Restricted Optimism

Snir Cohen
Restricted Optimism

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science

Snir Cohen

Submitted to the Senate of
The Technion — Israel Institute of Technology
Cheshvan 5778 Haifa November 2017
The research thesis was done under the supervision of Prof. Shie Mannor in the Department of Computer Science.

I would like to thank my wife, Hilla, for her invaluable support along the way.

The generous financial help of the Technion is gratefully acknowledged.
Contents

Abstract

Abbreviations and Notations

1 Introduction

2 Related Work

2.1 Stochastic Multi-Armed Bandit (MAB)

2.1.1 The optimistic approach

2.1.2 The Bayesian approach

2.2 Finite-horizon MDP

2.2.1 The optimistic approach

2.2.2 The Bayesian approach

2.2.3 An integration of the optimistic and the Bayesian approaches

3 Problem Formulation
List of Figures

4.1 Average cumulative regret and standard deviation of PSRL and UCFH over 1,000 trials and 10,000 episodes in a random MDP setting. . . . 14

5.1 Average cumulative regret of PSRL, PSCS(L), ROPS(L) and UCFH over 1,000 trials and 10,000 episodes. We choose $L \in \{2, 3, 4, 5\}$. . . . 32

5.2 PSRL and ROPS(L) average-standard deviation trade-off after 10,000 episodes for 1,000 trials. We choose $L \in \{2, 3, 4, 5, 10, 15, 20, 25, 30, 35, 40\}$. 34

5.3 PSRL and ROPS(5) histograms after 10,000 episodes for 1,000 trials. 35

5.4 PSRL and ROPS 90th, 95th and 99th percentiles after 10,000 episodes for 1,000 trials. We choose $L \in \{2, 3, 4, 5, 10, 15, 20, 25, 30, 35, 40\}$. . . . 36

5.5 Average value function comparison of PSRL, ROPS(L) and UCFH over 1,000 trials and 10,000 episodes. We choose $L \in \{2, 3, 4, 5, 10, 15, 20\}$. 38

5.6 FrozenLake8x8 from OpenAI gym. . . . . . . . . . . . . . . . . . . . 40

5.7 Average cumulative regret of PSRL and ROPS(L) over 100 trials and 50,000 episodes. We choose $L \in \{2, 3, 4\}$. . . . . . . . . . . . . . . . . . . . 42

5.8 PSRL and ROPS(L) average-standard deviation trade-off after 50,000 episodes for 100 trials. We choose $L \in \{2, 3, 4\}$. . . . . . . . . . . . . . . . . . . . 43

5.9 PSRL and ROPS 90th, 95th and 99th percentiles after 50,000 episodes for 100 trials. We choose $L \in \{2, 3, 4\}$. . . . . . . . . . . . . . . . . . . . 44
List of Tables

5.1 Comparison of the average cumulative regret, standard deviation and 90th, 95th and 99th percentiles for PSRL, ROPS(L), PSCS(L) and UCFH over 1,000 trials after 10,000 episodes ........................................ 45

5.2 Comparison of the average cumulative regret, standard deviation and 90th, 95th and 99th percentiles for PSRL and ROPS(L) over 100 trials after 50,000 episodes ..................................................... 45
Abstract

Markov Decision Processes (MDPs) are commonly used to model Reinforcement Learning problems in which an agent interacts with an unknown environment aiming to maximize a given criterion. In many real-world applications, a decision maker’s task is to formulate a strategy for a fixed number of steps in order to achieve a certain objective. For example, in adaptive routing, packets must be sent through a network of routers before the maximum number of hops is reached and sending fails, while reducing the latency to a minimum. These kind of problems are referred to as finite-horizon or fixed-horizon. More examples can be found in portfolio management, power systems management and games.

Optimistic methods for solving Reinforcement Learning problems are very popular in the literature. In practice, however, these methods show inferior performance compared to other methods, such as Posterior Sampling. We propose a novel concept of Restricted Optimism to balance the well known exploration vs. exploitation trade-off for finite-horizon MDPs. We harness Posterior Sampling to construct two algorithms in the spirit of our Restricted Optimism principle. We provide theoretical guarantees for them and demonstrate through experiments that there exists a trade-off between the average cumulative regret suffered by the agent and the variance. The agent can influence this trade-off by tuning the level of optimism carried out by our proposed algorithms through a regularization parameter.
### Abbreviations and Notations

#### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDP</td>
<td>Markov Decision Process</td>
</tr>
<tr>
<td>OFU</td>
<td>Optimism in the Face of Uncertainty</td>
</tr>
<tr>
<td>UCRL</td>
<td>Upper Confidence for Reinforcement Learning</td>
</tr>
<tr>
<td>PSRL</td>
<td>Posterior Sampling for Reinforcement Learning</td>
</tr>
<tr>
<td>UCB</td>
<td>Upper Confidence Bound</td>
</tr>
<tr>
<td>PSCS</td>
<td>Posterior Sampled Confidence Sets for Reinforcement Learning</td>
</tr>
<tr>
<td>ROPS</td>
<td>Restricted Optimistic Posterior Sampling for Reinforcement Learning</td>
</tr>
<tr>
<td>MAB</td>
<td>Multi-Armed Bandit</td>
</tr>
<tr>
<td>UCFH</td>
<td>Upper Confidence for Fixed Horizon</td>
</tr>
<tr>
<td>PAC</td>
<td>Probably Approximately Correct</td>
</tr>
<tr>
<td>BOSS</td>
<td>Best Of Sampled Set</td>
</tr>
<tr>
<td>BEB</td>
<td>Bayesian Exploration Bonus</td>
</tr>
<tr>
<td>OPS</td>
<td>Optimistic Posterior Sampling</td>
</tr>
<tr>
<td>MLE</td>
<td>Maximum Likelihood Estimation</td>
</tr>
</tbody>
</table>
Notations

\( M \) — An MDP
\( S \) — Finite state space
\( A \) — Finite action space
\( p \) — Transition matrix
\( r \) — Reward function
\( H \) — Horizon
\( r(s, a) \) — Reward obtained at state \( s \) after choosing action \( a \)
\( p(s' | s, a) \) — The probability of transitioning from state \( s \) to state \( s' \) after choosing action \( a \)
\( \rho \) — Initial state distribution
\( \phi \) — Prior distribution of the true MDP
\( K \) — Number of episodes
\( T \) — Total number of timestamps
\( \mathcal{H}_k \) — The history prior to episode \( k \)
\( \Pi \) — Set of all deterministic time-variant policies
\( \mu \) — Policy
\( V^M_{\mu, h}(s) \) — Value function starting from state \( s \) at timestamp \( h \) with respect to MDP \( M \) and a policy \( \mu \)
\( \text{Regret}(T, \pi, M^*) \) — Regret of a reinforcement learning algorithm \( \pi \)
\( \Delta_k \) — Regret of the \( k \)th episode
\( \pi_k \) — Policy used by our algorithms at episode \( k \)
\( \text{BayesRegret}(T, \pi, \phi) \) — Bayesian Regret of a reinforcement learning algorithm \( \pi \) given prior \( \phi \)
\( \mathcal{M}_k \) — A plausible set of sampled MDPs at the \( k \)th episode
\( C_k \) — A confidence set of MDPs at the \( k \)th episode
\( L \) — Regularization parameter
\( \bar{V}_{k, h} \) — The optimistic value function at episode \( k \) and timestamp \( h \)
\( \text{Dir} \) — The Dirichlet distribution
\( \text{Beta} \) — The Beta distribution
\( x \) — State-action pair from \( S \times A \)
\( n_k(x) \) — Number of times the state-action pair \( x \) was visited prior to the \( k \)th episode
\( \sigma^2 \) — Variance
\( \mathbb{I}_{\text{Event}} \) — Indicator function of Event
Chapter 1

Introduction

In sequential decision problems, the most fundamental question a decision maker faces is how to balance between exploration and exploitation. She may choose to maximize immediate reward based on her current knowledge (exploit) or settle for a possibly lower immediate reward expecting a higher future reward (explore). This dilemma is known in the literature as the exploration vs. exploitation trade-off. On one hand, constant exploitation might lead to a sub-optimal policy as the agent might refrain from choosing optimal actions due to initial poor estimations. On the other hand, exaggerated exploration might also lead to inferior performance as the agent selects sub-optimal actions many times. Therefore, efficient algorithms must carefully balance between exploration and exploitation. In this thesis, we propose a novel concept by the name of Restricted Optimism to address this trade-off for the problem of learning to optimize a finite-horizon Markov Decision Process (MDP) that is, to maximize the cumulative reward obtained up to time $T$ by the agent as presented by Burnetas and Katehakis [1997]. It is often more useful to measure algorithms’ performance according to the concept of cumulative regret rather than cumulative reward. The regret is defined as the difference between the expected value of an optimal policy and the expected value of the policy the agent decided upon.

The most common approach for balancing exploration and exploitation relies on Upper Confidence Bounds (UCB) and is referred to as the Optimism in the Face of Uncertainty (OFU) principle. Upper Confidence for Reinforcement Learning (UCRL), an efficient algorithm based on this paradigm, was proposed by Auer and Ortner [2007] and was later improved by Jaksch et al. [2010] which presented UCRL2. The latter constructs a confidence interval based on previously obtained rewards and observed transitions for every unknown parameter, and defines a confidence set of plausible MDPs using the Cartesian product over these intervals. Given this confidence set, UCRL2 chooses an MDP from this set along the lines of the optimistic principle, calculates a near-optimal policy for this optimistic MDP and
follows this policy throughout the episode. The authors provide an upper bound on the cumulative regret showing the near-optimality of the algorithm.

Another very common method is a Bayesian approach proposed by Thompson [1933]. The basic idea is to maintain a posterior distribution for every unknown parameter and generate estimates for these parameters by sampling from these distributions. Strens [2000] proposes an algorithm now known as Posterior Sampling for Reinforcement Learning (PSRL) based on this idea. Empirical results from Osband et al. [2013] and Osband and Van Roy [2016] demonstrate that UCRL2 presents inferior performance in comparison to PSRL. Osband and Van Roy [2016] provide an improved upper bound compared to UCRL2 for the case of Bayesian Reinforcement Learning.

We introduce the principle of Restricted Optimism. Our main claim is that UCB-like algorithms, such as UCRL2, tend to construct too broad confidence sets every episode. This results in choosing overly optimistic MDPs and consequently higher regret and poor empirical performance. We argue that smaller confidence sets can yield a less, but still optimistic MDP and furthermore, the level of optimism can be tuned. Therefore, our approach can be considered as a member of the OFU paradigm family. Unlike classic OFU algorithms, we forgo the standard tool of confidence intervals. Instead, we harness Posterior Sampling to define finite, yet rich enough confidence sets.

We outline two algorithms in the spirit of our Restricted Optimism principle called Posterior Sampled Confidence Sets for Reinforcement Learning (PSCS) and Restricted Optimistic Posterior Sampling for Reinforcement Learning (ROPS). Both maintain a posterior distribution for every unknown parameter and sample \( L \geq 1 \) MDPs from it at the beginning of every episode. These samples will constitute the basis of the confidence sets; the difference between the two algorithms is the way they form these confidence sets. Similar to other optimistic oriented algorithms, they choose an optimistic MDP from these confidence sets and solve an optimal policy for it. \( L \) is used as a regularization parameter which controls the level of optimism of the algorithms. For the choice of \( L = 1 \), these algorithms are identical to PSRL and therefore can be considered as its extensions.

