Algorithms for Satisficing in Reinforcement Learning

Thomas Michel
Research internship supervised by Ronald Ortner

January 15, 2024

Abstract

In this internship report we consider the objective of *satisficing* in reinforcement learning (RL) problems. Instead of aiming to find an optimal strategy (policy), the learner is content with a policy whose average reward is above a given satisfaction level. We study this objective under two RL settings: multi-armed bandits and Markov decision processes (MDP). The greater part of this report consider the multi-armed bandit setting. We provide algorithms and analysis for the realizable case when such a satisficing policy exists as well as for the general case when this may not be the case. Introducing the notion of *satisficing regret*, our main result shows that in the general case it is possible to obtain constant satisficing regret when there is a satisficing arm (thereby correcting a contrary claim in literature), while standard logarithmic regret bounds can be re-established otherwise. Experiments illustrate that our algorithm is not only superior to standard algorithms in the satisficing setting, but maybe surprisingly also competitive in the classic bandit setting. We then consider the MDP setting and propose a metric for the evaluation of satisficing algorithms. We analyze an algorithm extending our results on multi-armed bandits to MDPs with deterministic transitions, and provide preliminary considerations toward the general case.

Contents

1 Introduction

Reinforcement learning(RL) is the training of machine learning models to make a sequence of decisions. The agent learns to achieve a goal in an uncertain and potentially complex environment. The usual case of use for reinforcement learning is the following setting: an agent is able to observe its environment and chooses an action to perform from a given set. The agent then receives a reward based on the consequences of its action on the environment, the new state of the environment can be observed and the cycle starts again. A reinforcement learning agent employs trial and error in order to learn about its environment and come up with an optimal sequence of actions, a policy, to maximize its reward.

In particular, we are interested here in the setting in which we want to maximize the total reward collected from the agent since the beginning of its training. In this setting the exploration of the environment has a cost since we miss potential rewards, so one may want to focus on promising actions that lead to high rewards. However, the agent should be careful to not leave aside other actions at the risk of missing out a potential optimal solution to the problem. The previous sentences illustrate the exploration versus exploitation dilemma, which is a central aspect in the design of reinforcement learning algorithms.

The concept of *satisficing* is related to the notion of bounded rationality which was introduced by Simon [1956] as a more realistic model to understand animal behaviors, in particular human ones. As our capacity to analyze a situation is limited and every action we make has a cost, either in time or energy, it may be unreasonable to try to optimize our solution to each complex tasks. This model introduces a satisfaction level above which an individual will be satisfied, will consider the situation to be sufficient and cease to look for a better solution, thus avoiding additional costs related to the exploration of other actions.

One of the reasons why reinforcement learning is in general difficult is that finding an *optimal* policy requires a lot of exploration. This is especially true with the increasing scale and complexity of the environments to which we want to apply RL, such as worldwide recommendation systems or autonomous vehicle driving. In practice, however, we do not need to find the very optimal solution to a problem, and we are often happy to perform a task just good enough. For example, when driving to work we will be content with a strategy that will let us arrive just in time and safely, while the computation of a policy that is 'optimal' in some sense (e.g., along the shortest route, or as fast as possible) may be prohibitive. Accordingly, it is to be expected that when considering a *satisficing* objective aiming to find a solution that is above a certain satisfaction level it is possible to learn a respective policy much faster.

Multiple models have been developed in order to model the environment in which an RL agent evolves. In this study, we consider two different models. The first one is the multi-armed bandit (MAB) setting. The name of this model refer to the slot machines (also called one-armed bandits) in a casino. The agent is the player who want to maximize its gains while playing the machines infinitely many times (in our case). Each machine is associated to a certain reward distribution unknown to the player. At each step, the player can choose to pull the arm of one of the machine and observe the gain (or loss) from its action. This setting has been intensively studied and extended through the years since, despite its simplicity, it can be applied to a wide range of real world problems. The second model we consider is the Markov decision process (MDP) setting. This model is more general than MABs in the sense that each action can now change the state of the environment according to stochastic transition rules (unknown to the agent) and thus the results of subsequent choices. This setting is very powerful and became a standard model for most of the environments of RL problems.

While there are some connections to multi-criterion RL [Roijers et al., 2013], there is hardly any literature on satisficing in RL, with a few exceptions for the MAB setting. Kohno and Takahashi [2017] and Tamatsukuri and Takahashi [2019] propose simple index policies, which are experimentally evaluated. Tamatsukuri and Takahashi [2019] also show that the suggested algorithm converges to a satisficing arm and that the regret is finite if the satisfaction level is chosen to be between the reward of the best and the second-best arm. Reverdy et al. [2017] consider a more general Bayesian setting, which also considers the learner's belief that some arm is satisficing. The notion of *expected satisficing regret* is introduced that measures the loss over all steps where a non-satisficing arm is chosen and the learner's degree of belief in the chosen arm was below some level $\delta \in [0, 1]$. For $\delta = 0$ this coincides with our notion of *satisficing regret* introduced below. Reverdy et al. [2017] present various bounds on the expected satisficing regret, including lower bounds as well as upper bounds for problems with Gaussian reward distributions when using adaptations of the UCL algorithm [Reverdy et al., 2014]. The given bounds for the case $\delta = 0$ that correspond to our setting will be discussed later.

Also related to our paper, Kano et al. [2019] consider the problem of identifying *all* arms above a given satisfaction level and derive sample complexity bounds for the pure-exploration setting with fixed confidence. Related sample complexity bounds can be found in Mason et al. [2020] for identification of all ε-good arms. Closer to our setting is the problem of identifying an arbitrary arm among the top m arms, for which sample complexity bounds are derived by Chaudhuri and Kalyanakrishnan [2017]. A follow-up paper [Chaudhuri and Kalyanakrishnan, 2019] considers the sample complexity of the more general problem of identification of any k of the best m arms. None of these latter investigations however considers the online learning setting with regret as performance measure as we do.

Investigating also the MAB setting, in this paper we introduce the notion of *satisficing regret* that measures the loss with respect to a given satisfaction level S. We first consider the realizable case, where this level can be satisfied. In this setting, quite a simple algorithm can be shown to have constant satisficing regret (i.e., no dependence on the horizon T). For the general setting we provide an algorithm that is able to extend on this result, giving constant satisficing regret in the realizable case, while obtaining logarithmic bounds on the ordinary regret with respect to the optimal arm as for classic MAB algorithms such as UCB1 [Auer et al., 2002]. Experiments not only confirm our theoretical findings but also show that our algorithm is competitive even in the standard setting. In the second part of this report, we present some preliminary results for *satisficing* in the more general setting of Markov decision processes.

2 Satisficing for Multi-armed Bandits

2.1 Setting

We consider the standard multi-armed bandit (MAB) setting with a set of K arms given, in the following denoted as $[\![1, K]\!] := \{1, 2, \ldots, K\}$. In discrete time steps $t = 1, 2, \ldots$ the learner picks an arm $A_t = i$ from $[\![1, K]\!]$ and observes a random reward r_t drawn from a fixed reward distribution specific to the chosen arm i with mean μ_i. In the following we assume that the reward distributions for each arm are sub-Gaussian. This is e.g. guaranteed when the reward distributions are bounded, which is a common assumption in the bandit setting. Additionally, we assume that the reward distributions are 1-sub-Gaussian for the sake of readability of the proofs. In particular, this assumption holds for random variables bounded to the interval $[0, 1]$.

The usual performance measure for a learning algorithm in the MAB setting is the *(pseudo-)regret* after T steps, defined as

$$R_T := \sum_{t=1}^{T} \big(\mu_* - \mu_{A_t}\big),$$

where $\mu_* := \max_i \mu_i$ is the maximal mean reward of all arms. The regret can be interpreted as the average loss in earning by using the algorithm instead of only choosing the optimal action (which is unknown) at each step. Another way to see it is as the cost incurred by the choice of a suboptimal action over the optimal one.

In the satisficing setting however, we only care about whether an arm with mean reward $\geqslant S$ is chosen, where S is the level of satisfaction we aim at. Accordingly, we modify the classic notion of regret and consider what we call the *satisficing (pseudo-)regret* with respect to S (short *S-regret*) defined as

$$R_T^S := \sum_{t=1}^{T} \max\big\{S - \mu_{A_t}, 0\big\}.$$

This definition reflects that we are happy with any arm having mean reward $\geqslant S$ and that there is no benefit in overfulfilling the given satisfaction level S. Note that the S-regret will be linear in T whenever there is no satisficing arm with mean reward $\geqslant S$, that is, if $\mu_* < S$. As already mentioned, a more general notion of regret that coincides with S-regret in our particular setting has been suggested by Reverdy et al. [2017].

