristic in nature [11–14], though some are more principled [15–17]. Since some of these methods lack strong theoretical underpinnings, it can be difficult to provide consistent guarantees about their expected policy performance. It is possible, nevertheless, for these ad hoc schemes to perform reasonably well in many situations and even on complicated state-action pairs.

Despite these efforts, it is difficult to quantify the trade-off between exploration and exploitation for arbitrary applications. Ideally, we would like to bound the expected returns for a given amount of exploration. We would also like to generate (near-)optimal policies that exist within those bounds.

In this paper, we develop an information-theoretic means for balancing exploration and exploitation that addresses these desires. Our approach is based upon Stratonovich's value of information criterion [18–20]. The value of information is a two-term criterion that describes the largest possible reduction of average costs associated with actions that carry a certain amount of information about the current state. To define the cost reduction amount, the criterion first measures the expected costs for actions that have no information about the state. It then offsets these costs using a term that measures the average penalties when the state-action information is bounded above by a prescribed amount.

When applied to reinforcement learning of Markov decision processes, this criterion provides bounds for the best possible returns that can be obtained for a prescribed exploration quota. Here, the exploration quota is dictated by a policy's degrees of freedom, which can be viewed as a measure of policy complexity. Policies with many degrees of freedom will inherently be exploration intensive. Conversely, policies with few degrees of freedom will favor less exploration and early exploitation during the learning process. We capture this exploration-exploitation trade-off with a user-tunable parameter, which measures the average number of bits per action.

We provide a novel, grouped-coordinate descent approach for finding locally optimal solutions of the value of information criterion. We show that the value of information framework can also be paired with tabular, model-based Q-learning to produce policies. For this class of reinforcement learning, the value of information gives rise to Boltzmann-style random exploration. Unlike traditional Boltzmann-based exploration [1, 21, 22], our version includes an extra term that promotes a range of exploration-exploitation strategies while attempting to maximize the expected policy returns. This term arises as a byproduct of the value of information optimization.

2. METHODOLOGY

2.1. Markov Decision Process Policies

Reinforcement learning for many problems is posed using Markov decision processes, which are models for a certain class of stochastic agent control problems. At each time step, the agent makes an observation of the environment state $s_t \in \mathcal{S}$ at time t, which is assumed to be a random variable with some known distribution. The agent interacts with the environment by taking an action $a_t \in \mathcal{A}$, from some decision space, at time t while in state s_t. This leads to a new state s_{t+1}. The new state is a random variable, and a probability is assigned to the transition. The choice of the next state is Markovian.

Associated with each state transition is a cost that maps from the product state and action spaces to the positive reals. In many applications, the costs are implicitly defined. Since state transitions are random, the rewards are too.

The way that an agent behaves in the environment is referred to as a policy, which is a mapping $\pi_a(s) : \mathcal{S} \to \mathcal{A}$. Policies can have a

This work was funded by ONR grant N00014-15-1-2103. The first author was additionally funded by a University of Florida Research Fellowship and a Robert C. Pittman Research Fellowship.

probabilistic interpretation, which implies that they characterize the probability distribution over actions given the state. Each policy has an associated action-value function $q(s,\pi_a(s)) : \mathcal{S}\times\mathcal{A}\rightarrow\mathcal{R}$. The goal of a reinforcement learning agent is to find a policy that optimizes the value function for all state-action pairs: $\arg\inf_{\pi_a}\mathbb{E}[q(s,\pi_a(s))]$. Here, the value function is defined in terms of the discounted future costs associated with a particular sequence of actions.

The problem of finding the best policy π_a^* can be explicitly written as follows

$$\inf_{\pi_{a_t}(s):\mathcal{S}\rightarrow\mathcal{A}}\left(\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\text{pr}(s)\text{pr}(a|s)q(s,\pi_a(s))\right). \tag{1}$$

Such a policy is given by the Bayes risk $\pi_a^* = \mathbb{E}[\inf_{a\in\mathcal{A}}\mathbb{E}[q(s,a)]]$. This policy may be stochastic or deterministic. If the policy is stochastic, then it is a mapping from the states to an action distribution.