As mentioned earlier, comparing the average cumulative regret of PSRL and UCRL2 proves that PSRL significantly outperforms UCRL2. Yet, both algorithms suffer from high variance. In addition, plotting the average value function shows that UCRL2 tends to overestimate the value function of the MDP of its choice—in accordance with its optimistic nature—, whereas PSRL is more inclined to underestimate it. This implies that PSRL cannot be considered an optimistic algorithm. Our proposed algorithms enable controlling the degree of optimism by setting the regularization parameter \( L \) and demonstrate via experiments that there exists a trade-off between the average cumulative regret and the variance. In particular, as
$L$ grows, the average cumulative regret increases, while the variance decreases. This result allows a user that can tolerate higher average cumulative regret to benefit from a lower variance and consequently a more stable performance.

The main contributions of this thesis are the following:

- Introduce the novel principle of *Restricted Optimism* for optimizing finite-horizon MDPs.
- Describe two algorithms in the spirit of this concept and provide theoretical guarantees for them similar to those provided for PSRL by Osband and Van Roy [2016].
- Demonstrate via experiments that there exists a trade-off between the average cumulative regret and the standard deviation, and that this trade-off can be controlled by the level of optimism of the algorithms.

This thesis is organized as follows: In Chapter 2 we review related work. In Chapter 3 we formally present the finite-horizon MDP problem. In Chapter 4 we discuss our Restricted Optimism principle, describe two algorithms in the spirit of this principle and provide theoretical guarantees for them. In Chapter 5 we assess the performance of our proposed algorithms to state-of-the-art algorithms. We conclude with a discussion in Chapter 6.
Chapter 2

Related Work

In this chapter we survey previous work related to finite-horizon MDPs and to efficient learning algorithms aiming to optimize them.

2.1 Stochastic Multi-Armed Bandit (MAB)

Our primary focus is dealing with the challenge of balancing between exploration and exploitation efficiently. Conventionally, in its purest form, this trade-off is associated with the stochastic Multi-Armed Bandit (MAB) problem presented by Robbins [1952]. In this setting, every timestamp the agent chooses an arm which yields a reward from an unknown distribution associated with this arm. Common methods for solving this setting can be divided into two major categories: an optimistic approach, which is based on upper confidence bounds, and a Bayesian approach.

2.1.1 The optimistic approach

Lai and Robbins [1985] prove under a few mild assumptions that any bandit algorithm must pull every sub-optimal arm at least \( o(\log n) \) times (asymptotically). They also propose algorithms that asymptotically achieve the lower bound, however these algorithm are intractable. In the seminal work by Gittins [1979], index policies are introduced. Gittins proves that in order to solve the problem efficiently, the agent only needs to calculate a priority index for each arm based on the samples obtained for each arm separately, and simply choose the arm with the highest index at every timestamp. Agrawal [1995] proposes to use upper confidence bounds to construct index policies which were computationally simpler. These policies were followed by the celebrated UCB1 algorithm presented by Auer et al. [2002]. UCB1 is a simple and easy to implement method based on upper confidence bounds aiming
to minimize the expected regret suffered by the agent up to time $T$. According to this algorithm, every arm is given an estimate based on its mean samples and some bonus. This bonus decays as more samples are received and the agent is more certain of her mean estimate. This paper initiated a line of works aiming to close the gap between the UCB1’s theoretical upper bound proved by Auer et al. [2002] and the lower bound. Notable examples are UCB-V by Audibert et al. [2009], MOSS by Audibert and Bubeck [2010] and KL-UCB by Garivier and Cappé [2011]. The latter is the first index policy that reaches the lower bound by Lai and Robbins [1985] for binary rewards.

2.1.2 The Bayesian approach

In an early work on stochastic bandit problems by Thompson [1933], a Bayesian approach was suggested to minimize the cumulative regret. In short, every arm is associated with a prior distribution of the parameters of its reward distribution and at every timestamp, the agent estimates each arm using a sample from its posterior distribution, and chooses the arm with the highest sample. Although this algorithm, which is referred to as Thompson Sampling, is known for a long time, it was not until Chapelle and Li [2011] presented excellent empirical results that the Bayesian approach became popular. Their experiments show that Thompson Sampling presents competitive performance comparing to state-of-the-art UCB-based algorithms and even outperforms them in many settings. Agrawal and Goyal [2012] prove a logarithmic upper bound on the expected cumulative regret. Kaufmann et al. [2012] improve this bound and provide an upper bound matching the asymptotic lower bound. Agrawal and Goyal [2013] also provide a problem-independent bound.

2.2 Finite-horizon MDP

2.2.1 The optimistic approach

Naturally, the results presented by Auer et al. [2002] encouraged researchers to apply optimistic methods to other settings. For the setting in question, i.e., the finite-horizon MDP, UCRL was proposed by Auer and Ortner [2007] and was later improved by Jaksch et al. [2010] who suggested UCRL2. Both adopt upper confidence bounds to choose an optimistic MDP from a confidence set. The authors prove that UCRL2’s total regret is upper bounded by $\tilde{O}(DS\sqrt{AT})$ after $T$ steps for any unknown MDP with $S$ states, $A$ actions and diameter $D$. (The notation $\tilde{O}(\cdot)$ hides logarithmic factors). Note that whereas calculating the best arm at every timestamp for the MAB setting is straightforward, this is not the case for the
finite-horizon MDP setting. The confidence set constructed by UCRL2 is a Cartesian product of confidence intervals around the mean estimates of both the reward function and the transition distributions. This set contains all MDPs \( M = (\tilde{p}, \tilde{r}) \) satisfying the following constraints:

\[
\begin{align*}
|\tilde{r}(s, a) - \hat{r}(s, a)| & \leq \sqrt{\frac{7 \log (2SA\eta_k/\delta)}{2 \max \{1, n_k(s, a)\}}} \\
\|\tilde{p}(\cdot | s, a) - \hat{p}(\cdot | s, a)\|_1 & \leq \sqrt{\frac{14S \log (2A\eta_k/\delta)}{\max \{1, n_k(s, a)\}}},
\end{align*}
\]

where \( t_k \) denotes the start time of the \( k \)th episode, \( n_k(s, a) \) denotes the number of times the state-action pair \((s, a)\) was visited prior to episode \( k \), and \( \tilde{r}(s, a) \) and \( \hat{r}(s, a) \) are mean estimates of the transition and the reward distributions, respectively. The bounds on the right side of these two expressions are inspired by concentration inequalities such as the Hoeffding inequality. This results in an infinite set of MDPs and requires using the Extended Value Iteration method to find a near-optimal policy for an optimistic MDP from this confidence set.

Bartlett and Tewari [2009] propose a similar optimistic algorithm named REGAL. The authors prove that REGAL achieves an upper bound of \( \tilde{O}(HS\sqrt{AT}) \) on the expected regret for horizon \( H \). Same as UCRL2, REGAL works in episodes, and at every episode, it chooses an MDP from the confidence set that maximizes a "regularized" average optimal reward. REGAL’s confidence set is identical to the one UCRL2 defines up to a constant.

One disadvantage of UCRL2 is the uneven length of its episodes. UCRL2 acts according to the same policy until a certain condition is satisfied. Policy updates are not necessarily aligned with the horizon. To mitigate this issue, Dann and Brunskill [2015] propose Upper Confidence for Fixed Horizon (UCFH), a variant of UCRL2, optimized for finite-horizon MDPs. The analysis of UCFH presents a Probably Approximately Correct (PAC) guarantee rather than an upper bound on the expected cumulative regret. Namely, for any \( 0 < \epsilon, \delta \leq 1 \), with probability at least \( 1 - \delta \), following UCFH yields at most \( \tilde{O}\left(\frac{H^2S^2A}{\epsilon^2} \ln \frac{1}{\delta}\right) \) episodes for which the expected regret is greater than \( \epsilon \). This PAC upper bound improves previously obtained bounds (e.g. Kakade et al. [2003], Strehl et al. [2006] and Jaksch et al. [2010]). We do not consider PAC-style guarantees in this work.

### 2.2.2 The Bayesian approach

Inspired by Thompson [1933] and Strens [2000] formulates an algorithm based on posterior sampling for solving finite-horizon MDPs. This algorithm was initially referred to as “Bayesian Dynamic Programming”, but is currently known as Posterior Sam-
pling for Reinforcement Learning (PSRL). PSRL maintains a posterior distribution for every unknown parameter. It samples from these distributions parameters every episode which construct an MDP. It then solves an optimal policy with respect to the sampled MDP and acts according to this policy throughout the episode. Osband et al. [2013] provide an upper bound similar to the one provided for REGAL. It should be emphasized that the bound provided for PSRL requires the assumption that the true MDP is sampled from a known prior distribution. This setting is known as Bayesian Reinforcement Learning. The mentioned upper bound was improved later by Osband and Van Roy [2016] to $\tilde{O} \left( \frac{H}{\sqrt{SAT}} \right)$. The authors also include a conjecture, stating that a more careful analysis should obtain a tighter upper bound of $\tilde{O} \left( \sqrt{HSAT} \right)$.

### 2.2.3 An integration of the optimistic and the Bayesian approaches

Another category of algorithms contains methods that combine the optimistic and Bayesian approaches. Asmuth et al. [2009] propose Best of Sampled Set (BOSS). BOSS samples $K$ models from the posterior distribution, merges them into one mixed MDP and follows an optimal policy for this MDP. Another example of integrated algorithms is Bayesian Exploration Bonus (BEB), which was introduced by Kolter and Ng [2009]. BEB tackles this problem from a different point of view. It adds reward bonus for state and action pairs inversely proportional to the number of times they were visited. Both BOSS and BEB have PAC guarantees.

An empirical work by Fonteneau et al. [2013] suggests Optimistic Posterior Sampling (OPS) to obtain optimistic Q-values based on several sampled MDPs for the discounted case. Originally, OPS was designed to solve an infinite-horizon MDP problem. Osband and Van Roy [2016] modified the algorithm in their experiments to meet the requirements of the finite-horizon setting. According to this variant, $K$ MDPs are sampled and their Q-values are computed. Then, at each state and timestamp, the action that maximizes the corresponding Q-values is chosen. Running this variant with different values of $K$, the authors concluded that the results are very similar to PSRL, thus determined that PSRL is preferable due to its lower computational cost. In other words, since solving MDPs is computationally expensive, the size of the confidence set used in these type of algorithms must increase in a higher rate than simply linearly as a function of the number of sampled MDPs. We address this issue in detail throughout this work.
Chapter 3

Problem Formulation

In this chapter we provide a formal description of the finite-horizon MDP setting, introduce the notations used throughout the paper and present the regret measure function.