2.2 The Realizable Case

We start with the *realizable case* when $\mu_* > S$. The main goal of this section is to show that suitable algorithms will have just constant S-regret in this case. Note that this does not hold for standard algorithms like UCB1 [Auer et al., 2002]. Lower bounds show that these algorithms will choose a suboptimal arm i for $\Omega\Big(\frac{logT}{(\mu^*-\mu_i)^2}\Big)$ times. This of course also holds for any arm below the satisfaction level S giving a contribution to the overall S-regret of $\Omega\Big(\frac{(S-\mu_i)\log T}{(\mu^*-\mu_i)^2}\Big)$.

2.2.1 Simple Algorithm

We start with a simple algorithm shown as Algorithm 1. It plays the empirical best arm so far if its empirical mean reward is $\geqslant S$ and explores uniformly at random otherwise. In the following, the empirical reward for arm i available at step t (i.e., before choosing the arm A_t) is denoted by $\hat{\mu}_i(t)$.

Algorithm 1

Require: K, S

Play each arm once, i.e., for time steps $t = 1, \ldots, K$ play arm $A_t = t$.
for time steps $t = K+1, \ldots$ **do**
    if $\exists i\, \hat{\mu}_i(t) \geqslant S$ **then**
        Play $A_t \leftarrow \operatorname{argmax}_{i \in [\![1,K]\!]} \hat{\mu}_i(t)$.
    else
        Choose A_t uniformly at random from $[\![1, K]\!]$.
    end if
end for

Analogously to the ordinary MAB setting where the gaps $\Delta_i := \mu_* - \mu_i$ to the optimal arm appear in bounds on the (classic) regret, when satisficing the gaps $\Delta_i^S = S - \mu_i$ for non-satisficing arms are important parameters describing the difficulty of the problem. Indeed, one can show the following bound on the S-regret.

Theorem 1. *If $S < \mu_*$ then Algorithm 1 satisfies for all $T \geqslant 1$*

$$R_T^S \leqslant \sum_{i:\Delta_i^S > 0} \Big(\Delta_i^S + \frac{2}{\Delta_i^S} + \frac{2\Delta_i^S}{|\Delta_*^S|^2} \Big).$$

For the proof we shall need the following result that follows by our assumption of 1-sub-Gaussianity and a Chernoff bound.

Lemma 1. *Let $\hat{\mu}_{i,n}$ be an empirical estimate for μ_i computed from n samples. Then for all $\varepsilon > 0$ and each $i \in [\![1, K]\!]$,*

$$\begin{aligned} \mathbb{P}(\hat{\mu}_{i,n} \geqslant \mu_i + \varepsilon) &\leqslant \exp(-\tfrac{n\varepsilon^2}{2}), \\ \mathbb{P}(\hat{\mu}_{i,n} \leqslant \mu_i - \varepsilon) &\leqslant \exp(-\tfrac{n\varepsilon^2}{2}). \end{aligned}$$

Proof of Theorem 1. Let i be the index of a non-satisficing arm. In the following we decompose the event that arm i is chosen at some step t. To do that we introduce the event $Z_t := \{\forall j \in [\![1, K]\!], \hat{\mu}_i(t) < S\}$ that all arms have empirical estimates below S, when the algorithm chooses an arm randomly according to line 6 of the algorithm. Then we have

$$\{A_t = i\} \subset \{t = i\} \cup \{A_t = i, Z_t^c\} \cup \{A_t = i, Z_t\}. \tag{1}$$

For the first two events we have

$$\sum_{t=1}^{T} \mathbb{P}(t = i) \leqslant 1 \tag{2}$$

and

$$\begin{aligned} \sum_{t=1}^{T} \mathbb{P}(A_t = i, Z_t^c) &\leqslant \sum_{t=1}^{T} \mathbb{P}\big(A_t = i, \hat{\mu}_i(t) \geqslant S\big) \\ &= \sum_{t=1}^{T} \mathbb{P}\big(A_t = i, \hat{\mu}_i(t) \geqslant \mu_i + \Delta_i^S\big) \leqslant \sum_{n=1}^{T} \mathbb{P}(\hat{\mu}_{i,n} \geqslant \mu_i + \Delta_i^S) \\ &\leqslant \sum_{n=1}^{T} \exp\big(-\tfrac{n(\Delta_i^S)^2}{2}\big) \leqslant \frac{e^{-\frac{(\Delta_i^S)^2}{2}}}{1 - e^{-\frac{(\Delta_i^S)^2}{2}}} \leqslant \frac{2}{(\Delta_i^S)^2}. \end{aligned} \tag{3}$$

Rewriting the probability of the third event in (2), using $*$ to refer to an arbitrary optimal arm, we obtain

$$\begin{aligned} \mathbb{P}(A_t = i, Z_t) &= \mathbb{P}(A_t = i | Z_t)\, \mathbb{P}(Z_t) = \tfrac{1}{K} \cdot \mathbb{P}(Z_t) \\ &= \mathbb{P}(A_t = * | Z_t)\, \mathbb{P}(Z_t) = \mathbb{P}(A_t = *, Z_t). \end{aligned}$$

Now summing over the time steps up to T yields

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{P}(A_t=i,Z_t)&=\sum_{t=1}^{T}\mathbb{P}(A_t=*,Z_t)\;\leqslant\sum_{t=1}^{T}\mathbb{P}\big(A_t=*,\hat{\mu}_*(t)\leqslant S\big)\\&=\mathbb{E}\Big(\sum_{t=1}^{T}\mathbb{1}\big\{A_t=*,\hat{\mu}_*(t)\leqslant S\big\}\Big)\;\leqslant\mathbb{E}\Big(\sum_{n=1}^{T}\mathbb{1}\{\hat{\mu}_{*,n}\leqslant S\}\Big)\\&=\sum_{n=1}^{T}\mathbb{P}(\hat{\mu}_{*,n}\leqslant S)\;=\sum_{n=1}^{T}\mathbb{P}\big(\hat{\mu}_{*,n}\leqslant\mu_*-|\Delta_*^S|\big)\\&\leqslant\sum_{n=1}^{T}\exp\big(-\tfrac{n|\Delta_*^S|^2}{2}\big)\;\leqslant\frac{2}{|\Delta_*^S|^2}.\end{aligned}\tag{4}$$

Finally writing

$$n_i(T)=\sum_{t=1}^{T}\mathbb{I}\{A_t=i\}$$

for the number of times arm i was pulled up to step T, we can combine (1)–(4) to obtain

$$\begin{aligned}R_T^S&=\sum_{i:\Delta_i^S>0}\Delta_i^S\,\mathbb{E}(n_i(T))\;=\sum_{i:\Delta_i^S>0}\Delta_i^S\sum_{t=1}^{T}\mathbb{P}(A_t=i)\\&\leqslant\sum_{i:\Delta_i^S>0}\Delta_i^S\left(1+\frac{2}{(\Delta_i^S)^2}+\frac{2}{|\Delta_*^S|^2}\right)\leqslant\sum_{i:\Delta_i^S>0}\Big(\Delta_i^S+\frac{2}{\Delta_i^S}+\frac{2\Delta_i^S}{|\Delta_*^S|^2}\Big).\end{aligned}$$

□

The algorithm as well as the analysis are adaptations from Bubeck et al. [2013] where ordinary regret bounds for the MAB setting are considered under the assumption that the learner knows the value of μ_* as well as (a bound on) the gap Δ between the optimal and the best suboptimal arm. The crucial insight is that what is actually needed in order to apply algorithm and analysis of Bubeck et al. [2013] is to have a reference value μ that separates the optimal from suboptimal arms, that is, $\mu_*>\mu>\mu_i$ for all suboptimal arms i. In our case this reference value is given by the satisfaction level S, which in the realizable case separates the good arms from the bad ones. Note that for this we need to have $S<\mu_*$, so that the special case $S=\mu_*$ does not give constant regret.

Remark. The constant regret bound of Theorem 1 not only improves over the logarithmic bounds given by Reverdy et al. [2017], it also is not consistent with a claimed lower bound that is also logarithmic in the horizon (not mentioned in the corrections [Reverdy et al., 2021]). This bound is obtained by application of a lower bound for the *multiple play* setting [Anantharam et al., 1987], where at each step m arms are chosen by the learner, who hence has to identify the m best arms. The given proof chooses m to be all arms above the given satisfaction level S. However, the lower bound is obviously not directly applicable to the satisficing setting: not *all* arms above the satisfaction level have to be found, but a single one is sufficient.

2.2.2 Exploration based on a potential function

Bubeck et al. [2013] also provide another algorithm with a more refined approach for exploration, that we adapt here to the satisficing setting. The respective algorithm shown as Algorithm 2 uses a potential function $\psi : [0, \infty) \to \mathbb{R}^+$ that is assumed to be differentiable and increasing.