2.2. Policy Exploration with Value of Information

There are many ways that we can envision promoting either exploration or exploitation for Markov decision processes policies. The option that we consider here is to bound the number of actions that an agent can perform. If the agent can carry out a large number of actions, then the chance that it can explore is high. If the agent can only take a small number of actions, then the chance that it will explore will be low. In either case, we are artificially constraining the policy search space by changing the policy's degrees of freedom.

Additionally, we would like to uncover policies that provide the largest possible reduction in the action-value function. To simultaneously handle both desires, we consider an optimization problem that involves two terms. Both terms are modifications of (1):

First Term: No-Information Returns. The first term captures the possible return for a policy in which no information about the states can be inferred from the action. This is used to establish the baseline agent performance. If the actions are not informative, then the optimal decision is based solely on the state random variable distribution. If, however, the actions are informative, then the returns for the simplest policy will be offset by a second term.

Second Term: Informative Returns. This second term encodes the returns associated with policies whose maximum action-state complexity is specified a priori. It is based on the expected return using an action-value function. The policy complexity is determined by a Shannon information constraint, which is bounded above by a constraint parameter. The magnitude of the constraint parameter dictates the degrees of freedom available for the resulting policy: the lower its value, the fewer the degrees of freedom for a policy. Setting this variable too low will compromise the anticipated policy returns.

The use of these two terms is motivated by the Shannon theory of value of information, as proposed by Stratonovich [18–20]. When reformulated for Markov decision process policies, the value of information criterion is given by (2) and (3). Note that the constraint term for the conditional probabilities (3) defines the amount of information that an action carries about the state under a given policy. By constraining the information, we are putting a limit on the average bit cost of the policy, which influences a policy's degrees of freedom.

The user-selectable constraint parameter r in (3) is used to define an upper bound of the average bit cost per action. The parameter also has an impact on the amount of exploration. As r decreases toward zero, the available search space is explored more coarsely, and the agent exploits more often than it explores. When r is zero, exploitation disappears, as the policy is a random walk. The agent may therefore take an inordinate amount of time to complete an objective. When the parameter is set near or above the state entropy, the policy complexity is equivalent to that of standard reinforcement learning approaches. The agent attempts to just maximize the possible returns in this case.

The combination of both terms in (2) describes the possible performance gains for varying levels of exploration when compared to the no-information baseline. In fact, it can provide optimal return bounds, in a Shannon sense, for a policy of a prescribed complexity. These bounds are analogous to rate-distortion functions [23].

2.2.1. Optimizing the Value of Information

The preceding value of information optimization problem, unfortunately, has some practical difficulties. In particular, investigators must have knowledge of the minimum expected cost associated with the globally optimal policy. Even just estimating this cost can be troublesome.

To address this issue, we equivalently re-write the criterion defined by (2) and (3) as (4) and (5), where r in (5) now defines a bound on the expected return. Since these problems are equivalent, however, we will stay with the original interpretation of r. Notice that the probability constraints in (5) are solely based on a current estimate of the global best policy. Such a policy can be iteratively estimated using reinforcement learning.

We now introduce Lagrange multipliers into (4) to handle the constraint in (5): γ, which handles the action-value function constraint and $\beta(s)$, which ensures that the conditional probabilities have unit sum. This allows us to convert the value of information criterion into an unconstrained problem. The solution to the unconstrained problem can be obtained by finding its gradient and setting it to zero:

$$\text{pr}(a|s) = \text{pr}(a)e^{q(s,a)/\gamma}\Big/\sum_{a\in\mathcal{A}}\text{pr}(a)e^{q(s,a)/\gamma}\ . \tag{6}$$

$$\inf_{a\in\mathcal{A}}\mathbb{E}[q(s,a)] - \inf_{\text{pr}(a|s)}\mathbb{E}[\inf_{a\in\mathcal{A}}\mathbb{E}[q(s,a)]] = \inf_{a\in\mathcal{A}}\left(\sum_{s\in\mathcal{S}}\text{pr}(s)q(s,a)\right) - \inf_{\text{pr}(a|s)}\left(\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\text{pr}(s)\text{pr}(a|s)q(s,\pi_a^*(s))\right) \tag{2}$$