A time-invariant Markov decision process is defined as $M^* = (S, A, p^*, r^*, H, \rho)$, where $S = \{1, \ldots, S\}$ is a finite state space, $A = \{1, \ldots, A\}$ is a finite action space, $H$ is the horizon and $\rho$ is the initial state distribution. The transition matrix $p^*$ and the reward function $r^*$ are distributed according to some prior $\phi$. Let $r^*(s, a) = E [r | r \sim r^*(s, a)]$.

The agent interacts with the true MDP $M^*$ in episodes. At the beginning of every episode $k$, the agent observes a state $s_{k1} \sim \rho$. For every timestamp $1 \leq h \leq H$ within the episode, the agent observes a state $s_{kh} \in S$, chooses an action $a_{kh} \in A$, obtains a reward $r_{kh} \sim r^*(s_{kh}, a_{kh})$ and transitions to a new state $s_{kh+1} \sim p^*(\cdot | s_{kh}, a_{kh})$.

A time-variant policy $\mu : S \times \{1, \ldots, H\} \rightarrow A$ is a mapping from state and timestamp pair to an action. Let $\Pi$ denote the set of all deterministic time-variant policies.

For every state $s$ and timestamp $1 \leq h \leq H$, let the value function starting from state $s$ at timestamp $h$ with respect to an MDP $M = (S, A, p, r, H, \rho)$ and a policy $\mu$ be

$$V_{\mu,h}^M(s) = E \left[ \sum_{j=h}^{H} r(s_{1j}, \mu(s_{1j}, j)) | s_{1h} = s \right]$$ (3.1)

where the expected value is taken over the random rewards obtained according to $r$ and the random transitions made according to $p$. 


With a slight abuse of notation, let $M = \left( S, A, \{ p_h \}_{h=1}^H, \{ r_h \}_{h=1}^H, H, \rho \right)$ be a time-variant MDP, where $p_h$ and $r_h$ are $M$’s transition matrix and reward function at timestamp $h$, respectively. We define the value function for a time-variant MDP $M$ analogously. As $S$, $A$, $H$ and $\rho$ remain unchanged throughout the paper, we use $M = (p, r)$ and $M = \left( \{ p_h \}_{h=1}^H, \{ r_h \}_{h=1}^H \right)$ to shorten our notations.

Let $\mu^M$ denote the optimal policy for MDP $M$; that is, $\mu^M$ satisfies $V_{\mu^M, h}^M (s) \geq V_{\mu, h}^M (s)$ for every policy $\mu$, state $s \in S$ and timestamp $1 \leq h \leq H$.

Define the history $H_k$ to be the sequence of obtained rewards and the sequence of transitions made by the agent prior to episode $k$.

Let $K = \lceil T/H \rceil$ be the number of episodes. We measure the regret of a reinforcement learning algorithm $\pi = (\pi_1, \ldots, \pi_K)$ up to time $T$ by

$$\text{Regret} (T, \pi, M^*) = \sum_{k=1}^K \Delta_k,$$

where $\Delta_k$ is the regret of the $k$th episode measured with respect to the true MDP $M^*$,

$$\Delta_k = \sum_{s \in S} \rho (s) \left( V_{\mu^M, h}^* (s) - V_{\pi_k, h}^* (s) \right).$$

We also define the Bayesian regret as the expected value of the regret taken over MDPs sampled from the prior $\phi$:

$$\text{BayesRegret} (T, \pi, \phi) = \mathbb{E} \left[ \text{Regret} (T, \pi, M^*) \mid M^* \sim \phi \right].$$

Note that according to this Bayesian setting, the true MDP $M^*$ is considered as a random variable. The agent’s goal is to minimize the expected cumulative regret over $T$ steps of interaction with the environment.

To ease our notations, we write $V^*$ instead of $V_{\mu^M}^*$ and $V_{\pi}^*$ instead of $V_{\mu^M}^*$. In general, we denote the value function with respect to an MDP $M_{k\ell}$ and an optimal policy for it $\mu^{M_{k\ell}}$ starting from state $s$ at timestamp $h$ by $V_{k\ell, h}^{M_{k\ell}} (s)$. 
Chapter 4

Restricted Optimism via Posterior Sampling

In this chapter we present a unified framework for both the optimistic and the Bayesian approaches and introduce two algorithms in the spirit of our Restricted Optimism principle. We also state and prove an upper bound on the cumulative expected regret of the second algorithm.

4.1 Motivation

We begin our discussion with an example of a comparison between the performance presented by UFCH, an optimized version of UCRL2 for finite-horizon MDPs suggested by Dann and Brunskill [2015], and PSRL, a Bayesian algorithm as described by Osband and Van Roy [2016]. In this experiment, we run both algorithms on a random generated MDP consists of 20 states and 10 actions with $H = 40$ for 10,000 episodes and 1,000 independent trials. Full experiment details can be found in the Chapter 5 along with a more thorough analysis. We plot the average cumulative regret suffered by the agent and the standard deviation.
Supporting our aforesaid argument, Figure 4.1 shows that the Bayesian algorithm outperforms the optimistic algorithm significantly, and even though both algorithms have theoretical guarantees, it is clear that PSRL is more practical than UCFH.

A more closer look, reveals that after 10,000 episodes, both algorithms have similar standard deviation. This result is quite surprising. Imagine an Olympic archer standing 50 meters from her target and shooting 1,000 times. Due to the long distance, one cannot expect that all the arrows would hit the exactly the same spot and her hits are going to be quite scattered. Let us revisit this exercise once more, only now the archer stands 5 meters from the target. Again, it is unreasonable to assume that all her shots will hit the same spot, however we can definitely expect them to be much less scattered as she is 10 times closer to the target. This intuitive expectation does not apply in our case according to Figure 4.1. Although the average cumulative regret presented by PSRL is without a doubt much better than UCFH’s, the variance remains high.

In this thesis, we aim for a middle ground. Our goal is to utilize optimism in a more moderate fashion using Posterior Sampling to benefit from an average cumulative regret similar to PSRL, but decrease the variance.
4.2 A unified framework for both the optimistic and the Bayesian approaches

In the context of finite-horizon MDPs, the concept of Optimism in the Face of Uncertainty (OFU) translates into constructing a confidence set of MDPs \( C_k \), which is supposed to contain the true MDP with high probability. An optimistic MDP \( M_k \in C_k \) is chosen from this set and the agent follows an optimal policy \( \pi_k \) with respect to this optimistic MDP throughout the episode.

In order to adjust this description to the Bayesian approach presented by algorithms such as PSRL and BOSS, we add a prior to the input of the framework. Algorithm 1 outlines the unified framework. Our proposed algorithms sample \( L \geq 1 \) MDPs from the posterior distribution and provide different implementations for the \texttt{ConstructConfidenceSet} method.

Algorithm 1 A unified framework for the optimistic and the Bayesian approaches

1: input: prior \( \phi \), \( L \geq 1 \), \( H_1 = \emptyset \).
2: for episode \( k = 1, 2, \ldots \) do
3: \( C_k \leftarrow \texttt{ConstructConfidenceSet}(\phi, H_k, L) \).
4: Solve \( \pi_k \in \arg\max_{\pi \in \Pi} \max_{M \in C_k} \mathbb{E}[V_{\pi, M}] \).
5: Observe state \( s_{k1} \).
6: for timestamp \( h = 1, 2, \ldots, H \) do
7: Take action \( a_{kh} = \mu_k(s_{kh}, h) \).
8: Observe reward \( r_{kh} \) and next state \( s_{kh+1} \).
9: Update history \( H_k \leftarrow H_k \cup \{s_{kh}, r_{kh}\} \).
10: end for
11: end for

As suggested by [Strens 2000], we use the Dirichlet distribution to model the transitions posterior, as it is a conjugate-prior to the Multinomial distribution. As for the rewards, for the rest of the thesis, we make the following two assumptions:

**Assumption 1.** The support of the rewards is \([0, 1]\).

**Assumption 2.** The rewards are drawn from the Bernoulli distribution.

The first is a standard assumption in the literature. The second allows us to use the Beta distribution to model the rewards posterior. This assumption can be easily lifted, however it would add unnecessary complexity to our analysis. We describe how to modify the algorithm to address a general reward distribution with support \([0, 1]\) in Remark 6.
4.3 Posterior Sampled Confidence Set for Reinforcement Learning (PSCS)

Classically, the confidence sets of OFU algorithms are consisted of a Cartesian product of confidence intervals formed around the estimated mean rewards and transition probabilities based on previously observed rewards and transitions. A more general alternative of this approach is to use appropriately generated confidence sets instead of the confidence intervals. In particular, for every unknown parameter:

1. Choose a reference estimate and add it to the parameter’s set of plausible values.
2. Add less likely estimates to this set.

Step 2 is formulated vaguely intentionally so the proposed framework can be considered a generalization for both the OFU and Posterior Sampling approaches. For example, UCRL2 uses the Maximum Likelihood Estimation (MLE) as its reference estimate for every parameter and then extends its confidence interval by adding less likely estimates with respect to the computed MLE.

UCB-like algorithms find this approach useful as their theoretical analysis depend on concentration inequalities such as the Hoeffding inequality. However, it results in continuous confidence intervals and a Cartesian product of these intervals constructs a too broad confidence set which leads to an overly optimistic MDP and inferior experimental results.

In accordance, we propose to construct a set of plausible MDPs, $M_k$, in two steps:

1. Sample a reference MDP from the posterior and add it to $M_k$.
2. Sample $L - 1$ more MDPs and add to $M_k$ only those which are less probable than the reference MDP with respect to the posterior distribution.

According to this procedure, $M_k$ is a set of sampled MDPs ranging in size from 1 to $L$.

Given a set of plausible MDPs $M_k$, OFU algorithms calculate a (near)-optimal policy for an optimistic MDP from $M_k$ and act according to this policy throughout the $k$th episode. UCRL2, for instance, uses the Extended Value Iteration procedure to find such a near-optimal policy for an optimistic MDP from the confidence set. In general, we are interested in the optimistic value function as defined by Iyengar.
\( \overline{V}_{k,1} = \sup_{\pi \in \Pi} \left\{ \sup_{M \in \mathcal{M}_k} \mathbb{E} \left[ V^M_{\pi,1} \right] \right\} \), \hspace{1cm} (4.1)

where \( \overline{V}_{k,1} \) denotes the optimistic value function at episode \( k \) and timestamp \( h = 1 \). In our case, \( \mathcal{M}_k \) and \( \Pi \) are finite sets and therefore \( \sup (\cdot) \) can be replaced by \( \max (\cdot) \). The optimization problem described in (4.1) is not difficult considering that

\[
\max_{\pi \in \Pi} \left\{ \max_{M \in \mathcal{M}_k} \mathbb{E} \left[ V^M_{\pi,1} \right] \right\} = \max_{M \in \mathcal{M}_k} \left\{ \max_{\pi \in \Pi} \mathbb{E} \left[ V^M_{\pi,1} \right] \right\} \tag{4.2}
\]

Solving (4.1) according to equation (4.2) is equivalent to calculating an optimal policy for every MDP in \( \mathcal{M}_k \) and choosing the policy for which \( \mathbb{E} \left[ V^M_{\mu,1} \right] \) has the highest value. However, this procedure yields an algorithm very similar to Optimistic Posterior Sampling (OPS) presented in Fonteneau et al. \[2013\]. Its performance as presented by Osband and Van Roy \[2016\] is similar to PSRL’s performance with additional computational cost.