Algorithm 2

Require: K, S, T

Play each arm once, i.e., for time steps $t = 1, \dots, K$ play arm $A_t = t$.
for time steps $t = K + 1, \dots, T$ **do**
  if $\exists i \, \hat{\mu}_i(t) \geqslant S$ **then**
    Play $A_t \leftarrow \operatorname{argmax}_{i \in [\![1,K]\!]} \hat{\mu}_i(t)$.
  else
    Choose randomly an arm according to the probability distribution defined by

$$p_{i,t} = \frac{1}{\alpha \times \psi(|S - \hat{\mu}_i(t)|)}, \text{ where } \alpha = \sum_{j=1}^{K} \frac{1}{\psi(|S - \hat{\mu}_j(t)|)}.$$

  end if
end for

Theorem 2. *Let $\psi : [0, \infty) \to \mathbb{R}^+$ be a differentiable and increasing function. If $\mu_* < S$ then Algorithm 2 satisfies for all $T \geqslant 1$,*

$$R_T^S \leqslant \sum_{i:\Delta_i^S > 0} \left(\Delta_i^S + \frac{8}{\Delta_i^S} + \frac{\Delta_i^S}{\psi(\frac{\Delta_i^S}{2})} \left(\frac{2\psi(0)}{(\Delta_*^S)^2} + \int_0^{+\infty} \frac{2\psi'(x)}{e^{\frac{(|\Delta_*^S| + x)^2}{2}} - 1} \, dx \right) \right) \tag{5}$$

The proof of Theorem 2 can be found Appendix A. Choosing the potential function to be $\psi(x) = x^2$ as suggested by Bubeck et al. [2013] yields e.g. a bound similar to that of Theorem 1, that is,

$$R_T^S \leqslant \sum_{i:\Delta_i^S > 0} \left(\Delta_i^S + \frac{8}{\Delta_i^S} + \frac{\Delta_i^S}{(\Delta_*^S)^2} \left(2 + 8 \log \left(\frac{\sqrt{2}}{|\Delta_*^S|} \right) \right) \right).$$

As pointed out by Bubeck et al. [2013] other choices may give improved bounds. Another particular potential function is $\psi(x) = e^{\frac{x}{\lambda}}$ that gives a distribution similar to the one of the Softmax algorithm.

2.3 The General Case

Now let us consider the general case where it is not guaranteed that the chosen satisfaction level S is realizable, that is, it may happen that $S > \mu_*$. Then contrary to the realizable case the satisfaction level S does not give the learner any useful information so that we cannot hope to perform better than in an ordinary MAB setting. Obviously the S-regret will be linear, but we aim at getting bounds on the (classic) regret. On the other hand, if there is at least one arm above the satisfaction level S, we would like to re-establish constant bounds on the S-regret as in the realizable case.

For the general setting we propose Algorithm 3. It uses a more refined approach for exploitation. Instead of just deciding based on the empirical estimate $\hat{\mu}_i$ of each arm i, it considers for each arm

a confidence interval defined by the two values (the first one being similar to the classical value suggested for the UCB1 algorithm of Auer et al. [2002])

$$\mathrm{UCB}_i(t) := \hat{\mu}_i(t) + \beta_i(t), \quad \text{and} \quad \mathrm{LCB}_i(t) := \hat{\mu}_i(t) - \beta_i(t), \tag{6}$$
$$\text{where } \beta_i(t) = \sqrt{\tfrac{2\log(f(t))}{n_i(t-1)}} \text{ with } f(t) = 1 + t\log^2(t).$$

The upper confidence bound of arm i is chosen such that the expected reward of the arm is below the bound with high probability and the range of the confidence interval decrease as the number of samples increase. More formally, we deduce from Lemma 1 that

$$\mathbb{P}(\hat{\mu}_i(t) \geqslant \mathrm{UCB}_i(t)) \leqslant \frac{1}{f(t)}.$$

The lower confidence bound is chosen in a similar fashion. UCB1 algorithm of Auer et al. [2002] implements the paradigm of optimism in the face of uncertainty. At each time step, the chosen arm is one with the highest plausible expected reward (ie. the highest upper confidence bound). In addition, the degree of confidence $\frac{1}{f(t)}$ is increased overtime in order to avoid getting stuck on a suboptimal arm after a few unlucky samples of the optimal arm. This increasing level of confidence leads to the sampling of suboptimal arms infinitely many times in order to confirm previous observations and decrease their upper confidence bounds. This behavior should to be avoided in the satisficing setting while keeping the increasing level of confidence to avoid not satisfying arms.

The algorithm 3 chooses the arm for which the largest share of this confidence interval is above the satisfaction level S (cf. line 4 of the algorithm) provided that the upper confidence bound UCB_i of some arm i is at least S. Otherwise, if all arms appear to be below the satisfaction level, the algorithm chooses an arm according to UCB1, that is, an arm i maximizing UCB_i.

Algorithm 3

Require: K, S, T

1: Play each arm once, i.e., for time steps $t = 1, \ldots, K$ play arm $A_t = t$.
2: **for** time steps $t = K+1, \ldots, T$ **do**
3: **if** $\exists i\ \mathrm{UCB}_i(t) \geqslant S$ **then**
4: Choose $i \in \underset{j \in [\![1,K]\!]}{\operatorname{argmax}} \left\{ \frac{\mathrm{UCB}_j(t) - \max\{S, \mathrm{LCB}_j(t)\}}{\beta_j(t)} \right\}$.
5: **else**
6: Play arm $A_t \leftarrow \underset{j \in [\![1,K]\!]}{\operatorname{argmax}} \mathrm{UCB}_j(t)$.
7: **end if**
8: **end for**

The following two theorems show that Algorithm 3 is able to achieve constant S-regret if $\mu_* > S$, while the regret is bounded as for UCB1 otherwise [Auer et al., 2002].

Theorem 3. *If $\mu_* > S$ then Algorithm 3 satisfies for all $T \geqslant 1$,*

$$R_T^S \leqslant \sum_{i:\Delta_i^S > 0} \left(\Delta_i^S + \frac{2}{\Delta_i^S} + \frac{7\Delta_i^S}{|\Delta_*^S|^2} \right).$$

Proof. As before we write the S-regret as

$$R_T^S = \sum_{i:\Delta_i^S>0}^{k} \mathbb{E}(n_i(T))\,\Delta_i^S$$

and proceed bounding $\mathbb{E}(n_i(T)) = \sum_{t=1}^{T} \mathbb{P}(A_t = i)$ for all non-satisficing arms i. Thus let i be the index of a non-satisficing arm. Defining $G_t = \{\forall j \in [\![1, K]\!], \mathrm{UCB}_j(t) < S\}$ to be the event that all arms are emprically below the satisfaction level, we decompose the event $\{A_t = i\}$ as

$$\begin{aligned}\{A_t = i\} \subset \{t = i\} \cup \big\{A_t = i, \hat{\mu}_i(t) \geqslant S, t > K\big\} \\ \cup \big\{A_t = i, \hat{\mu}_i(t) < S, t > K, G_t\big\} \cup \big\{A_t = i, \hat{\mu}_i(t) < S, t > K, G_t^c\big\}. \end{aligned} \tag{7}$$

For the first two events we have

$$\sum_{t=1}^{T} \mathbb{P}(t = i) \leqslant 1 \tag{8}$$

and

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{P}\big(A_t = i, \hat{\mu}_i(t) > S, t > K\big) &\leqslant \sum_{t=1}^{T} \mathbb{P}\big(A_t = i, \hat{\mu}_i(t) \geqslant \mu_i + \Delta_i^S\big) \\ &\leqslant \sum_{n=1}^{T} \mathbb{P}(\hat{\mu}_{i,n} \geqslant \mu_i + \Delta_i^S) \leqslant \sum_{n=1}^{T} \exp\big(-\tfrac{n(\Delta_i^S)^2}{2}\big) \\ &\leqslant \frac{e^{-\frac{(\Delta_i^S)^2}{2}}}{1 - e^{-\frac{(\Delta_i^S)^2}{2}}} \leqslant \frac{2}{(\Delta_i^S)^2}\,. \end{aligned} \tag{9}$$

For bounding the probability of the fourth event of (7), we claim that $A_t = i$ and $\exists j \in [\![1, K]\!]$ such that $\mathrm{UCB}_j(t) \geqslant S$ implies that

$$i = \operatorname*{argmax}_j \left\{ \frac{\mathrm{UCB}_j(t) - \max\{S, \mathrm{LCB}_j(t)\}}{\beta_j(t)} \right\}.$$

Indeed, we have

$$\frac{\mathrm{UCB}_i(t) - \max\{S, \mathrm{LCB}_i(t)\}}{\beta_i(t)} \geqslant \frac{\mathrm{UCB}_*(t) - \max\{S, \mathrm{LCB}_*(t)\}}{\beta_*(t)}, \tag{10}$$

so that if $\hat{\mu}_i(t) < S$ then also $\hat{\mu}_*(t) < S$ as otherwise the term on the left hand side of (10) would be < 1 and the term on the right hand side > 1. Hence, similar as before we obtain