$$\text{such that pr}(a|s) : \left(\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\text{pr}(s)\text{pr}(a|s)\log(\text{pr}(a|s)) - \sum_{s\in\mathcal{S}}\sum_{o\in\mathcal{A}}\text{pr}(s,a)\log(\text{pr}(a))\right)\leq r,\ r>0 \tag{3}$$

$$\inf_{a\in\mathcal{A}}\left(\sum_{s\in\mathcal{S}}\text{pr}(s)q(s,a)\right) - \inf_{\text{pr}(a|s)}\left(\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\text{pr}(s)\text{pr}(a|s)\log(\text{pr}(a|s)) - \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\text{pr}(s,a)\log(\text{pr}(a))\right) \tag{4}$$

$$\text{such that pr}(a|s) : \left(\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\text{pr}(s)\text{pr}(a|s)q(s,\pi_a(s))\right)\leq r,\ r>0 \tag{5}$$

Algorithm 1: Value-of-Information-based Policy Learning

1 Choose a non-negative value for γ.
2 Initialize the conditional probabilities $\text{pr}(a|s)$, $\forall s \in \mathcal{S}$, $a \in \mathcal{A}$.
3 Initialize the action-value function $q(a, s)$, $\forall s \in \mathcal{S}$, $a \in \mathcal{A}$.
4 **for** $t = 0, 1, 2, \ldots$ **do**
5 Initialize the conditional probabilities $\text{pr}_t^{(0)}(a|s)$, $\forall s \in \mathcal{S}$, $a \in \mathcal{A}$.
6 Update the state visitation probabilities $\text{pr}_t(s)$, $\forall s \in \mathcal{S}$.
7 **for** $k = 0, 1, 2, \ldots$ **do**
8 Update $\text{pr}_t^{(k)}(a)$ using (7) $\forall a \in \mathcal{A}$.
9 Update $\text{pr}_t^{(k+1)}(a|s)$ using (6) $\forall s \in \mathcal{S},\ a \in \mathcal{A}$.
10 Update the policy using $\text{pr}_t^{(k+1)}(a|s)$.
11 Choose an action $a_t \in \mathcal{A}$ from the policy.
12 Transition from $s_t \in \mathcal{S}$ to the next state $s_{t+1} \in \mathcal{S}$.
13 Update the action-value function estimates q.

for all $\forall s \in \mathcal{S},\ a \in \mathcal{A}$. Equation (6) must be solved together with

$$\text{pr}(a) = \sum_{s \in \mathcal{S}} \text{pr}(a|s)\text{pr}(s), \tag{7}$$

for all $\forall a \in \mathcal{A}$, in an alternating manner. That is, an initial guess is supplied for $\text{pr}(a|s)$, which is kept fixed while updating $\text{pr}(a)$ using (7). The probabilities $\text{pr}(a)$ are then fixed while revising $\text{pr}(a|s)$ using (6). This process continues until the difference between sequential updates falls below some threshold.

It is important to notice that the update in (6) is related to Boltzmann-based exploration [1]. For this type of exploration, the preference for one action over another is dictated by a Boltzmann distribution: actions with better returns are associated an increased preference chance. The final action is chosen at random using those preferences. It is well known that this type of weighted random exploration often outperforms purely random exploration strategies, since it accounts for the influence of each state-action pair on the returns.

Exploration with the value of information criterion will behave in an analogous way. However, our approach explicitly discounts possible action sequences based on the complexity of the policy, which is measured by the bit cost needed to encode those action sequences. Agents may therefore use only a subset of the entire action space, which often makes the learning task easier due to a potentially decreased need for exploration. Traditional Boltzmann-based exploration would potentially consider the entire action space.

2.2.2. Finding Policies with the Value of Information

We outline, in algorithm 1, a Boltzmann-exploration-inspired algorithm for finding policies using the value of information.