An alternative to this approach is solving the optimization problem using dynamic programming, which effectively expends the confidence set exponentially. According to this method, different transition matrices and reward functions from the set of plausible MDPs can be chosen at different timestamps. For every state \( s \in \mathcal{S} \), episode \( k \) and timestamp \( 1 \leq h \leq H \), the following holds:

\[
\overline{V}_{k,h}(s) = \max_a \max_{(p,r) \in \mathcal{M}_k} \left\{ r(s,a) + \sum_{s' \in \mathcal{S}} p(s' | s,a) \overline{V}_{k,h+1}(s') \right\}
\]

and also for every state \( s \in \mathcal{S} \) and episode \( k \), \( \overline{V}_{kH+1}(s) = 0 \).

This equation set is similar to the Bellman equation (Theorem 2.1 in Iyengar \[2005\]) devised to efficiently compute the robust value function and is equivalent to solving (4.1) with the following confidence set instead of \( \mathcal{M}_k \):

\[
\mathcal{C}_k = \left\{ M = \left\{ (p_h^{H}_{h=1} , (r_h^{H}_{h=1}) \right\} \mid \forall 1 \leq h \leq H, (p_h, r_h) \in \mathcal{M}_k \right\} \tag{4.3}
\]

It should be emphasized that each option leads to a different value function and a different policy. Our proposed algorithms solve the max-max optimization problem based on \( \mathcal{C}_k \) rather than \( \mathcal{M}_k \) for a couple of reasons. First, PSRL chooses MDPs from the set of time-invariant MDPs. This is a reasonable course of action as the true MDP is known to be a member of this set according to the setting in question. However, Scherrer and Lesner \[2012\] prove for the setting of infinite-horizon stationary MDP with discounted rewards that the accuracy of a learning algorithm can be improved by exploring the larger set of time-variant MDPs even
if it is known that there exists a stationary optimal policy. Second, based on $C_k$, our algorithms choose an MDP from an exponentially larger set of MDPs with no additional computational cost and no additional samples compared to the original formulation, supporting our primary objective to construct an efficient optimistic algorithm.

A full algorithm description can be found in Algorithm 2.

**Algorithm 2** Posterior Sampled Confidence Sets for Reinforcement Learning (PSCS):

```
ConstructConfidenceSet implementation
1: input: prior $\phi$, history $H_k$, $L \geq 1$.
2: Sample $M_{k1} \sim \phi(\cdot | H_k)$ and set $M_k = \{M_{k1}\}$.
3: for $\ell = 2, \ldots, L$ do
4: Sample $M_{k\ell} \sim \phi(\cdot | H_k)$.
5: if $Pr(M_{k\ell}) < Pr(M_{k1})$ then
6: $M_k \leftarrow M_k \cup \{M_{k\ell}\}$.
7: end if
8: end for
9: Construct $C_k$ based on $M_k$ as in (4.3).
10: return $C_k$.
```

The probability of an MDP $M_{k\ell}$ is defined as a product of the pdfs of every sampled parameter of the MDP according to the posterior distributions based on $H_k$.

### 4.4 Restricted Optimistic Posterior Sampling for Reinforcement Learning (ROPS)

PSCS presents an analogous version to OFU algorithms using posterior sampling. However, it has one significant drawback as it assigns a disproportionate weight to the reference MDP compared to the next $L - 1$ sampled MDPs. Even though the samples are independent and the agent does not prefer one over the other, the order of the samples becomes critical, as different orderings may result in very different policies. This might lead to high variance.

In order to solve this issue, we propose a slightly different variant called Restricted Optimistic Posterior Sampling for Reinforcement Learning (ROPS). According to this algorithm, we simply add all $L$ sampled MDPs to $M_k$ and then construct the confidence set $C_k$ based on $M_k$ as done by PSCS. A full algorithm description can be found in Algorithm 3.
Algorithm 3 Restricted Optimistic Posterior Sampling for Reinforcement Learning (ROPS):

ConstructConfidenceSet implementation

1: **input:** prior $\phi$, history $H_k$, $L \geq 1$.
2: Sample $M_{k\ell} \sim \phi(\cdot | H_k)$ for $1 \leq \ell \leq L$.
3: Set $M_k = \{M_{k1}, \ldots, M_{kL}\}$.
4: Construct $C_k$ based on $M_k$ as in (4.3).
5: **return** $C_k$.

The computational complexity of ROPS is equivalent to solving $L$ MDPs using dynamic programming.

**Remark 3.** For the choice $L = 1$, both PSCS and ROPS are identical to PSRL. Therefore, they can be considered as optimistic extensions of PSRL.

We provide an upper bound of the regret for ROPS in the following theorem.

**Theorem 4.** Let $M^*$ be the true MDP distributed according to a prior $\phi$ with any independent Dirichlet prior over the transitions and any independent Beta prior over the rewards. Then, the Bayesian regret for ROPS is bounded by

$$\text{BayesRegret}(T, \pi^{\text{ROPS}}, \phi) = O\left(H \sqrt{SAT \ln (2LSAT)}\right).$$

We prove this theorem in the next section.

This result generalizes PSRL’s regret upper bound provided by Osband and Van Roy [2016] for any $L \geq 1$.

**Remark 5.** The provided proof can be easily modified to satisfy a similar regret upper bound for PSCS as well.

**Remark 6.** Note that PSCS and ROPS can be easily modified to handle any reward distribution with $[0, 1]$ support. This can be achieved by applying the elegant extension presented by Agrawal and Goyal Agrawal and Goyal [2012]: for any reward $0 \leq r_{kh} \leq 1$, perform a Bernoulli trial with success probability $r_{kh}$ and use its output to update the posterior distribution. The analysis remains intact.

4.5 Proof of Theorem 4

This proof mainly follows the proof provided by Osband and Van Roy [2016] of the upper bound on the Bayesian regret of PSRL with some modifications.

The mentioned proof uses a classic result on the tail bounds of sub-Gaussian random variables for upper bounding the deviations from the rewards. As for the devia-
tions from the transitions, it adopts a non-trivial argument presented by Osband et al. [2014] to upper bound a Dirichlet random variable by a matching Gaussian random variable. The authors define the concept of stochastic optimism which is closely related to second order stochastic dominance, and use it to prove that a Gaussian random variable (with certain parameters) is stochastically optimistic for a Dirichlet random variable. This mathematical tool allows them to upper bound the deviations from the transition using the same classic result on the tail bounds of sub-Gaussian random variables they used for the deviations from the rewards.

Instead of using this complex argument, we apply a recent result obtained by Mar- chal and Arbel [2017] which formulates an optimal proxy variance for Beta distributions. This result helps us derive a shorter and simpler proof over all.

For simplicity, we assume that the prior on the transitions is Dir (1,\ldots,1) and the prior on the rewards is Beta (1,1), but the proof remains correct for any Dirichlet and Beta prior distributions.

4.5.1 Posterior Sampling Lemma

The first step of the proof relies on an observation made by Russo and Van Roy [2014] that conditional on the history \( H_k \), every MDP in \( M_k \) is equal in distribution to the true MDP \( M^* \). We state this claim formally in the following lemma.

**Lemma 7** (Posterior Sampling). Let \( \phi \) be the distribution of the true MDP \( M^* \), then for any \( \sigma (H_k) \)-measurable function \( f \) and \( 1 \leq \ell \leq L \),

\[
\mathbb{E} [ f( M_{k\ell} ) \mid H_k ] = \mathbb{E} [ f( M^* ) \mid H_k ] .
\]

Lemma 7 suggests that \( \mathbb{E} [ V^*_{k,1} \mid H_k ] = \mathbb{E} [ V^{k1}_{k1,1} \mid H_k ] \) for every episode \( k \) and times-tamp \( h \). This allows us to decompose the regret as follows:

\[
\mathbb{E} [ \Delta_k \mid H_k ] = \mathbb{E} [ V^*_{k,1} \mid H_k ] - \mathbb{E} [ V^*_{k1,1} \mid H_k ] = \mathbb{E} [ V^{k1}_{k1,1} \mid H_k ] - \mathbb{E} [ V^*_{k1,1} \mid H_k ] \quad (4.4)
\]

Denote by \( M_k \) an MDP from the confidence set \( C_k \) such that

\[
\mathbb{E} [ V^{k1}_{k1,1} ] = \max_{\pi \in \Pi} \max_{M \in C_k} \mathbb{E} [ V^M_{\pi,1} ] .
\]

Clearly, the policy \( \pi_k \) which was chosen by the algorithm is optimal for this MDP. Note that \( M_k \) is a time-variant MDP, whereas \( M_{k1} \) is a time-invariant MDP. Let \( M_{k1} = (p^1_k, r^1_k) \). Equivalently, we can consider \( M_{k1} \) as a time-variant MDP for which \( p^1_k \) and \( r^1_k \) are used as the transition matrix and the reward function, respec-
tively, for every timestamp throughout the episode. Clearly, both forms have the same expected value. In particular, the time-variant form is a member of $C_k$ and therefore according to $M_k$’s definition, $E[V_{k,1}^k \mid H_k] \geq E[V_{k,1}^{k,1} \mid H_k]$, i.e. $M_k$ can be considered more optimistic than the MDP that PSRL has sampled. Hence, (4.4) can be upper bounded by

$$E[\Delta_k \mid H_k] \leq E[V_{k,1}^k \mid H_k] - E[V_{k,1}^* \mid H_k] \quad (4.5)$$

Let $M_k = \left( \{p_{kh}\}_{h=1}^H, \{r_{kh}\}_{h=1}^H \right)$ where $p_{kh}$ and $r_{kh}$ are $M_k$’s transition matrix and reward function at timestamp $h$, respectively. Note that for every timestamp $h$, there exists $1 \leq \ell_{kh} \leq L$ such that $p_{kh} = p_{k\ell_{kh}}$ and $r_{kh} = r_{k\ell_{kh}}$.