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{P}\big(A_t = i, \hat{\mu}_i(t) < S, t > K, G_t^c\big) &\leqslant \sum_{t=1}^{T} \mathbb{P}\big(A_t = i, \hat{\mu}_*(t) < S\big) \\ &\leqslant \sum_{n=1}^{T} \mathbb{P}(\hat{\mu}_{*,n} \leqslant S) \leqslant \frac{2}{|\Delta_*^S|^2}\,. \end{aligned} \tag{11}$$

Finally, the probability of the third event of (7) is bounded by the probability of G_t so that

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{P}\big(A_t = i, \hat{\mu}_i(t) < S, t > K, G_t\big) &\leqslant \sum_{t=1}^{T} \mathbb{P}(G_t) \;= \sum_{t=1}^{T} \mathbb{P}\big(\forall j \in [\![1, K]\!], \mathrm{UCB}_j(t) < S\big) \\
&\leqslant \sum_{t=1}^{T} \mathbb{P}\big(\,\mathrm{UCB}_*(t) < S\big) \;= \sum_{t=1}^{T} \mathbb{P}\big(\hat{\mu}_*(t) < \mu_* - (|\Delta_*^S| + \beta_*(t))\big) \\
&\leqslant \sum_{t=1}^{T} \sum_{n=1}^{t} \mathbb{P}\left(\hat{\mu}_{*,n} < \mu_* - \left(|\Delta_*^S| + \sqrt{\tfrac{2\log(f(t))}{n}}\right)\right) \\
&\leqslant \sum_{t=1}^{T} \sum_{n=1}^{t} \frac{1}{f(t)} \exp\big(-\tfrac{n|\Delta_*^S|^2}{2}\big) \leqslant \frac{2}{|\Delta_*^S|^2} \sum_{t=1}^{T} \frac{1}{f(t)} \leqslant \frac{5}{|\Delta_*^S|^2}\,. \qquad (12)
\end{aligned}$$

The last inequality is obtained by observing that $\sum_{t=1}^{T} \frac{1}{f(t)} \leqslant 1 + \sum_{t=2}^{T} \frac{1}{t\log^2(t)}$ and then bounding the sum with an integral.

Finally, by putting everything together, we obtain from equations (7), (8), (9), (11), and (12) the claimed result

$$R_T^S = \sum_{i:\Delta_i^S > 0} \Delta_i^S\, \mathbb{E}(n_i(T)) \leqslant \sum_{i:\Delta_i^S > 0} \Big(\Delta_i^S + \frac{2}{\Delta_i^S} + \frac{7\Delta_i^S}{|\Delta_*^S|^2}\Big). \qquad \square$$

Remark. Instead of choosing the arm whose share of confidence interval above S is maximal one may consider simpler ways to choose an arm empirically above S. Actually our analysis goes through as long as it holds that if i is chosen in the fourth event of (7) it holds that $\hat{\mu}_*(t) < S$. Thus, it would be for example sufficient if an arm having maximal empirical mean reward is chosen. Another alternative policy for which the bound of Theorem 3 holds would be to employ UCB1 to choose among the arms with empirical mean reward above the satisfaction level. We will see that Algorithm 3 empirically outperforms these simpler alternatives. However, in our analysis we were not able to prove regret bounds (which always consider the worst case) confirming this also theoretically.

Theorem 4. *If $\mu_* \leqslant S$ then Algorithm 3 satisfies for all $T \geqslant 1$*

$$R_T \leqslant \sum_{i:\Delta_i > 0} \inf_{\varepsilon \in (0, \Delta_i)} \Delta_i \left(1 + \frac{5}{\varepsilon^2} + \frac{2(\log f(T) + \sqrt{\pi \log f(T)} + 1)}{(\Delta_i - \varepsilon)^2}\right). \qquad (13)$$

Furthermore,

$$\limsup_{T\to\infty} \frac{R_T}{\log(T)} \leqslant \sum_{i:\Delta_i > 0} \frac{2}{\Delta_i}. \qquad (14)$$

Thus, for a constant $C > 0$ it holds that

$$R_T \leqslant C \sum_{i:\Delta_i > 0} \left(\Delta_i + \frac{\log(T)}{\Delta_i}\right).$$

Proof. The proof can be done analogously to that of Theorem 8.1 from Lattimore and Szepesvári [2020] (deriving regret bounds for UCB). We start with the standard regret decomposition

$$R_T = \sum_{i:\Delta_i>0} \mathbb{E}\left(n_i(T)\right) \Delta_i.$$

In the following, we bound for each suboptimal arm i the number of times $n_i(T)$ it is played. Note that arm i is only chosen if either

$$\hat{\mu}_i(t) + \beta_i(t) \geqslant \hat{\mu}_*(t) + \beta_*(t) \quad \text{or} \quad \hat{\mu}_i(t) + \beta_i(t) \geqslant S.$$

Accordingly, we can decompose the event $A_t = i$ using some arbitrary but fixed $\varepsilon \in (0, \Delta_i)$ as

$$\begin{aligned}
\{A_t = i\} \subseteq& \big\{A_t = i \text{ and } \hat{\mu}_*(t) + \beta_*(t) \leqslant \mu_* - \varepsilon\big\} \cup \big\{A_t = i \text{ and } \hat{\mu}_*(t) + \beta_*(t) \geqslant \mu_* - \varepsilon\big\} \\
\subseteq& \big\{\hat{\mu}_*(t) + \beta_*(t) \leqslant \mu_* - \varepsilon\big\} \\
&\cup \big\{A_t = i \text{ and } \hat{\mu}_i(t) + \beta_i(t) \geqslant \hat{\mu}_*(t) + \beta_*(t) \geqslant \mu_* - \varepsilon\big\} \\
&\cup \big\{A_t = i \text{ and } \hat{\mu}_*(t) + \beta_*(t) \geqslant \mu_* - \varepsilon \text{ and } \hat{\mu}_i(t) + \beta_i(t) \geqslant S\big\} \\
\subseteq& \big\{\hat{\mu}_*(t) + \beta_*(t) \leqslant \mu_* - \varepsilon\big\} \\
&\cup \big\{A_t = i \text{ and } \hat{\mu}_i(t) + \beta_i(t) \geqslant \mu_* - \varepsilon\big\},
\end{aligned}$$

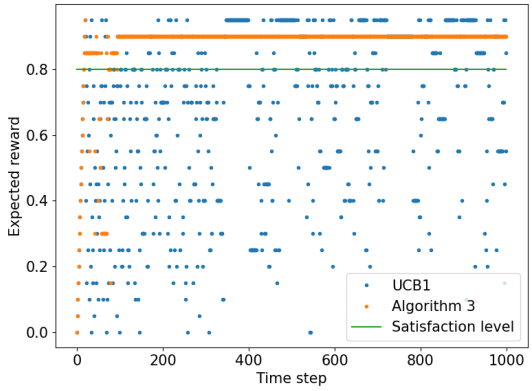
where the last inclusion is due to the assumption that $\mu_* \leqslant S$. It follows that

$$n_i(T) \leqslant \sum_{t=1}^{T} \mathbb{I}\big\{\hat{\mu}_*(t) + \beta_*(t) \leqslant \mu_1 - \varepsilon\big\} + \sum_{t=1}^{T} \mathbb{I}\big\{A_t = i \text{ and } \hat{\mu}_i(t) + \beta_i(t) \geqslant \mu_* - \varepsilon\big\}.$$

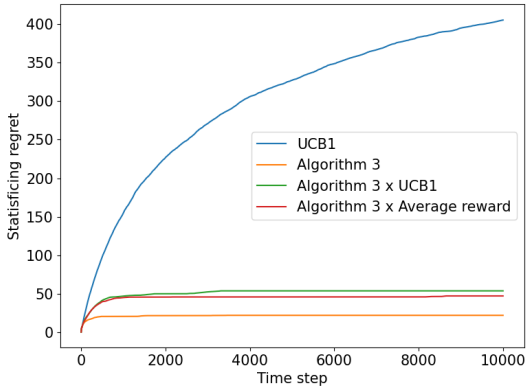
The obtained decomposition is the same as the one in the proof of Theorem 8.1 from Lattimore and Szepesvári [2020] and the very same arguments can be used to finish the proof of (13). The second part of the theorem, that is eq. (14), follows by choosing $\varepsilon = \log^{-1/4}(T)$ and taking the limit as T tends to infinity. □

2.4 Experiments

We compared our Algorithm 3 to standard bandit algorithms in order to show that the latter keep accumulating S-regret, while Algorithm 3 sticks to a satisficing arm after finite time, thus confirming the results of Theorem 3. We started with comparing Algorithm 3 to UCB1 [Auer et al., 2002] in a setting with 20 arms and normally distributed rewards with standard deviation 1 and the mean reward of arm i set to $\frac{i-1}{20}$. The satisfaction level was set to 0.8. Figure 1 shows the results with (1a) depicting a showcase run illustrating that Algorithm 3 soon focuses on a satisficing arm, while UCB1 keeps exploring. Figure 1b gives the S-regret averaged over 50 runs. Although the latter in general is smaller than classic regret, UCB1 suffers growing S-regret due to ongoing exploration of arms below the satisfaction level. Figure 1b also compares Algorithm 3 to variants that use a different criterion for choosing among empirically satisficing arms. That is, instead of using the fraction index in line 4 of Algorithm 3 we consider using UCB1 for choosing an arm or pick the arm having the highest empirical mean, respectively. We see in Figure 1b that the original version of Algorithm 3 works best.