There are a few implementation concerns. The first relates to the initial value of the conditional probabilities in step 5, which influences the solution to which the iteration in steps 7–10 converges. One can set these probabilities to the final result from the previous iteration of steps 7–10. This assumes, though, that the policy changes relatively smoothly across time.

For the inner loop over steps 7-10, the policy is repeatedly updated until the difference between the action probability distribution is negligible. When this occurs, the iterates are likely in the neighborhood of a locally optimal solution.

The update given in step 9 provides a trade-off between the complexity of the policy, measured by the expected number of bits per action, and the average returns. This trade-off is governed by the multiplier γ. As γ goes to zero, emphasis is placed on the agent performing actions that yield the minimum cost regardless of the policy complexity. This implies that the agent will explore more than exploit. As γ goes to infinity, the agent seeks policies that are increasingly simple. It will still, however, attempt to uncover policies with the best returns.

Choosing a value for γ will be both environment-dependent and subject to the desires of the investigator. In our future work, we will investigate an automated means of finding γ values that balance the search times and provide (near-)optimal returns.

3. SIMULATIONS

To empirically evaluate the performance of the value of information criterion for reinforcement learning, we consider the mountain car benchmark problem [25]. For this problem, an agent aims to drive a car to the top of a steep mountain. The car cannot simply accelerate forward toward the goal, though, since its engine is not powerful enough to overcome gravity. Instead, the agent must learn to drive backwards up the opposite hill, which enables the car to build enough inertia to reach the goal before its velocity sufficiently decreases.

The agent's state at time t consists of its current position p_t and its current velocity $\dot{p}_t$. It receives a reward of -1 at each time step until reaching the goal; the episode terminates when this occurs. The agent's available actions involve increasing the throttle, which can cause the vehicle to move right, reversing the throttle, which can cause the vehicle to move left, or putting the car in neutral. The following equations dictate the car's movement: $p_t = \text{bound}(p_t + \dot{p}_{t+1})$ and $\dot{p}_{t+1} = \text{bound}(\dot{p}_t + 0.001a_t - 0.0025\cos(3p_t))$. Here, $a_t \in \{-1, 0, 1\}$ is the action that the agent takes at time t. The bound function for p_t constrains the position to $p_t \in [-1.5, 0.5]$, while the bound function for $\dot{p}_{t+1}$ constrains the velocity to $\dot{p}_{t+1} \in [-0.7, 0.7]$. We uniformly discretized the two continuous variables into 200 bins each, for a total space size of 40,000 bins.

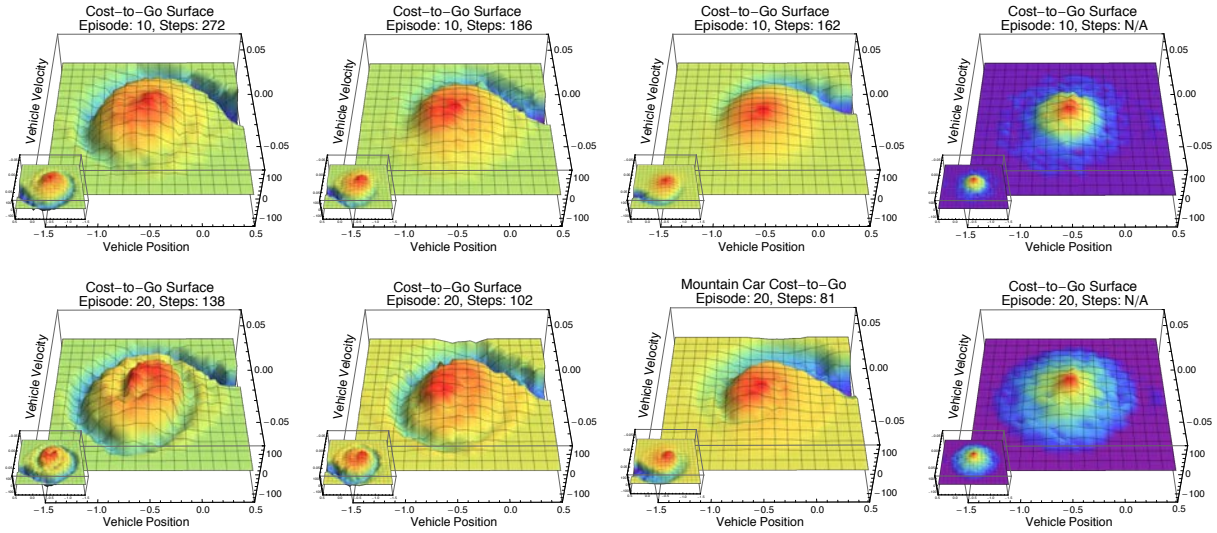
We used tabular Q-learning for updating the value function in step 13 of algorithm 1. A learning rate of 0.7 was used so that the agent weights current information more than previously acquired information. A discount factor of 0.9 were used so that the agent would seek action sequences with low long-term costs.