### 4.5.2 Regret decomposition in terms of the Bellman error

The next step is to decompose the regret in terms of the Bellman error as presented by Osband et al. [2013]. We apply the recursive rule to the right-hand side of (4.5) in order to simplify the regret formulation for the policy $\pi_k$ with respect to $M_k$ and the true MDP $M^*$. Denote the actual states visited throughout the $k$th episode with respect to the true MDP $M^*$ by $s_{kh}$ and recall that $a_{kh} = \mu_k(s_{kh}, h)$. We also adapt the shorter notations and denote the state-action pair $(s_{kh}, a_{kh})$ by $x_{kh}$.

$$\left(V_{k,1}^k - V_{k,1}^*\right)(s_{k1})$$

$$= (r_{k1} - r^*)(x_{k1})$$

$$+ \sum_{s'} ((p_{k1} - p^*) (s' \mid x_{k1})) V_{k,2}^k (s')$$

$$+ \sum_{s'} p^* (s' \mid x_{k1}) \left(V_{k,2}^k - V_{k,2}^*\right)(s')$$

$$= (r_{k1} - r^*)(x_{k1}) + ((p_{k1} - p^*) (\cdot \mid x_{k1}))^T V_{k,2}^k$$

$$+ \sum_{s'} p^* (s' \mid x_{k1}) \left(V_{k,2}^k - V_{k,2}^*\right)(s')$$

$$- \left(V_{k,2}^k - V_{k,2}^*\right)(s_{k2}) + \left(V_{k,2}^k - V_{k,2}^*\right)(s_{k2}) \quad (4.6)$$

We define $d_{k1} = \left(\sum_{s'} p^* (s' \mid x_{k1}) \left(V_{k,2}^k - V_{k,2}^*\right)(s')\right) - \left(V_{k,2}^k - V_{k,2}^*\right)(s_{k2})$ and apply the recursive rule on the last term of (4.6) and so on. This yields the following
formulation:

\[
\begin{align*}
&\left( V^k_{k,1} - V^*_{k,1} \right) (s_{k1}) \\
&= \sum_{h=1}^{H} (r_{kh} - r^*) (x_{kh}) \\
&\quad + \sum_{h=1}^{H} \left( (p_{kh} - p^*) (\cdot | x_{kh}) \right)^T V^k_{k,h+1} + \sum_{h=1}^{H} d_{kh}
\end{align*}
\]  

(4.7)

where 

\[
d_{kh} = \left( \sum_{s'} p^* (s' | x_{kh}) \left( V^k_{k,h+1} - V^*_{k,h+1} \right) (s') \right) - \left( V^k_{k,h+1} - V^*_{k,h+1} \right) (s_{kh+1}).
\]

By definition,

\[
E \left[ \left( V^k_{k,h+1} - V^*_{k,h+1} \right) (s_{kh+1}) \right] = \sum_{s'} p^* (s' | x_{kh}) \left( V^k_{k,h+1} - V^*_{k,h+1} \right) (s')
\]

so we can conclude that for 1 ≤ h ≤ H the following holds:

\[
E [d_{kh} | H_k] = 0 \quad (4.8)
\]

Combining (4.7) and (4.8) leads us to the following formulation of the expected regret:

\[
E [\Delta_k | H_k] \leq E \left[ \sum_{h=1}^{H} (r_{kh} - r^*) (x_{kh}) \right] + E \left[ \sum_{h=1}^{H} \left( (p_{kh} - p^*) (\cdot | x_{kh}) \right)^T V^k_{k,h+1} \right]
\]

(4.9)

The expected value is taken with respect to the initial state sampled according to \( \rho \) and the trajectory itself given the history.

In the next two subsections we handle the upper bounds on the two sums in (4.9).

### 4.5.3 An upper bound on the deviations from the rewards

We start with the deviations caused by the rewards. For every timestamp 1 ≤ h ≤ H, we will upper bound \( E \left[ (r_{kh} - r^*) (x_{kh}) \right] \) separately. Recall that there exists some 1 ≤ \( \ell_{kh} \) ≤ \( L \) such that \( r_{kh} = r^* \). Therefore, we will upper bound
Denote the number of times the pair $x \in S \times A$ was visited prior to the $k$th episode by $n_k(x)$. Also denote the number of times prior to episode $k$ that we visited $x$ and obtained a reward of 1 by $n^+_k(x)$ and set $n^-_k(x) = n_k(x_{kh}) - n^+_k(x)$.

Let $E^+_\delta$ be the event that $\forall s \in S, \forall a \in A, \forall k \in \{1, \ldots, K\}$ and $\forall \ell \in \{1, \ldots, L\}$ the following holds:

$$\left| r^\ell_k(x) - F^x(x) \right| \leq \sqrt{\frac{\ln (2/\delta)}{2(3 + n_k(x))}} ,$$

for any $\delta > 0$. Let $E^-_\delta$ denote $E^+_\delta$'s complementary event.

**Lemma 8.** For any $\delta > 0$, $\Pr (E^-_\delta \mid H_k) \leq 2LSA\delta$.

**Proof.** The proof is based on Theorem 1 from [Marchal and Arbel, 2017] and a standard concentration bound for sub-Gaussian random variables. Note that conditioned on the history $H_k$,

$$r^\ell_k(x), F^x(x) \sim \text{Beta} \left(1 + n^+_k(x), 1 + n^-_k(x)\right) .$$

According to the mentioned theorem, conditioned on $H_k$, $r^\ell_k(x)$ and $F^x(x)$ are $\frac{1}{\sqrt{3+n_k(x)}}$-sub-Gaussian.

For a $\sigma^2$-sub-Gaussian random variable $X$ with mean $\mu$, the following standard concentration inequality holds:

$$\Pr (|X - \mu| > \epsilon) \leq 2e^{-\frac{\epsilon^2}{2\sigma^2}} ,$$

for every $\epsilon > 0$. Rearranging the inequality in terms of $\delta$, we get:

$$\Pr (|X - \mu| > \sqrt{2\sigma^2 \ln (2/\delta)}) \leq \delta \quad (4.10)$$

for every $\delta > 0$.

Applying this concentration bound on $r^\ell_k(x)$ yields:

$$\Pr \left(\left| r^\ell_k(x) - F^x(x) \right| > \sqrt{\frac{\ln (2/\delta)}{2(3 + n_k(x))}} \right) \leq 2 \Pr \left( r^\ell_k(x) - \mathbb{E} \left[ r^\ell_k(x) \right] \left| H_k \right\left| H_k \right) \leq 2 \delta \quad (4.11)$$

23
The first inequality uses the triangle inequality and the last uses (4.10) with \( \sigma^2 = \frac{1}{4(3+n_k(x))} \).

We use the union bound over \( S \) states, \( A \) actions and \( L \) sampled MDPs along with (4.11) to obtain the desired result.

Now we can complete the upper bound on the deviations from the rewards.

\[
\mathbb{E} [(r_{kh} - r^*) (x_{kh}) \mid H_k] = \mathbb{E} [(r_{kh}^L - r^*) (x_{kh}) \mid H_k] = \mathbb{E} [(r_{kh}^* - r^*) (x_{kh}) \mid E_{\delta}^T, H_k] \Pr (E_{\delta}^T \mid H_k) + \mathbb{E} [(r_{kh}^L - r^*) (x_{kh}) \mid E_{\delta}^T, H_k] \Pr (E_{\delta}^T \mid H_k) \\
\leq \sqrt{\frac{\ln (2/\delta)}{2 (3+n_k(x))}} + 2LSA\delta \quad (4.12)
\]

The inequality uses Lemma 8 and the assumption that the rewards are bounded in \([0, 1]\).

**4.5.4 An upper bound on the deviations from the transitions**

We move to upper bound the second sum of equation (4.9).

The basic argument for upper bounding the deviations from the transitions is similar to the one we used for the rewards. However, in this case we wish to upper bound an expression involving a Dirichlet random variable rather than a Beta random variable, which requires a preliminary trick.

Analogously to the case of the deviations from the rewards, we apply the equality \( p_{kh} = p_{kh}^L \), so it is sufficient to upper bound \( \mathbb{E} \left[ \left( \left( p_{kh}^L - p^* \right) (\cdot \mid x_{kh}) \right)^T \mathbb{V}_{k,h+1} \right] \mid H_k \) for every \( 1 \leq h \leq H \).

Let \( p_k^{kh} (\cdot \mid x_{kh}) = (p_1, \ldots, p_S) \) and \( p^* (\cdot \mid x_{kh}) = (q_1, \ldots, q_S) \). Note that conditioned on \( H_k \), both \( p_k^{kh} (\cdot \mid x_{kh}) \) and \( p^* (\cdot \mid x_{kh}) \) are distributed according to \( \text{Dir} (1 + \alpha_k^1, \ldots, 1 + \alpha_k^S) \), where \( \alpha_k^s \) counts the number of times we visited \( x_{kh} \) and transitioned to state \( s \) prior to episode \( k \). Clearly, \( \sum_{s'=1}^S p_k^{kh} (s' \mid x_{kh}) = 1 \) and
\[
\sum_{s=1}^{S} p_s^* (s' | x_{kh}) = 1,
\]
thus enabling the following simple manipulation:

\[
\left( \left( p_{kh}^{s_a} - p^* \right) \cdot |x_{kh} \right)^T V_{k,h+1}^{k}
= \sum_{s=1}^{S} (p_s - q_s) V_{k,h+1}^{k} (s)
= \sum_{s=1}^{S-1} (p_s - q_s) V_{k,h+1}^{k} (s) + \left( \left( 1 - \sum_{s=1}^{S-1} p_s \right) - \left( 1 - \sum_{s=1}^{S-1} q_s \right) \right) V_{k,h+1}^{k} (S)
= \sum_{s=1}^{S-1} (p_s - q_s) V_{k,h+1}^{k} (s) - V_{k,h+1}^{k} (S) \sum_{s=1}^{S-1} (p_s - q_s)
= \sum_{s=1}^{S-1} (p_s - q_s) \left( V_{k,h+1}^{k} (s) - V_{k,h+1}^{k} (S) \right)
\leq \left( \sum_{s=1}^{S-1} (p_s - q_s) \right) \max_{s \in \{1, \ldots, S-1\}} \left\{ V_{k,h+1}^{k} (s) - V_{k,h+1}^{k} (S) \right\}
\leq H \left( \sum_{s=1}^{S-1} p_s - \sum_{s=1}^{S-1} q_s \right)
(4.13)
\]

Again, the last inequality uses the assumption that the rewards are bounded between 0 and 1 and therefore \( \max_{s \in \{1, \ldots, S-1\}} \left\{ V_{k,h+1}^{k} (s) - V_{k,h+1}^{k} (S) \right\} \leq H \). This implies that

\[
E \left( \left( \left( p_{kh}^{s_a} - p^* \right) \cdot |x_{kh} \right)^T V_{k,h+1}^{k} \mid \mathcal{H}_k \right) \leq H \cdot E \left[ \sum_{s=1}^{S-1} p_s - \sum_{s=1}^{S-1} q_s \mid \mathcal{H}_k \right]
(4.14)
\]

Note that conditioned on \( \mathcal{H}_k \), \( \sum_{s=1}^{S-1} p_s \) and \( \sum_{s=1}^{S-1} q_s \) are distributed according to Beta \( \left( S - 1, \sum_{s=1}^{S-1} \alpha_k^s, 1 + \alpha_k^S \right) \). Now we can handle the deviations from the transitions according to (4.14) in a similar manner to the way we handled the deviations from the rewards. Let \( E_\delta^p \) be the event that \( \forall s \in S, \forall a \in A, \forall k \in \{1, \ldots, K\} \) and \( \forall \ell \in \{1, \ldots, L\} \) the following holds:

\[
\left| \sum_{s=1}^{S-1} p_s - \sum_{s=1}^{S-1} q_s \right| \leq \sqrt{\frac{\ln (2/\delta)}{2 (1 + S + n_k (x))}}
\]
for any \( \delta > 0 \). Let \( \overline{E}_\delta^p \) denote \( E_\delta^p \)'s complementary event.