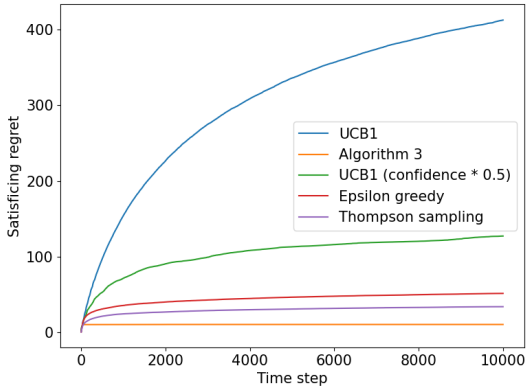


(a) Arm pulls for each algorithm in exemplary run.

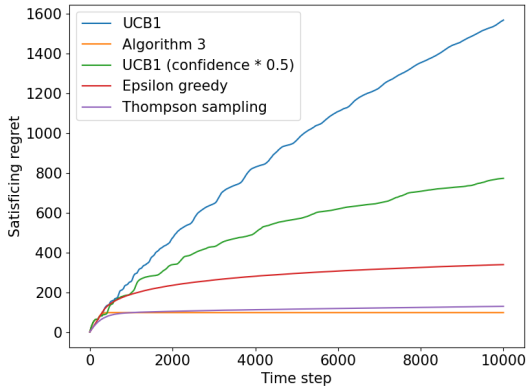


(b) S-regret averaged over 50 runs.

Figure 1: Experiments with Gaussian bandits comparing Algorithm 3 to UCB1.



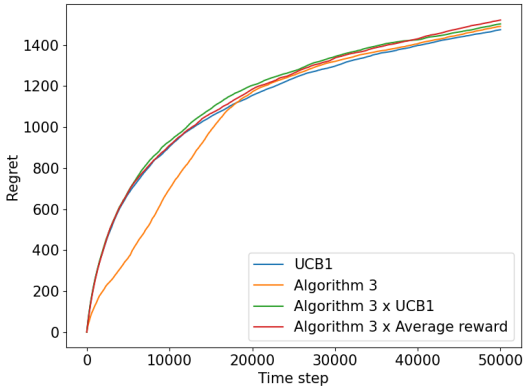
(a) S-regret for 20 arms averaged over 50 runs.



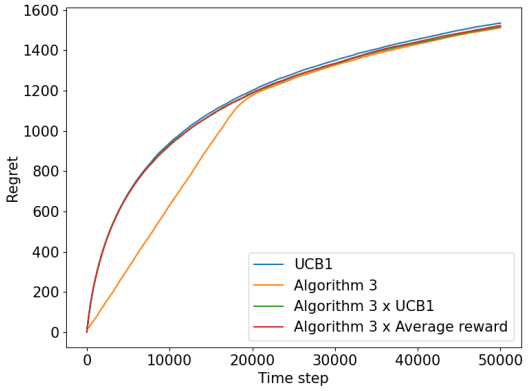
(b) S-regret for 200 arms averaged over 50 runs.

Figure 2: Experiments with Bernoulli bandits comparing Algorithm 3 to classic bandit algorithms.

Next, we considered Bernoulli distributed rewards and added Thompson sampling [Thompson, 1933] and ε-greedy [Auer et al., 2002] to the comparison. For ε-greedy we chose $\varepsilon_t := \frac{K}{10t}$ at each step t, while we used a version of Thompson sampling adapted to Bernoulli rewards, using Beta distributions for the estimate. Further, for UCB1 we halved the bonus term for focusing more on exploitation. Figure 2 shows that Algorithm 3 is still the only one giving constant S-regret. Not surprisingly, if the number of arms is raised from 20 to 200, S-regret increases. However, maybe with the exception of Thompson sampling the classic bandit algorithms seem to suffer more from the increase of the number of arms.



(a) Classic regret for Gaussian bandits averaged over 50 runs.



(b) Classic regret for Bernoulli bandits averaged over 50 runs.

Figure 3: Experiments for the not realizable case when $S > \mu_*$.

Finally we also had a look at the not realizable case when the satisfaction level is chosen above μ_*. For this we used the same setup as before but chose $S = 1 > \mu_* = \frac{19}{20}$. Here the regret of the variants of Algorithm 3 is practically the same as for UCB1. Surprisingly, figure 3 shows that Algorithm 3 itself is superior to UCB1 for a long time, until the regret curve finally joins that of UCB1.

3 Satisficing for Markov Decision Processes

3.1 Setting

We now consider Markov decision processes (MDP) with finite state space $\mathcal{S}$ and finite action space $\mathcal{A}$ (their respective sizes are S and A). In an MDP M, a learner needs to choose an action a to execute from its current state. When executing action a in state s, the learner receives a random reward drawn independently from some distribution. Then, according to the transition probabilities $p(s'|s,a)$, a random transition to a state $s' \in S$ occurs. As in the multi-armed bandits case, we will consider that the reward distributions are subgaussians.

The strategy followed by the decision maker is called a policy. In particular, a stationary policy π specifies for each state $s \in \mathcal{S}$ the action to execute $\pi(s) \in \mathcal{A}$ independently of the current time step as well as the previously chosen actions and their outcomes.

We can define the accumulated reward of an algorithm $\mathfrak{A}$ after T steps in an MDP M with initial state s as

$$R(M, \mathfrak{A}, s, T) = \sum_{t=1}^{T} r_t$$

where r_t is the random reward obtained at step t after playing action a_t chosen by the algorithm. When it exists, the average reward of the process is

$$\rho(M, \mathfrak{A}, s) := \lim_{T \to \infty} \frac{1}{T} \mathbb{E}[R(M, \mathfrak{A}, s, T)]$$

This value can be maximized by an appropriate stationary policy $\pi : \mathcal{S} \to \mathcal{A}$ in place of the algorithm $\mathfrak{A}$. In the following we only consider stationary policies.

Furthermore, we assume the MDP to be communicating which is usually the most general case considered in the regret analysis of reinforcement learning algorithm such as UCRL2 Jaksch et al. [2010].

Definition 1. *An MDP is communicating if for every states i,j there exists a deterministic policy π, which may depend on i and j, such that in the Markov chain induced by π the state j is accessible from state i.*

This condition ensures, in particular, that no parts of the MDP are made inaccessible from previous choices, which may otherwise lead to linear regret. This also implies that there exists an optimal stationary policy with an average reward ρ^* independent of the initial state [Puterman, 1994]. Thus, we can set

$$\rho^*(M) := \rho^*(M, s) = \max_{\pi} \rho(M, \pi, s).$$

For an MDP M, the standard definition for the total regret of $\mathfrak{A}$ after T steps is

$$\Delta(M, \mathfrak{A}, s, T) := T\rho^*(M) - R(M, \mathfrak{A}, s).$$

As for the MAB, we consider satisficing in mean reward. However, in the MDP setting, it seems too restrictive to consider an individual action as being satisfying. It is more natural to consider that playing a policy with an average reward above a certain level is sufficient, even if some of its actions lead to rewards below the satisfaction level.

Definition 2. *A policy π is satisfying for the MDP M and the initial state s with respect to the satisfaction level σ if $\rho(M, \pi, s) \geqslant \sigma$.*
A policy π is satisfying for the MDP M with respect to the satisfaction level σ if the previous condition holds for every initial states.

This definition is consistent with the one used for the MAB setting. Indeed, a K-armed bandit can be seen has an MDP with a unique state and K actions looping back to the original state, so choosing a stationary policy is nothing more than choosing an arm to play at every step. The average reward of the policy is exactly the average reward of the arm played.

As the notion of satisfaction is now related to a policy, the satisficing regret cannot be adapted from the standard regret definition as we did before in the MAB setting. Indeed, playing only a satisfying policy may lead to an expected total regret linear in the number of steps, as some unsatisfying action can be played infinitely often while keeping an average reward above the satisfaction level. Some MDPs have satisfying policies but no policy without unsatisfying actions. We need to develop alternative metrics for the evaluation of a satisficing algorithms. A first idea is to divide a run of the algorithm into multiple time intervals of a given size and computing the difference between the average observed reward during a time interval and the satisfaction level. One could also consider a definition derived from the (standard) regret in which we replace the reward of the current action by the average reward of the policy currently played. Both these metrics require arbitrary choices and may not be suited to compare objectively different RL algorithms.