In each Q-learning episode, the agent begins at the basin ($p_0 = -0.5$) and has zero velocity ($\dot{p}_0 = 0$). The episode can conclude when the agent reaches the goal position $p_t = 0.5$, regardless of $\dot{p}_t$. An episode can conclude if the length of the action sequence exceeds the specified policy degrees of freedom.

Simulation results, in the form of cost-to-go surface plots, are presented in fig. 1. Each column of plots represents the best-performing policy, with a certain number of degrees of freedom, out of 1000 Monte Carlo simulations. The policy degrees of freedom, which is bounded above by the number of bits per action sequence, is controlled by γ.

In the cost-to-go surfaces, states with higher costs are denoted using warmer colors, while states with lower costs are denoted with cooler colors. Given the initial conditions for the agent, we expect a valley-like region of low cost to form in the surface that begins at $(p_0, \dot{p}_1) = (-0.5, 0)$ and concludes at $(p_t, \dot{p}_t) = (0.5, \cdot)$. The agent travels along this valley, if it exists, to reach the goal condition. The number of times the agent passes through the line $(p_t, \dot{p}_t) = (-0.5, \cdot)$ provides a lower bound on the number of oscillations around the mountain basin. It is well known that the mountain-car problem can be solved with a single basin oscillation.

There are a few findings that can be gleaned from these plots. The most immediate is that there is an improvement in the agent's behavior as the policy degrees of freedom are decreased. In the left-most column, which corresponds to a policy dimensionality of 625 (see fig. 1(a)), it can be seen that a sub-optimal policy is returned early in the learning process. The agent requires two oscillations



(a) (b) (c) (d)

Fig. 1: Cost-to-go surface plots for the mountain car problem. For these plots, the x axis highlights the vehicle's position in the environment. The y axis gives its velocity. The z axis shows the costs associated with that position and velocity pair. A smaller plot in the bottom, left-hand corner highlights an alternate view of the cost-to-go surface. Higher costs in these plots are associated with warmer colors. Lower costs are associated with cooler colors.

Going from left to right, each column of plots highlights the cost-to-go surface for the best-performing policy with the following degrees of freedom: 625 (a), 500 (b), 375 (c), 75 (d), respectively. Going from top to bottom, each row shows the early evolution of the cost-to-go surface at episode 10 and 20, respectively. The number of actions (steps) needed for the agent to reach the goal is shown in the plot title.

around the mountain basin before it can reach the goal. In this case, the agent reverses to reach $(p_t, \dot{p}_t) = (-0.81, 0)$, accelerates to reach $(p_t, \dot{p}_t) = (0.01, 0)$, reverses again to reach $(p_t, \dot{p}_t) = (-1.29, 0)$, and finally accelerates to arrive at the goal $(p_t, \dot{p}_t) = (0.5, 0.02)$. There are instances where the agent performs some unnecessary actions, such as applying no throttle. Each of these general trends follows from tracing the low-cost valley in the cost-to-go surface.