**Lemma 9.** For any \( \delta > 0 \), \( \Pr ( \overline{E}_\delta^p \mid \mathcal{H}_k ) \leq 2 LSA \delta \).

**Proof.** The proof is very similar to the proof of lemma \( \text{8} \). According to Theorem 1 in [Marchal and Arbel 2017], conditioned on \( \mathcal{H}_k \), \( \sum_{s=1}^{S-1} p_s \) and \( \sum_{s=1}^{S-1} q_s \) are \( \frac{1}{4(1+S+n_k(x))} \)-sub-Gaussian. Using the same concentration bound for sub-Gaussian
random variables, we get:

\[
\Pr \left( \left| \sum_{s=1}^{S-1} p_s - \sum_{s=1}^{S-1} q_s \right| > \sqrt{\frac{\ln(2/\delta)}{2(1 + S + n_k(x))}} \mathcal{H}_k \right) \\
\leq 2\Pr \left( \left| \sum_{s=1}^{S-1} p_s - \mathbb{E} \left[ \sum_{s=1}^{S-1} p_s \right] \right| > \sqrt{\frac{\ln(2/\delta)}{2(1 + S + n_k(x))}} \mathcal{H}_k \right) \\
\leq 2\delta
\] (4.15)

Once more, we use the union bound over \( S \) states, \( A \) actions and \( L \) sampled MDPs along with (4.15) to obtain the desired result.

Finally, we bound the deviations from the transitions.

\[
\mathbb{E} \left[ ((p_{kh} - p^*) (\cdot | x_{kh}))^T v_{k,h+1}^k | \mathcal{H}_k \right] \\
= \mathbb{E} \left[ ((p_{kh}^e - p^*) (\cdot | x_{kh}))^T v_{k,h+1}^k | \mathcal{H}_k \right] \\
\leq H \cdot \mathbb{E} \left[ \sum_{s=1}^{S-1} p_s - \sum_{s=1}^{S-1} q_s | \mathcal{H}_k \right] \\
= H \cdot \mathbb{E} \left[ \sum_{s=1}^{S-1} p_s - \sum_{s=1}^{S-1} q_s | e_{\delta, k}^p, \mathcal{H}_k \right] \Pr (e_{\delta, k}^p | \mathcal{H}_k) \\
+ H \cdot \mathbb{E} \left[ \sum_{s=1}^{S-1} p_s - \sum_{s=1}^{S-1} q_s | e_{\delta, k}^p, \mathcal{H}_k \right] \Pr (e_{\delta, k}^p | \mathcal{H}_k) \\
\leq H \sqrt{\frac{\ln(2/\delta)}{2(1 + S + n_k(x))}} + 2HLSA\delta
\] (4.16)

### 4.5.5 Complete regret upper bound

We now upper bound (4.9) using (4.12) and (4.16). Recall that the rewards are bounded between 0 and 1, hence the maximal regret suffered by the agent at every timestamp is 1.
\[ \mathbb{E}[\Delta_k \mid \mathcal{H}_k] \leq \sum_{h=1}^{H} \min \left\{ \left( \sqrt{\frac{\ln (2/\delta)}{2 (3 + n_k (x_{kh})} + 2 LSA\delta} \right) + \left( H \sqrt{\frac{\ln (2/\delta)}{2 (1 + S + n_k (x_{kh}))} + 2 HLSA \delta} \right), 1 \right\} \]

\[ \leq \sum_{h=1}^{H} \min \left\{ (H + 1) \left( \sqrt{\frac{\ln (2/\delta)}{2 (1 + n_k (x_{kh}))} + LSA \delta} \right), 1 \right\} \quad (4.17) \]

for every \( \delta > 0 \).

As for the Bayesian regret:

\[ \mathbb{E}\left[ \text{Regret}\left(T, \pi_{OPSRL}, M^*\right)\right] = \mathbb{E} \left[ \sum_{k=1}^{K} \Delta_k \right] = \mathbb{E} \left[ \sum_{k=1}^{K} \mathbb{E}[\Delta_k \mid \mathcal{H}_k] \right] \]

\[ \leq \min \left\{ \mathbb{E} \left[ \sum_{k=1}^{K} \sum_{h=1}^{H} \min \left\{ (H + 1) \left( \sqrt{\frac{\ln (2/\delta)}{2 (1 + n_k (x_{kh}))} + 2 LSA \delta} \right), 1 \right\} \right], T \right\} \]

\[ \leq \min \left\{ \mathbb{E} \left[ \sum_{k=1}^{K} \sum_{h=1}^{H} \min \left\{ 2H \sqrt{\frac{\ln (2/\delta)}{2 (1 + n_k (x_{kh}))}, 1 \right\} \right], T \right\} \]

\[ \leq \min \left\{ \mathbb{E} \left[ \sum_{k=1}^{K} \sum_{h=1}^{H} \min \left\{ 2H \sqrt{\frac{\ln (2/\delta)}{2 (1 + n_k (x_{kh}))}, 1 \right\} \right] + 4H^2 KLSA \delta, T \right\} \quad (4.18) \]

The second equality uses the law of total expectation and the first inequality uses (4.17).

The next step is to upper bound the following term

\[ \sum_{k=1}^{K} \sum_{h=1}^{H} \min \left\{ 2H \sqrt{\frac{\ln (2/\delta)}{2 (1 + n_k (x_{kh}))}, 1 \right\} \]

from (4.18). We follow the calculations from Osband et al. [2013], separate this
term according to the event \( \{ n_k(x_{kh}) \leq H \} \) and upper bound each of them:

\[
\begin{align*}
& \sum_{k=1}^{K} \sum_{h=1}^{H} \left( \min \left\{ 2H \sqrt{\frac{\ln (2/\delta)}{2(1 + n_k(x_{kh}))}}, 1 \right\} \right) \\
& \leq \sum_{k=1}^{K} \sum_{h=1}^{H} \mathbb{I}_{\{ n_k(x_{kh}) \leq H \}} \\
& + \sum_{k=1}^{K} \sum_{h=1}^{H} \mathbb{I}_{\{ n_k(x_{kh}) > H \}} 2H \sqrt{\frac{\ln (2/\delta)}{2(1 + n_k(x_{kh}))}} 
\end{align*}
\]  

(4.19)

Note that for every state-action pair \( x \in S \times A \), the event \( \{ n_k(x_{kh}) \leq H \} \) can occur no more than \( 2H \) times. Applying the union bound for every state and action, results in the following upper bound on the first sum in the right-hand side of (4.19):

\[
\sum_{k=1}^{K} \sum_{h=1}^{H} \mathbb{I}_{\{ n_k(x_{kh}) \leq H \}} \leq 2HSA 
\]  

(4.20)

With a slight abuse of notation, let \( n_{kh}(x) \) denote the number of times the agent visited state-action pair \( x \) prior to timestamp \( h \) of episode \( k \). If \( n_k(x) > H \), then for every \( 1 \leq h \leq H \) the following holds: \( 1 + n_{kh}(x) \leq 1 + n_k(x) + H \leq 2(1 + n_k(x)) \). Thus,

\[
\sum_{k=1}^{K} \sum_{h=1}^{H} \mathbb{I}_{\{ n_k(x_{kh}) > H \}} \sqrt{\frac{1}{2(1 + n_k(x_{kh}))}} \leq \sum_{k=1}^{K} \sum_{h=1}^{H} \sqrt{\frac{1}{1 + n_{kh}(x_{kh})}} 
\]  

(4.21)

Every state-action pair \( x \) was visited \( n_{K+1}(x) \) times. We can rearrange the sums
from (4.21) by the state-action pairs:

\[
\sum_{k=1}^{K} \sum_{h=1}^{H} \sqrt{\frac{1}{1 + n_{kh} (x_{kh})}}
\]

\[
= \sum_{x \in S \times A} \sum_{j=1}^{n_{K+1}(x)} j^{-1/2}
\]

\[
\leq \sum_{x \in S \times A} \int_{0}^{n_{K+1}(x)} z^{-1/2} \, dz
\]

\[
= \sum_{x \in S \times A} 2 \sqrt{n_{K+1}(x)}
\]

\[
\leq 2 \sqrt{SA} \sum_{x \in S \times A} n_{T+1}(x)
\]

\[
= 2 \sqrt{SA T}
\]

The last inequality uses Jensen inequality. This completes the upper bound of the second sum in (4.19).

Combining the upper bounds we formulated in (4.20) and (4.22) completes the upper bound of (4.18). To conclude our calculations, we choose \( \delta = \frac{1}{LSAT} \).

\[
E \left[ \text{Regret} \left( T, \pi^{ROPS}, M^* \right) \right]
\]

\[
\leq \min \left\{ \sum_{k=1}^{K} \sum_{h=1}^{H} \min \left\{ 2H \sqrt{\frac{\ln (2/\delta)}{2 (1 + n_{k} (x_{kh}))}}, 1 \right\} + 4H^2 KLSA \delta, T \right\}
\]

\[
\leq \min \left\{ 2HSA + 4H \sqrt{SAT \ln (2/\delta)} + 4H^2 KLSA \delta, T \right\}
\]

\[
\leq \min \left\{ 2HSA + 4H \sqrt{SAT \ln (2LSAT)} + 4 \frac{H^2 K}{T}, T \right\}
\]

(4.23)

Under the mild assumption that the learning problem is not degenerate and \( T \geq H \), we can write \( T \leq KH \leq T + H \leq 2T \), which implies that (4.23) can be upper bounded by:

\[
E \left[ \text{Regret} \left( T, \pi^{ROPS}, M^* \right) \right]
\]

\[
\leq \min \left\{ 2HSA + 4H \sqrt{SAT \ln (2LSAT)} + 8H, T \right\}
\]

\[
\leq \min \left\{ 10HSA + 4H \sqrt{SAT \ln (2LSAT)}, T \right\}
\]

\[
\leq 14H \sqrt{SAT \ln (2LSAT)}
\]

(4.24)

Equation (4.24) completes the proof of Theorem 4.
Chapter 5

Simulations

In this chapter we compare the performance of PSCS and ROPS with state-of-the-art algorithms.

We study the performance of our proposed algorithms on two different settings:

1. A random generated MDP in accordance to the conditions specified in Theorem [1].
2. FrozenLake, a standard benchmark from OpenAI gym (Brockman et al. [2016]).

5.1 Random MDP

Setup

Based on our Bayesian setting, we generate a random MDP consisted of 20 states and 10 actions. The transitions are sampled from Dir (1,...,1) and the rewards are sampled independently from Beta (1, 1). We set the horizon \( H \) is to 40 and assume that the agent always starts at state \( s = 1 \), that is \( \rho = (1, 0, \ldots, 0) \).

To maintain a fair comparison, the transitions and the rewards are sampled in advance, such that for every trial, at the \( n \)th visit to the state-action pair \((s, a)\) all the algorithms will receive the same reward before transitioning to the same state.

Compared algorithms

We compare the following algorithms:

- UCFH from [Dann and Brunskill 2015], an optimized version of UCRL2 for finite-horizon MDPs.
• PSRL from Osband and Van Roy [2016].
• Our proposed PSCS with various values of $L$.
• Our proposed ROPS with various values of $L$.

Recall that for the choice of $L = 1$, ROPS and PSCS are identical to PSRL.