As a first approach, we consider the algorithm to be composed of a succession of episodes during which a policy with a well-defined average reward is followed. Under this assumption, given a

communicating MDP M and a satisfaction level σ, we can define the number of unsatisfying steps using the algorithm $\mathfrak{A}$ after t steps as

$$T^U(\mathfrak{A}, M, \sigma, t) := |\{t' | t' \leqslant t \wedge \rho(\pi_t^{\mathfrak{A}}, M, s_t) \geqslant \sigma\}|$$

where $\pi_t^{\mathfrak{A}}$ is the policy played by the algorithm $\mathfrak{A}$ at step t.

We denote the gap between the average reward of a policy π and the satisfaction level σ as

$$\Delta_\pi := \max_{s \in \mathcal{S}}(\sigma - \rho(\pi, M, s))$$

.

3.2 The Deterministic Case

The algorithm UCRL2 [Jaksch et al., 2010] is a successful implementation of the paradigm of optimism in the face of uncertainty and achieves near-optimal results regarding standard regret. This algorithm can roughly be seen as a generalization of UCB1 [Auer et al., 2002]. The idea of UCRL2 is to play a policy achieving the highest reward among a set of plausible MDPs instead of the action with the highest reward among a set a plausible MAB instances as it was the case with UCB1. We want to perform a similar adaptation in order to design an algorithm for satisficing in MDPs.

One of the main difficulties with MDP compared to MAB is that both rewards and transition probabilities are unknown and must be learned overtime. This added uncertainty makes the accurate estimation of the average reward of a policy more challenging. Additionally, the structure of an MDP can make the collection of specific samples harder since moving from a state to another is now a random process, so the progression of the estimation accuracy of the probabilities and average rewards may differ from the intended behavior. Indeed, some actions may be sampled exceedingly often during the process of moving toward barely accessible states to sample promising but very uncertain actions.

We first consider MDPs with deterministic transitions, meaning that for all states s and s' and all actions a we have $p(s, a) \in \{0, 1\}$. This case can be seen as a bridge between MAB and MDP. Indeed, after an initial exploration all the transitions are known and every action is accessible easily from any state. The average reward estimates only depend on the estimated rewards of each action since the structure of the MDP is known perfectly.

We propose a policy designed as an extension of Algorithm 1 for the realizable case. In the algorithm, M_k is the estimated MDP at the beginning of episode k and $\rho_k(\pi, s)$ is the average reward of a policy π in M_k and s_k is the current state at the beginning of the episode k. The computation and update of these values are omitted for the sake of simplicity and readability.

Algorithm 4

Input: $\mathcal{S}$, $\mathcal{A}$ and the satisfaction level σ.

While $\exists (s,a) \in \mathcal{S} \times \mathcal{A}$ such that $N(s,a) = 0$ and s is accessible from the current state **do**:

1. Follow the shortest path from the current state to s.
2. Play action a
3. Update the shortest paths between each pair of state in the multi-graph associated to the currently discovered MDP

For episodes $k = 1, 2, \dots$ **do**

4. Compute the optimal policy π_k of the estimated MDP M_k and its average reward $\rho_k(\pi_k, s_k)$ in M_k.
5. **If** $\rho_k(\pi_k) \geqslant \sigma$ **then** use π_k on M for S steps.
6. **Else** Sample at least once the rewards associated to each action of each state by following the shortest path from the current state to the next action to be sampled.

Theorem 5. *Let M be a communicating MDP with deterministic transition with 1-subgaussian rewards and σ a satisfaction level. If $\sigma > \max_{\pi,s} \rho(\pi, s)$ then Algorithm 4 satisfies for all $K \geqslant 1$*

$$\mathbb{E}(T^U(\mathcal{A}_4, M, \sigma, t)) \leqslant S^2 A \left[1 + \frac{2SA}{\Delta_U^2} + \frac{2}{\Delta_*^2}\right], \tag{15}$$

where $\Delta_U = \min_{\pi:\Delta_\pi>0} \Delta_\pi$ is the gap between the satisfaction level and the average reward of the best unsatisfying policy and $\Delta_* = \min_\pi \Delta_\pi$ is the gap between the satisfaction level and the average reward of the optimal policy.

While it may be possible to already obtain a finite bound for the number of unsatisfying steps by adapting the analysis of Algorithm 1, the final results would not really be satisfying. Indeed, such an analogy considers each policy as an individual arm, so the corresponding MAB would have A^S arms. As a sum over all the arms, the derived satisficing regret bound would be largely overestimated. Intuitively, not all the policies need to be explored as a lot of actions are in common, so the samples obtained by following one policy can be used to improve the estimated rewards of other policies. An implementation of this idea is used in the proof to evaluate the improvement of the estimated reward of individual actions while following unsatisfying policies.

Remark. Since we consider MDP with deterministic transitions, we can compute the optimal policy by reducing the problem to finding a cycle with maximum average weight in a directed graph which can be computed using Karp's maximum mean cycle algorithm [Karp, 1978]. The algorithm also works using value iteration, which returns an ε-optimal policy for M_k, as long as we sufficiently increase the accuracy overtime.

3.3 Toward the General Case

In order to generalize the previous algorithm to communicating MDP the estimation errors due to the uncertainty of the transition probabilities has to be taken into account. Previous works

introduce the notion of MDP approximation [Ortner et al., 2014] which allows to bound the average reward estimation error using the local error bounds on the rewards and probabilities. However, this bound depends on the highest local error, thus requires to sample every action sufficiently many times, leading to high regret. A different approach may be needed in a satisficing setting in order to avoid too much exploration when we already discovered promising policies.

Furthermore, exploration becomes more challenging since given a state and chosen action, the next state is uncertain. An efficient method to explore an unknown MDP is by using GOSPRL algorithm [Tarbouriech et al., 2021]. This algorithm walks through the MDP in order to fulfil a sampling goal efficiently. This goal can be set in order to obtain an MDP approximation of a given precision or to explore a subset of the MDP in a focused manner. The alternative and probably preferred way to explore is by following a policy similar to UCRL2 [Jaksch et al., 2010]. This may allow achieving near optimal regret bounds in the not realizable case, unlike the more uniform exploration based on GOSPRL.

We have found these two approaches promising to design a satisficing algorithm in a more general MDP setting, and we believe that more research along these lines may prove successful in achieving and analysing such algorithm.

4 Conclusion

While the particular case of Markov decision process presented in this study may seem quite limited, we hope it can give some insights for the analysis of the general case. It is already clear that defining a proper metric to evaluate the notion of *satisficing* is not quite simple. More work on a definition reflecting the intuitive notion of satisficing would benefit both the MAB and the MDP settings.

Also for the MAB setting itself, further improvements are possible. While we were happy to compete mainly with UCB1 in the non-realizable case, we are sure that suitable modifications of Algorithm 3 would give improved experimental performance while keeping logarithmic regret bounds.

A lesson to take from the MAB setting is that the savings from considering a satisficing instead of an optimizing objective –at least with respect to regret– is not that there are arms that need no exploration at all. Rather in the worst case (as always considered by notions of regret) one still has to explore all arms, however the amount of necessary exploration is now constant and independent of the horizon. It is an interesting question whether there may be alternative criteria that could be considered which are more in accordance with the intuition that with a satisficing objective one can (in the best case) limit the exploration to a small part of the policy space. After all, when the first arm sampled by the learner is already satisficing, there may be no need of further exploration at all. Accordingly, it seems difficult to obtain any nontrivial lower bounds on the satisficing regret.

Even if concrete applications have yet to be found, we believe that the 'satisficing' framework may prove useful in constrained environments in which resources such as time and computing power should be limited. This approach follows the same idea as budgeted learning which aims at finding methods to optimize the learning of the agent given the resources available instead of just considering the end result after enough training. Both these objectives are alternatives to the current trend of using more and more computing power and very complex models to train artificial agents, at the cost of a very high economic and ecological impact.

References

Venkatachalam Anantharam, Pravin Varaiya, and Jean Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays—part I: I.i.d. rewards. *IEEE Trans. Autom. Control*, 32:968–976, 12 1987. doi: 10.1109/TAC.1987.1104491.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, 2002. doi: 10.1023/A:1013689704352.

Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet. Bounded regret in stochastic multi-armed bandits. In *COLT 2013 – The 26th Annual Conference on Learning Theory*, volume 30 of *JMLR Workshop and Conference Proceedings*, pages 122–134, 2013. URL `http://proceedings.mlr.press/v30/Bubeck13.html`.

Arghya Roy Chaudhuri and Shivaram Kalyanakrishnan. PAC identification of a bandit arm relative to a reward quantile. In Satinder Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 1777–1783, 2017. URL `http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14335`.

Arghya Roy Chaudhuri and Shivaram Kalyanakrishnan. PAC identification of many good arms in stochastic multi-armed bandits. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, pages 991–1000. PMLR, 2019. URL `http://proceedings.mlr.press/v97/chaudhuri19a.html`.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

Hideaki Kano, Junya Honda, Kentaro Sakamaki, Kentaro Matsuura, Atsuyoshi Nakamura, and Masashi Sugiyama. Good arm identification via bandit feedback. *Mach. Learn.*, 108(5):721–745, 2019.