As the policy degrees of freedom are reduced, the agent is able to find increasingly better policies that converge to this behavior. When the degrees of freedom are 500 (see fig. 1(b)), the agent needs two oscillations around the basin to ascend the mountain. The agent starts from rest by reversing to reach $(p_t, \dot{p}_t) = (-0.87, 0)$, accelerating to reach $(p_t, \dot{p}_t) = (0.01, 0)$, reversing to reach $(p_t, \dot{p}_t) = (-1.15, 0)$, and finally accelerating to reach $(p_t, \dot{p}_t) = (0.5, 0.01)$. Not reversing as much on the second oscillation, when compared to the preceding case in fig. 1(a), allowed for the agent to take fewer steps overall to arrive at the goal. When the degrees of freedom are 325 (see fig. 1(c)), the agent needed only a single oscillation to reach the goal. It began by reversing to reach $(p_t, \dot{p}_t) = (-1.15, 0)$ and accelerating to reach $(p_t, \dot{p}_t) = (0.5, 0.03)$.

This improvement occurs due to a decrease in the size of the action search domain. That is, less exploration is needed to find an optimal policy, provided that it exists in the given size-constrained search domain. The agent can therefore switch to exploitation early during the training process.

Setting the policy degrees of freedom too low can adversely impact the returns. In such cases, the agents may not be capable of performing enough actions to reach the goal. For this problem, we encountered this issue when setting the degrees of freedom below 75 (see fig. 1(d)). The agent simply oscillated aimlessly around the basin and was unable to climb the mountain. This behavior is evident from the lack of a clear low-cost valley in the cost-to-go surface.

The value-of-information-based policies tend to converge quickly to a steady state. This behavior is a byproduct of how actions are selected. In particular, the influence of the agent's actions on the expected returns is taken into account when randomly deciding which action should be taken. Early in the learning process, the expected returns will be low, which promotes significant exploration in an attempt to improve those returns. As better action sequences are found in the first few episodes, the returns rise, leading the agent to start exploiting. Eventually, the returns reach a point where exploitation becomes the dominant strategy. For our simulations here, this happened around the twentieth to thirtieth episode. The agent will continue to explore, though, until the sized-constrained search domain is thoroughly investigated. It may take many iterations before this event occurs, however.

To provide context for these results, we compared against epsilon-greedy and soft-max based exploration. We found that learning using Boltzmann-based exploration required anywhere from ten to thirty times the number of episodes to reach the average performance of value-of-information-derived policies from the first twenty episodes. Learning using epsilon-greedy-based exploration needed anywhere from forty to more than a hundred times the number of episodes to achieve policies whose performance matched the average returns from value-of-information-derived policies at twenty episodes.

4. CONCLUSIONS

We have introduced an information-theoretic approach of addressing the exploration-exploitation dilemma in reinforcement learning. Our approach is based on a value of information criterion. When applied to reinforcement learning, this criterion provides a trade-off between the degrees of freedom for a policy, and hence an upper bound for the amount of exploration, and the expected policy returns.

We have demonstrated that the use of this criterion can have a profound impact on the received returns. When limiting a policy's degrees of freedom, the agent explores the policy search space coarsely. This can permit the learning process to converge quickly to (near-) optimal policies. The agent then can switch to exploiting the learned policy after a short period of time, the actual length of which will be application dependent. Conversely, raising a policy's degrees of freedom causes a finer exploration of the search space.

5. REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[2] M. Wiering and M. van Otterlo (Eds.), *Reinforcement Learning: State of the Art*. New York, NY, USA: Springer, 2012.

[3] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1988. [Online]. Available: http://dx.doi.org/10.1023/A:1022633531479

[4] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992. [Online]. Available: http://dx.doi.org/0.1023/A:1022676722315

[5] M. Wiering and J. Schmidhuber, "Fast online $Q(\lambda)$," *Machine Learning*, vol. 33, no. 1, pp. 105–115, 1998. [Online]. Available: http://dx.doi.org/10.1023/A:1007562800292

[6] S. B. Thrun and K. Möller, "Active exploration in dynamic environments," in *Advances in Neural Information Processing Systems*, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. Cambridge, MA, USA: MIT Press, 1992, pp. 531–538.