For UCFH, we follow the experimental setting from Osband and Van Roy [2016] and set $\delta = 0.05$ and $\epsilon = 0.1$.

Performance

We run 1,000 independent trials for each tested algorithm. We log the empirical cumulative regret throughout the trials and the value function of the MDP each algorithm choose at every episode.

We focus on the following measurements:

• The empirical average cumulative regret.
• The standard deviation of the empirical cumulative regret.
• The 90th, 95th and 99th percentiles of the empirical cumulative regret.
• The average value function of the MDP each algorithm choose at every episode.

5.1.1 Performance Comparison

Empirical average cumulative regret

We start by comparing the empirical average cumulative regret of the algorithms for 10,000 episodes. Since the algorithms’ performance may vary, PSRL’s performance is used as a baseline in every graph (blue line).

Full results of all compared algorithms can be found in Table 5.1.
Figure 5.1: Average cumulative regret of PSRL, PSCS(L), ROPS(L) and UCFH over 1,000 trials and 10,000 episodes. We choose $L \in \{2, 3, 4, 5\}$. 
Figure 5.1 shows that PSRL outperforms PSCS and ROPS as for their average cumulative regret, but all of them are significantly better than UCFH. Recall that the size of the confidence sets constructed by PSCS and ROPS is exponential with $L$. Therefore, even for small values of $L$, the size of their confidence sets is very large, yet it is not infinitely large as used by UCFH. Both PSCS and ROPS present results which are much more similar to those presented by PSRL rather than by UCFH.

In addition, after 10,000 episodes, it is clear that PSCS’s and ROPS’s average cumulative regret grows monotonically with $L$.

These results also demonstrate that PSCS’s increase of the average cumulative regret is more moderate than ROPS’s. This behaviour is expected due to the size of the set of plausible MDPs $M_k$ each of the algorithms set up before constructing the confidence set $C_k$. In PSCS, the size of $M_k$ ranges between 1 and $L$ and is dependent on the first sampled MDP, whereas in ROPS there are exactly $L$ sampled MDPs in $M_k$ every episode. In some sense, PSCS offers a middle ground between different $L$ values of ROPS. However, this comes at the price of higher variance and less stable performance as we previously mentioned. We revisit this notion in Chapter 6 and focus on ROPS’s performance for the rest of this chapter.

**Empirical cumulative regret standard deviation**

Next, we compare the average cumulative regret and the standard deviation for ROPS with various values of $L$.

Figure 5.2 displays a clear picture regarding the connection between the average cumulative regret and the standard deviation: as $L$ grows, the agent suffers from a slight increase of the average cumulative regret, but benefits from a decrease of the standard deviation. The results show that even for $L = 2$, the standard deviation is significantly improved, whereas the average cumulative reward remains about the same comparing to PSRL. Consistent variance reduction is achieved for larger values of $L$ as well. This allows the user to balance between these two measurements with respect to her computational power and performance requirements.

This experiment arises an interesting theoretical question on whether the decrease of the standard deviation is always monotone, monotone under some conditions or inconclusive. This graph of course does not answer this question, as we are uncertain whether the fact that it is not monotone is due to a property of the algorithm, a property of the problem or simply a result of an insufficient number of trials. We revisit this notion as well in Chapter 6.
Figure 5.2: PSRL and ROPS(L) average-standard deviation trade-off after 10,000 episodes for 1,000 trials. We choose $L \in \{2, 3, 4, 5, 10, 15, 20, 25, 30, 35, 40\}$.

**Empirical cumulative regret quantiles**

Our algorithms also improve the quantiles of the average cumulative regret. Before we present a comparison between the quantiles of PSRL and ROPS with various values of $L$, we would like to show an example of a comparison between PSRL’s and ROPS’s histograms of the average cumulative regret after 10,000 episodes. For ROPS, we choose $L = 5$. 

34
Figure 5.3: PSRL and ROPS(5) histograms after 10,000 episodes for 1,000 trials.

Figure 5.3 illustrates the two sides of the coin. On one hand, the main mass of PSRL’s cumulative regret results after 10,000 episodes for 1,000 trials is located left of that presented by ROPS(5), thus leading to a lower average cumulative regret. On the other hand, PSRL has a heavier tail than ROPS(5), which implies that it suffers from higher standard deviation and also shows inferior performance as for different quantiles compared to ROPS(5). The following graph presents the 90th, 95th and 99th percentiles after 10,000 episodes for 1,000 trials for PSRL and ROPS with various values of $L$. 
Figure 5.4 shows PSRL’s quantiles at $L = 1$ and ROPS at the rest of $L$’s values. From a general perspective, all three percentiles behave similarly: they decrease before reaching a minimal point and then start increasing. This representation sheds more light on the trade-off between the average cumulative regret and the concentration of the performance of the algorithms. Even though the standard deviation decreases overall as $L$ grows, at some point the increase of the average cumulative regret makes it not worthwhile. The quantiles representation allows the user to choose values of $L$ which are more suitable for her performance requirements. For example, if the user’s most important aspect of performance is stability, she should choose the greatest $L$ value possible within her computational resources. On a different scenario, if she is more interested in minimizing the performance of 95% of the trials, she may prefer $L = 4$.

It is interesting to notice that ROPS outperforms PSRL with respect to the 99th percentiles for all tested $L$ values. Also note that the 99th percentile is not as smooth as the 90th and the 95th percentiles due to the number of trials. In our experiments setting, the 99th percentile is determined by the 10th highest cumulative regret value among 1,000 trials. It is fair to assume that by running more trials, we would
get a more smooth plot, similar to these obtained for the other percentiles.

The trade-off between the average cumulative regret and the above percentiles resembles a behaviour usually associated with robust algorithms. The user agrees to settle for a higher average cumulative regret in return for a guarantee that the threshold for which a certain percentage of the trials pass is minimized. This is an unusual property for optimistic algorithms.
5.1.2 Optimism Comparison

We wish to empirically justify that ROPS is indeed worthy of its title. Figure 5.5 presents a comparison between $V^*_k$, the true value function of the random MDP we sampled, and $V^k_{k,1}$, the value function used in every episode by each algorithm averaged over 1,000 trials. Again, we use PSRL as a baseline.

![PSRL vs. UCFH](image1)

![PSRL vs. ROPS](image2)

Figure 5.5: Average value function comparison of PSRL, ROPS(L) and UCFH over 1,000 trials and 10,000 episodes. We choose $L \in \{2, 3, 4, 5, 10, 15, 20\}$. 

38
The upper graph demonstrates that UCFH is overly optimistic, which results in inferior performance. Considering the assumption that the rewards are bounded between 0 and 1, the maximum value function possible for the true MDP is $H$, which is 40 in our experiment setting. Although UCFH is aware of this assumption, it still acts according to MDPs which have a very high value function and it takes about 2,000 episodes for it to drop below 50, which is still very high.

Also, according to the upper graph it is almost impossible to distinguish between the true value function and PSRL’s value function and one might wrongly assume that they are the same. The lower graph clearly shows that on average PSRL samples MDPs that are pessimistic compared to the true MDP $M^*$, before conjoining with the true value function. This result supports our claim that PSRL is not an optimistic algorithm. In accordance to our previous argument, ROPS presents moderate optimistic value functions. The user can control the measure of optimism by setting the regularization parameter and balance between the average cumulative regret and the standard deviation.

5.2 FrozenLake

Setup

We follow the framework called FrozenLake, a standard benchmark from OpenAI gym, with a few modifications.

In FrozenLake, an agent is placed on a grid world, trying to get from a starting point to a target point by moving in four possible directions (up, down, left and right). Some of the tiles are holes and reaching any one of them results in failure. The rest of the tiles are slippery and the actual movement direction is uncertain and not fully determined by the action taken by the agent. Figure 5.6 presents the grid world as used by the larger version of this framework which is referred to as FrozenLake8x8. This grid world is consisted of 64 states and the agent may choose one out of four actions from each state. $S$ denotes the starting point, $G$ denotes the goal, $H$ denotes holes and $F$ denotes safe but slippery tiles.
We made one small modification to the problem definition in order to match this setting to our finite-horizon framework. In the original setting, after the agent either reaches a hole or the goal tile, she goes straight back to the starting point. This results in uneven episodes. In order to mitigate this issue, we simply modified the transitions from the holes and from the goal tile such that all actions result in the agent staying at the corresponding tile until the number of steps in the episode reaches the horizon $H$. This small adjustment guarantees fixed length episodes, which is critical property for our framework.

As for the rewards, according to the FrozenLake setting, the agent receives a reward of 1 after transitioning into the goal tile for the first time within an episode. The rest of the transitions does not grant any reward, i.e. a reward of 0. This implies that the algorithms must be altered accordingly. Note that in contrast to our original definition of the value function in $(3.1)$, in FrozenLake, the reward function depends not only on the current state and the action the agent chooses, but also on the state the agent transitions into. This distinction requires a more general definition of the value function:

$$V_{\mu,h}^M (s) = \mathbb{E} \left[ \sum_{j=1}^{H} r(s_{1j}, \mu(s_{1j}, j), s_{1j+1}) | s_{1h} = s \right]$$

where $r(s, a, s')$ is the reward obtained by the agent after choosing an action $a$ while at state $s$ and transitioning into $s'$. 

Figure 5.6: FrozenLake8x8 from OpenAI gym.
Note that this setting resembles real-world applications better than the random MDP setting. It is more common to encounter scenarios in which the reward is sparse and is obtained only after the task is completed, which makes the learning assignment harder.

Consistent with our previous framework, we sampled the transitions and the rewards to maintain a fair comparison. We set the horizon $H$ to 40.

**Compared algorithms**

We compare the following algorithms:

- PSRL from Osband and Van Roy [2016].
- Our proposed ROPS with various values of $L$.

We slightly modified the algorithms to match the more general version of the value function as presented in [5.1]. We emphasize that the algorithms are unaware of the structure of the transitions, i.e. they do not assume that the agent can move only to adjacent tiles or to remain in the current tile. Instead, they assume that the agent might move from every state $s$ to every other state $s'$, including $s$ itself.

**Performance**

Due to long running times, we run 100 independent trials for each tested algorithm. We log the empirical cumulative regret after each episode for every trial.

We focus on the following measurements:

- The empirical average cumulative regret.
- The standard deviation of the empirical cumulative regret.
- The 90th and the 95th percentiles of the empirical cumulative regret.

**5.2.1 Performance comparison**

**Empirical average cumulative regret**

We start once more by comparing the empirical average cumulative regret of the algorithms for 50,000 episodes.

The full results of all compared algorithms can be found in Table 5.2.
Figure 5.7: Average cumulative regret of PSRL and ROPS(L) over 100 trials and 50,000 episodes. We choose $L \in \{2, 3, 4\}$.

The results in Figure 5.7 show that ROPS improves on the average cumulative reward compared to PSRL. Though this decrease in the average cumulative reward is not monotone, it is not negligible. This outcome demonstrate once more that ROPS is competitive with PSRL and even outperforms it for the tested values of $L$ in a real-world application.

As we show next, this improvement does not have a negative impact on the expected behaviour of the standard deviation.