Richard M Karp. A characterization of the minimum cycle mean in a digraph. *Discrete mathematics*, 23(3): 309–311, 1978.

Yu Kohno and Tatsuji Takahashi. A cognitive satisficing strategy for bandit problems. *International Journal of Parallel, Emergent and Distributed Systems*, 32(2):232–242, 2017. doi: 10.1080/17445760.2015.1075531.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Blake Mason, Lalit Jain, Ardhendu Tripathy, and Robert Nowak. Finding all ε-good arms in stochastic bandits. In *Advances in Neural Information Processing Systems 33, NeurIPS 2020*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/edf0320adc8658b25ca26be5351b6c4a-Abstract.html`.

Ronald Ortner, Odalric-Ambrym Maillard, and Daniil Ryabko. Selecting near-optimal approximate state representations in reinforcement learning. In *International Conference on Algorithmic Learning Theory*, pages 140–154. Springer, 2014.

Martin L. Puterman. Markov decision processes: Discrete stochastic dynamic programming. In *Wiley Series in Probability and Statistics*, 1994.

Paul Reverdy, Vaibhav Srivastava, and Naomi Ehrich Leonard. Modeling human decision making in generalized Gaussian multiarmed bandits. *Proc. IEEE*, 102(4):544–571, 2014. doi: 10.1109/JPROC.2014.2307024.

Paul Reverdy, Vaibhav Srivastava, and Naomi Ehrich Leonard. Satisficing in multi-armed bandit problems. *IEEE Trans. Autom. Control.*, 62(8):3788–3803, 2017. doi: 10.1109/TAC.2016.2644380.

Paul Reverdy, Vaibhav Srivastava, and Naomi Ehrich Leonard. Corrections to "Satisficing in multiarmed bandit problems". *IEEE Trans. Autom. Control*, 66(1):476–478, 2021. doi: 10.1109/TAC.2020.2981433.

Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *J. Artif. Intell. Res.*, 48:67–113, 2013. doi: 10.1613/jair.3987.

Herbert A Simon. Rational choice and the structure of the environment. *Psychological review*, 63(2), 1956.

Akihiro Tamatsukuri and Tatsuji Takahashi. Guaranteed satisficing and finite regret: Analysis of a cognitive satisficing value function. *Biosystems*, 180:46–53, June 2019. doi: 10.1016/j.biosystems.2019.02.009.

Jean Tarbouriech, Matteo Pirotta, Michal Valko, and Alessandro Lazaric. A provably efficient sample collection strategy for reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 7611–7624, 2021.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3–4):285–294, 1933.

Appendices

A Proof of Theorem 2

As in the proof of Theorem 1 we aim at a bound on $\mathbb{E}(n_i(T)) = \sum_{t=1}^{T} \mathbb{P}(A_t = i)$ for each non-satisficing arm i. First, we decompose the event $\{A_t = i\}$ as

$$\{A_t = i\} \subset \{t = i\} \cup \big\{A_t = i, \hat{\mu}_i(t) > S - \tfrac{\Delta_i^S}{2}, t > K\big\} \cup \big\{A_t = i, \hat{\mu}_i(t) \leqslant S - \tfrac{\Delta_i}{2}, t > K\big\} \tag{16}$$

For the first two events we have

$$\sum_{t=1}^{T} \mathbb{P}(t = i) \leqslant 1 \tag{17}$$

and

$$\sum_{t=1}^{T} \mathbb{P}\big\{A_t = i, \hat{\mu}_i(t) > S - \tfrac{\Delta_i^S}{2}, t > K\big\} \leqslant \frac{8}{(\Delta_i^S)^2}. \tag{18}$$

For the probability of the third event in (16) we have

$$\begin{aligned}
\mathbb{P}\big(A_t = i, \hat{\mu}_i(t) \leqslant S - \tfrac{\Delta_i^S}{2}, t > K\big) &\leqslant \mathbb{P}\big(A_t = i, \hat{\mu}_i(t) \leqslant S - \tfrac{\Delta_i^S}{2}, Z_t\big) \\
&= \mathbb{E}\big(p_{i,t} \mathbb{1}\big\{\hat{\mu}_i(t) \leqslant S - \tfrac{\Delta_i^S}{2}, Z_t\big\}\big) \\
&= \mathbb{E}\Big(\frac{p_{i,t}}{p_{*,t}} p_{*,t} \mathbb{1}\big\{\hat{\mu}_i(t) \leqslant S - \tfrac{\Delta_i^S}{2}, Z_t\big\}\Big) \\
&\leqslant \mathbb{E}\Big(\frac{\psi(|S - \hat{\mu}_*(t)|)}{\psi\big(\frac{\Delta_i^S}{2}\big)} p_{*,t} \mathbb{1}\big\{\hat{\mu}_i(t) \leqslant S - \tfrac{\Delta_i^S}{2}, Z_t\big\}\Big) \\
&\leqslant \frac{1}{\psi\big(\frac{\Delta_i^S}{2}\big)} \mathbb{E}\big(\psi(|S - \hat{\mu}_*(t)|)\, p_{*,t}\, \mathbb{1}\{Z_t\}\big) \\
&\leqslant \frac{1}{\psi\big(\frac{\Delta_i^S}{2}\big)} \mathbb{E}\big(\psi(|S - \hat{\mu}_*(t)|)\, \mathbb{1}\{A_t = *, Z_t\}\big).
\end{aligned}$$

Summing up the expectation value over all t yields

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{E}\big(\psi(|S-\hat{\mu}_*(t)|)\,\mathbb{1}\{A_t=*, Z_t\}\big) &\leqslant \sum_{t=1}^{T} \mathbb{E}\big(\psi(|S-\hat{\mu}_{*,t}|)\,\mathbb{1}\{\hat{\mu}_{*,t}\leqslant S\}\big) \\
&= \sum_{n=1}^{T} \int_0^{\infty} \mathbb{P}\big(\psi(|S-\hat{\mu}_{*,n}|)\,\mathbb{1}\{\hat{\mu}_{*,n}\leqslant S\}\geqslant x\big)\,dx \\
&= \sum_{n=1}^{T} \int_0^{\psi(0)} \mathbb{P}\big(\psi(|S-\hat{\mu}_{*,n}|)\,\mathbb{1}\{\hat{\mu}_{*,n}\leqslant S\}\geqslant x\big)\,dx \\
&\quad + \sum_{n=1}^{T} \int_{\psi(0)}^{\infty} \mathbb{P}\big(\psi(|S-\hat{\mu}_{*,n}|)\,\mathbb{1}\{\hat{\mu}_{*,n}\leqslant S\}\geqslant x\big)\,dx \\
&= \sum_{n=1}^{T} \int_0^{\psi(0)} \mathbb{P}\big(\hat{\mu}_{*,n}\leqslant S\big)\,dx \\
&\quad + \sum_{n=1}^{T} \int_{\psi(0)}^{\psi(\infty)} \mathbb{P}\big(\hat{\mu}_{*,n}\leqslant S-\psi^{-1}(x)\big)\,dx,
\end{aligned}$$

noting that, since ψ is increasing, for $x\leqslant\psi(0)$ the inequality $\psi(S-\hat{\mu}_{*,n})\mathbb{1}\{\hat{\mu}_{*,n}\leqslant S\}\geqslant x$ is equivalent to $\mathbb{1}\{\hat{\mu}_{*,n}\leqslant S\}=1$, while for $x\geqslant\psi(0)$ it is equivalent to $\psi(S-\hat{\mu}_{*,n})\geqslant x$. Further, note that if $x>\psi(\infty):=\lim_{y\to\infty}\psi(y)$ then the integrand is equal to 0.