[7] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, no. 1, pp. 237–285, 1996. [Online]. Available: http://dx.doi.org/10.1613/jair.301

[8] P. Auer, "Using confidence bounds for exploration-exploitation trade-offs," *Journal of Machine Learning Research*, vol. 3, no. 1, pp. 397–422, 2002.

[9] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multi-armed bandit problem," *Machine Learning*, vol. 47, no. 2, pp. 235–256, 2002. [Online]. Available: http://dx.doi.org/10.1023/A:1013689704352

[10] S. Mannor and J. N. Tsitsiklis, "The sample complexity of exploration in the multi-armed bandit problem," *Journal of Machine Learning Research*, vol. 5, no. 12, pp. 623–648, 2004.

[11] A. W. Moore and C. G. Atkenson, "Prioritized sweeping: Reinforcement learning with less data and less real time," *Machine Learning*, vol. 13, no. 1, pp. 103–130, 1993. [Online]. Available: http://dx.doi.org/10.1007/BF00993104

[12] W. G. Macready and D. H. Wolpert II, "Bandit problems and the exploration/exploitation tradeoff," *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 1, pp. 2–22, 1998. [Online]. Available: http://dx.doi.org/10.1109/4235.728210

[13] M. Guo, Y. Liu, and J. Malec, "A new Q-learning algorithm based on the Metropolis criterion," *IEEE Transactions on Systems, Man, and Cybernetics: B*, vol. 34, no. 5, pp. 2140–2143, 2005. [Online]. Available: http://dx.doi.org/10.1109/TSMCB.2004.832154

[14] J. Chen, B. Xin, Z. Peng, and J. Zhang, "Optimal contraction theorem for exploration-exploitation tradeoff in search and optimization," *IEEE Transactions on Systems, Man, and Cybernetics: A*, vol. 39, no. 3, pp. 680–691, 2009. [Online]. Available: http://dx.doi.org/10.1109/TSMCA.2009.2012436

[15] P. Dayan and T. J. Sejnowski, "Exploration bonuses and dual control," *Machine Learning*, vol. 25, no. 1, pp. 5–22, 1996. [Online]. Available: http://dx.doi.org/10.1007/BF00115298

[16] S. P. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Machine Learning*, vol. 38, no. 3, pp. 287–308, 2000. [Online]. Available: http://dx.doi.org/10.1023/A:1007678930559

[17] R. I. Brafman and M. Tennenholtz, "A general polynomial time algorithm for near-optimal reinforcement learning," *Journal of Machine Learning Research*, vol. 3, no. 1, pp. 213–231, 2002. [Online]. Available: http://dx.doi.org/10.1162/153244303765208377

[18] R. L. Stratonovich, "On value of information," *Izvestiya of USSR Academy of Sciences, Technical Cybernetics*, vol. 5, no. 1, pp. 3–12, 1965.

[19] R. L. Stratonovich and B. A. Grishanin, "Value of information when an estimated random variable is hidden," *Izvestiya of USSR Academy of Sciences, Technical Cybernetics*, vol. 6, no. 1, pp. 3–15, 1966.

[20] R. L. Stratonovich, *Information Theory*. Moscow, Russia: Sovetskoe Radio, 1975.

[21] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning: Planning and teaching," *Machine Learning*, vol. 8, no. 3, pp. 293–321, 1992. [Online]. Available: http://dx.doi.org/10.1007/BF00992699

[22] S. P. Singh, "Transfer of learning by composing solutions of elemental sequential tasks," *Machine Learning*, vol. 8, no. 3, pp. 323–339, 1992. [Online]. Available: http://dx.doi.org/10.1007/BF00992700

[23] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: John Wiley and Sons, 2006.

[24] G. Tesauro, "Practical issues in temporal difference learning," *Machine Learning*, vol. 8, no. 3, pp. 257–277, 1992. [Online]. Available: http://dx.doi.org/10.1007/BF00992697

[25] J. A. Boyan and A. W. Moore, "Generalization in reinforcement learning: Safely approximating the value function," in *Advances in Neural Information Processing Systems*, G. Tesauro, D. S. Touretzky, and T. Leen, Eds. Cambridge, MA, USA: MIT Press, 1995, pp. 369–376.