**Empirical cumulative regret standard deviation**

Figure 5.8 compares the average cumulative regret and the standard deviation of PSRL and ROPS together as done earlier in Figure 5.2. In this setting, the user does not have to settle for a higher average cumulative regret, as she can benefit from both a decrease in the average cumulative regret and the standard deviation. Clearly, this result is not sustainable for large values of $L$, but for small values of $L$ it is preferable to choose ROPS over PSRL. In this example, the decrease in the standard deviation is significant and in contrast to the random MDP experiment, it only costs additional computing resources.
Figure 5.8: PSRL and ROPS(L) average-standard deviation trade-off after 50,000 episodes for 100 trials. We choose $L \in \{2, 3, 4\}$.

**Empirical cumulative regret quantiles**

Our final comparison deals with PSRL and ROPS’s quantiles. Since we only ran 100 trials in this experiment, we present results of the 90th and the 95th percentiles and give up the 99th percentile since it is not based on enough samples and could lead to incorrect conclusions. Nevertheless, the results in Figure 5.9 is very similar to these we presented in Figure 5.4. ROPS outperforms PSRL for all the tested $L$ values. The user can benefit from a decrease in these percentiles only in exchange for additional computation cost, which is an excellent result.
Figure 5.9: PSRL and ROPS 90th, 95th and 99th percentiles after 50,000 episodes for 100 trials. We choose $L \in \{2, 3, 4\}$.
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average Cumulative Regret</th>
<th>Standard Deviation</th>
<th>90% percentile</th>
<th>95% percentile</th>
<th>99% percentile</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSRL</td>
<td>921.3</td>
<td>153.3</td>
<td>1084.3</td>
<td>1181.5</td>
<td>1574.2</td>
</tr>
<tr>
<td>ROPS(2)</td>
<td>922.6</td>
<td>127.7</td>
<td>1056.4</td>
<td>1121.6</td>
<td>1426.0</td>
</tr>
<tr>
<td>ROPS(3)</td>
<td>937.2</td>
<td>115.5</td>
<td>1045.5</td>
<td>1110.1</td>
<td>1317.8</td>
</tr>
<tr>
<td>ROPS(4)</td>
<td>950.7</td>
<td>110.0</td>
<td>1053.5</td>
<td>1109.2</td>
<td>1339.1</td>
</tr>
<tr>
<td>ROPS(5)</td>
<td>963.6</td>
<td>98.8</td>
<td>1065.3</td>
<td>1116.6</td>
<td>1378.0</td>
</tr>
<tr>
<td>ROPS(10)</td>
<td>1010.1</td>
<td>87.7</td>
<td>1095.1</td>
<td>1141.0</td>
<td>1296.3</td>
</tr>
<tr>
<td>ROPS(15)</td>
<td>1041.1</td>
<td>74.3</td>
<td>1122.4</td>
<td>1155.0</td>
<td>1268.6</td>
</tr>
<tr>
<td>ROPS(20)</td>
<td>1065.3</td>
<td>74.4</td>
<td>1148.7</td>
<td>1177.8</td>
<td>1280.5</td>
</tr>
<tr>
<td>ROPS(25)</td>
<td>1086.8</td>
<td>79.8</td>
<td>1168.0</td>
<td>1197.2</td>
<td>1294.9</td>
</tr>
<tr>
<td>ROPS(30)</td>
<td>1101.8</td>
<td>68.3</td>
<td>1187.6</td>
<td>1214.1</td>
<td>1279.2</td>
</tr>
<tr>
<td>ROPS(35)</td>
<td>1116.4</td>
<td>72.3</td>
<td>1197.7</td>
<td>1230.9</td>
<td>1301.3</td>
</tr>
<tr>
<td>ROPS(40)</td>
<td>1128.9</td>
<td>67.4</td>
<td>1209.5</td>
<td>1237.2</td>
<td>1305.9</td>
</tr>
<tr>
<td>PSCS(2)</td>
<td>924.1</td>
<td>146.5</td>
<td>1076.0</td>
<td>1152.4</td>
<td>1547.7</td>
</tr>
<tr>
<td>PSCS(3)</td>
<td>931.6</td>
<td>147.5</td>
<td>1075.4</td>
<td>1151.3</td>
<td>1442.2</td>
</tr>
<tr>
<td>PSCS(4)</td>
<td>936.8</td>
<td>128.6</td>
<td>1052.6</td>
<td>1124.8</td>
<td>1437.0</td>
</tr>
<tr>
<td>PSCS(5)</td>
<td>938.7</td>
<td>107.5</td>
<td>1049.2</td>
<td>1110.6</td>
<td>1317.8</td>
</tr>
<tr>
<td>UCFH</td>
<td>26495.2</td>
<td>184.0</td>
<td>26730.5</td>
<td>26798.2</td>
<td>26897.0</td>
</tr>
</tbody>
</table>

Table 5.1: Comparison of the average cumulative regret, standard deviation and 90th, 95th and 99th percentiles for PSRL, ROPS(L), PSCS(L) and UCFH over 1,000 trials after 10,000 episodes.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average Cumulative Regret</th>
<th>Standard Deviation</th>
<th>90% percentile</th>
<th>95% percentile</th>
<th>99% percentile</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSRL</td>
<td>2861.5</td>
<td>41.8</td>
<td>2913.6</td>
<td>2922.1</td>
<td>2974.8</td>
</tr>
<tr>
<td>ROPS(2)</td>
<td>2846.2</td>
<td>38.5</td>
<td>2895.3</td>
<td>2903.5</td>
<td>2930.0</td>
</tr>
<tr>
<td>ROPS(3)</td>
<td>2840.0</td>
<td>38.8</td>
<td>2885.6</td>
<td>2904.6</td>
<td>2942.9</td>
</tr>
<tr>
<td>ROPS(4)</td>
<td>2842.3</td>
<td>36.8</td>
<td>2884.2</td>
<td>2900.4</td>
<td>2914.9</td>
</tr>
</tbody>
</table>

Table 5.2: Comparison of the average cumulative regret, standard deviation and 90th, 95th and 99th percentiles for PSRL and ROPS(L) over 100 trials after 50,000 episodes.
Chapter 6

Discussion

We introduced the principle of Restricted Optimism for Reinforcement Learning and analysed the two main approaches for solving finite-horizon MDPs. Although the OFU principle is very elegant and leads to theoretically efficient algorithms, if overdone, it could lead to significantly inferior performance comparing to other methods such as Posterior Sampling. On the other hand, we claimed that Bayesian methods can benefit from optimism. Our work focused on constructing smaller, yet rich enough confidence sets, thus restricting and even controlling the level of optimism carried out by the algorithms. Inspired by previous works, we adopted Posterior Sampling as the main instrument for constructing these confidence sets.

We presented two algorithms in the spirit of our Restricted Optimism principle. First, we described Posterior Sampled Confidence Sets for Reinforcement Learning (PSCS), an analogous version to OFU algorithms using Posterior Sampling. Then, we proposed Restricted Optimistic Posterior Sampling for Reinforcement Learning (ROPS), a simpler variant of PSCS, and provided an upper bound on its expected cumulative regret similar to the one provided for PSRL.

We concluded with simulations on two different frameworks. These experiments demonstrated that there exists a trade-off between the average cumulative regret and its standard deviation, which can be controlled by the level of optimism of the algorithm. In addition, our proposed algorithms show significant improvement in terms of the quantiles of the average cumulative regret, a property which is usually associated with robust algorithms.

For future work we see two main interesting directions: (1) formulate the trade-off between the average cumulative regret and the standard deviation, and (2) come up with variants of the algorithms presented, for which the level of optimism can be even more fine tuned, improving on PSCS.
Bibliography


התורמות העיקריות של עבודה זו כוללות:

- הצגת עיקרון חדש ניב דמויות שלא אופטימיות מוגבלות ב setDefaultCloseOperation עבורי ייעול של תהליך ההלטה
- מורכבים בעל אופק סופי
- תיאור של שני אלגוריתמים המבוססים על עיקרון האופטימיות המוגבלת למגבלות ולכוון הולכת של
- ת습니다 תיאורים על תוחלת הזרת המפרожет בודמה של אלגוריתמים דומים
- הדגמה באימון ניסויים שיקים קרש הדום בינ מзыва הזרת המפרожет בין סטיית
- התחק, הוא ישים להשיפה על הקישור הזה באימון ניסויים על רמה האופטימיות של
- האלגוריתמים.
אופטימי, אך פחות אופטיים מאשר שריד נבחר במקור, ובכך ניתן לה nulla את האופטימיות של האלגוריתם. לשם כך, אנו מציעים לנהל את הרווח של החופות וביתוחי עלילות, ומציינים כי לאפשרות המבוססת על אופטימיות ثنائية שיפורים המתודיים המושרי לע מצב כל반 בקובץ ביטחון סופי או(UnityEngine

לאלה-

ל yakın האופטימיות המבוססות על אופטימיות מוגבלות, נﺷקף גם תהליך החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, בצウォー הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, בצウォー הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, בצウォー הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, בצウォー הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, בצウォー הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, בצウォー הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, בצウォー הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, בצウォー הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, בצウォー הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, בצウォー הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, בצウォー הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, בצウォー הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, בצウォー הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המרחוקים בעין רך, צbrero הרווח החילות מרחוקים ביניים. עבור כל פרק, מזין השולח את התהליכים המר

ii

Technion - Computer Science Department - M.Sc. Thesis  MSC-2018-02 - 2018
תקציר

בעיות_DECISION_SEQplaylistוב分かるEmittertega נתונים כדי להחלטות את העיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות שבהן ניתן לנקף את הסכנתה של התוכן שיתון ממדיך. בעיות DECISION_SEQplaylist ובין חקירות perchè הוחזק הפשרה של הሰט פוליציים יותר בלתי זמ المقبل, قامت בנוסף, בכיבוש קבוצת ותור תגרה בחרת לשתי התפקידים במקביל.

השכון בחשד נספה גם היא מהשמת הכצבים. תרומת הבסיסים זו לחתוך התפרצנות פיסטרוור לכל פרמטרה של פלטונר devук Middleton והסליקוסיסים, ימיות אוסטריקוסיס ובו מערך זה ל请联系 devuk Middleton והסליקוסיסים, ימיות אוסטריקוסיס ובו מערך זה LALTONетесь ל pstmtסוסיסים. מהאריות של והמתקנים המופרדים לעבר התפקידים, דוגמת מציאת התוכן בכסה ל-LALTON下面是小וסוסיסים.
המ훗ך נעשע בבחינת פורום יש מנור בפקולטה למדעי המחשב.

אני רוחה להודות לאשטי, הילה, על התמיכה הרב, שבってくれתי והזינו לי את הראור.

אני מודה לטכניון על התמיכתה הדידית במשתלמתי.
אופטימיות מוגבלת

חיבור על מחקר

לשם מועילה חלקי של הדרישות לקבלת התואר
מגיסטר למדעי מחשב

שמייה כהן

הוגש לסטטט הטכנית – מכון טכנולוגי לישראל
השינה החשע"ת, חיפה
נובמבר 2017
אופטימיות מוגבלת

שניר כהן