We continue with the analysis of the same term and obtain

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{E}\big(\psi(|S-\hat{\mu}_*(t)|)\,\mathbb{1}\{A_t=*, Z_t\}\big) &\leqslant \sum_{n=1}^{T} \psi(0)\,\mathbb{P}\big(\hat{\mu}_{*,n}\leqslant S\big) \\
&\quad + \sum_{n=1}^{T} \int_0^{\infty} \mathbb{P}\big(\hat{\mu}_{*,n}\leqslant S-u\big)\,\psi'(u)\,du \\
&\leqslant \sum_{n=1}^{T} \psi(0)\exp\big(-\tfrac{n(\Delta_*^S)^2}{2}\big) \\
&\quad + \sum_{n=1}^{T} \int_0^{\infty} \exp\big(-\tfrac{n(|\Delta_*^S|+u)^2}{2}\big)\,\psi'(u)\,du \\
&\leqslant \frac{2\psi(0)}{(\Delta_*^S)^2} + \int_0^{\infty} \sum_{n=1}^{T} \exp\big(-\tfrac{n(|\Delta_*^S|+u)^2}{2}\big)\,\psi'(u)\,du \\
&\leqslant \frac{2\psi(0)}{(\Delta_*^S)^2} + \int_0^{\infty} \frac{\psi'(u)}{e^{\frac{(|\Delta_*^S|+u)^2}{2}}-1}\,du.
\end{aligned}$$

Finally, by putting everything together, we obtain

$$\begin{aligned} R_T^S &= \sum_{i:\Delta_i^S>0} \mathbb{E}(n_i(T))\,\Delta_i^S \\ &\leqslant \sum_{i:\Delta_i^S>0} \left(\Delta_i^S + \frac{8}{\Delta_i^S} + \frac{\Delta_i^S}{\psi\left(\frac{\Delta_i^S}{2}\right)}\left(\frac{2\psi(0)}{(\Delta_*^S)^2} + \int_0^{+\infty} \frac{2\psi'(x)}{e^{\frac{(|\Delta_*^S|+x)^2}{2}}-1}\,dx\right)\right), \end{aligned}$$

which completes the proof. □

B Proof of Theorem 5

T_K is the number of steps during which we play an unsatisfying policy during the K first episodes of a run of Algorithm 4. Notice that $(T_K)_{K\in\mathbb{N}}$ is a subsequence of the non-decreasing sequence $(T^U(\mathcal{A}_4, M, \sigma, t)_{t\in\mathbb{N}}$, so upper bounding the limit of one implies the same result on the other.

By considering that the exploration episodes are not satisfying we can write the following upper bound for T_K

$$T_K \leqslant L_0 + \sum_{k=1}^{K} \mathbb{1}(\rho_k(\pi_k, s_k) \geqslant \sigma \wedge \rho(\pi_k, s_k) < \sigma)L_k + \sum_{k=1}^{K} \mathbb{1}(\rho_k(\pi_k, s_k) < \sigma)L_k$$

The main observation that makes the proof possible is that if the error between the estimated average reward and the true average reward of a policy is superior to some ε then there exist some state and action such that the estimated reward error is also more than ε. Indeed, in the case of MDP with deterministic transitions, a deterministic and stationary policy induces a Markov Chain with deterministic transitions. Since the transitions of the Markov chain are deterministic, the average reward when starting from some state is exactly the average reward of one of the recurrent classes of the Markov Chain. The recurrent classes of this type of chain are cycles, and their average rewards are the unweighted average over the reward of their edges. Additionally, since for any $(u_1, ..., u_n) \in \mathbb{R}^n$ and $\varepsilon \in \mathbb{R}^+$ we have $\frac{1}{n}\sum_{k=1\ n} u_k > \varepsilon \Rightarrow \exists k \in [\![1, n]\!], u_k > \varepsilon$ we can deduce that an error on the estimated average reward of the policy induces an error of at least the same amount on one of the estimated rewards. In particular, there exists a reward with an approximation error greater than ε among the edges originating from states of the recurrent classes in which the policy leads to.

Now suppose we play a non-satisfying policy π_k at step k. This means that the average reward of the policy is over-estimated by at least Δ_{π_k}. The previous observation indicates that there exists a reward overestimated by at least the same amount among the edges of and recurrent component of the induced Markov Chain. Since we follow the policy for S steps, some state will be visited twice. The transitions of the Markov chain being deterministic, this ensures that all the state of the recurrent class have been visited. Hence, the following implications hold:

$$\begin{aligned} \rho_k(\pi_k) \geqslant \sigma \wedge \rho(\pi_k) < \sigma &\Rightarrow \rho_k(\pi_k) \geqslant \rho(\pi_k) + \Delta_{\pi_k} \\ &\Rightarrow \exists (s,a) \in \mathcal{S}\times\mathcal{A}, r_k(s,a) \geqslant r(s,a) + \Delta_{\pi_k} \wedge N_k(s,a) > N_{k-1}(s,a) \end{aligned}$$

where $r_k(s,a)$ is the estimated average reward at step k for the state s and the action a, and $r(s,a)$ is the true expected reward for the couple (s,a).

We will write $\bar{r}_n(s,a)$ for the empirical average reward for using action a from state s, obtained by averaging the n first samples, and $\Delta_U = \min_{\pi:\Delta_\pi>0} \Delta_\pi$. The previous implication allows bounding the first sum :

$$\begin{aligned}
\sum_{k=1}^{K} \mathbb{1}(\rho_k(\pi_k, s_k) \geqslant \sigma \wedge \rho(\pi_k, s_k) < \sigma) L_k &= S \sum_{k=1}^{K} \mathbb{1}(\rho_k(\pi_k, s_k) \geqslant \sigma \wedge \rho(\pi_k, s_k) < \sigma) \\
&\leqslant S \sum_{k=1}^{K} \mathbb{1}(\exists (s,a) \in \mathcal{S} \times \mathcal{A}, r_k(s,a) \geqslant r(s,a) + \Delta_{\pi_k} \\
&\qquad \wedge N_k(s,a) > N_{k-1}(s,a)) \\
&\leqslant S \sum_{k=1}^{K} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathbb{1}(r_k(s,a) \geqslant r(s,a) + \Delta_{\pi_k} \wedge N_k(s,a) > N_{k-1}(s,a))
\end{aligned}$$

$$\begin{aligned}
\sum_{k=1}^{K} \mathbb{P}(\rho_k(\pi_k, s_k) \geqslant \sigma \wedge \rho(\pi_k, s_k) < \sigma) L_k &= S \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \sum_{k=1}^{K} \mathbb{P}(r_k(s,a) \geqslant r(s,a) + \Delta_{\pi_k} \wedge N_k(s,a) > N_{k-1}(s,a)) \\
&\leqslant S \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \sum_{n=1}^{K} \mathbb{1}(\bar{r}_n(s,a) \geqslant r(s,a) + \Delta_{\pi_k}) \\
&\leqslant S \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \sum_{n=1}^{K} \mathbb{1}(\bar{r}_n(s,a) \geqslant r(s,a) + \Delta_U)
\end{aligned}$$

In the case in which $\rho_k(\pi_k, s_k) < \sigma$ and since we chose π_k to be the policy with the maximal average reward on the estimated MDP M_k, we have in particular that $\rho_k(\pi_*, s_k) < \sigma$. Similarly to the previous case, the average reward of the optimal policy is underestimated by at least Δ^* so there exists an underestimated reward by at least the same amount. In this particular case, we chose to resample each actions of each state so the underestimated action will be played at least once. As the MDP is communicating and the transitions are known at this point of the algorithm, we can bound the number of steps by S^2A.

$$\begin{aligned}\sum_{k=1}^{K}\mathbb{1}(\rho_k(\pi_k,s_k)<\sigma)L_k &\leqslant S^2A\sum_{k=1}^{K}\mathbb{1}(\rho_k(\pi_k,s_k)<\sigma\wedge\rho_k(\pi_*,s_k)<\sigma)\\ &\leqslant S^2A\sum_{k=1}^{K}\mathbb{1}(\exists(s,a)\in\mathcal{S}\times\mathcal{A},r_k(s,a)<r(s,a)-|\Delta_*|\wedge N_k(s,a)>N_{k-1}(s,a))\\ &\leqslant S^2A\sum_{k=1}^{K}\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\mathbb{1}(r_k(s,a)<r(s,a)-|\Delta_*|\wedge N_k(s,a)>N_{k-1}(s,a))\end{aligned}$$

$$\begin{aligned}\sum_{k=1}^{K}\mathbb{P}(\rho_k(\pi_k,s_k)<\sigma)L_k &\leqslant S^2A\sum_{k=1}^{K}\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\mathbb{P}(r_k(s,a)<r(s,a)-|\Delta_*|\wedge N_k(s,a)>N_{k-1}(s,a))\\ &\leqslant S^2A\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\sum_{n=1}^{K}\mathbb{P}(\bar{r}_n(s,a)<r(s,a)-|\Delta_*|)\end{aligned}$$

Finally, assuming the reward distributions are 1-subgaussians, we can deduce a bound on the expectancy of T_k from Chernoff inequalities.

$$\begin{aligned}\mathbb{E}(T_k) &\leqslant S^2A+S\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\sum_{k=1}^{K}\mathbb{P}(\bar{r}_k(s,a)\geqslant r(s,a)+\Delta_{\pi_k})+S^2A\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\sum_{k=1}^{K}\mathbb{P}(\bar{r}_k(s,a)<r(s,a)-|\Delta_*|)\\ &\leqslant S^2A+S\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\sum_{k=1}^{K}\exp\left(\frac{-k\Delta_U^2}{2}\right)+S^2A\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\sum_{k=1}^{K}\exp\left(\frac{-k\Delta_*^2}{2}\right)\\ &\leqslant S^2A+S^2A\frac{2}{\Delta_U^2}+S^3A^2\frac{2}{\Delta_*^2}\\ &= S^2A\left[1+\frac{2}{\Delta_U^2}+SA\frac{2}{\Delta_*^2}\right]\end{aligned}$$