Anytime many-armed bandits

Olivier Teytaud, Sylvain Gelly, Michèle Sebag

Anytime many-armed bandits

Olivier Teytaud, Sylvain Gelly and Michele Sebag

Equipe TAO (Inria), LRI, UMR 8623 (CNRS - Université Paris-Sud),
bât 490 Université Paris-Sud 91405 Orsay Cedex France,
olivier.teytaud@inria.fr, sylvain.gelly@lri.fr,
michele.sebag@lri.fr

Abstract :

This paper introduces the many-armed bandit problem (ManAB), where the number of arms is large comparatively to the relevant number of time steps. While the ManAB framework is relevant to many real-world applications, the state of the art does not offer anytime algorithms handling ManAB problems. Both theory and practice suggest that two problem categories must be distinguished; the easy category includes those problems where good arms have reward probability close to 1; the difficult category includes other problems. Two algorithms termed FAILURE and MUCBT are proposed for the ManAB framework. FAILURE and its variants extend the non-anytime approach proposed for the denumerable-armed bandit and non-asymptotic bounds are shown; it works very efficiently for easy ManAB problems. Meanwhile, MUCBT efficiently deals with difficult ManAB problems.

1 Introduction

One mainstream paradigm for online learning is known as the multi-armed bandit formulated by Lai & Robbins (1985); given n bandit arms with (unknown) reward probabilities p_i, in each time step t the player selects an arm j and receives a reward r_t, where $r_t = 1$ with probability p_j, and $r_t = 0$ otherwise. The goal is to maximize the cumulated reward gathered over all time steps, or minimize the loss incurred compared to the best strategy (playing the arm with maximal reward probability in each time step), referred to as regret.

Indeed, such optimization problems can be solved exactly using dynamic programming approaches when the number N of time steps is known in advance, as shown by Bellman (1957) and Bertsekas (1995). Currently, the multi-armed bandit litterature focuses on anytime algorithms (N is not known beforehand), with good asymptotic bounds on the regret and which are less computationally expensive than the prohibitive dynamic programming approach.

Devised by Auer *et al.* (2001), the so-called Upper Bound Confidence (UCB) algorithms enforce an optimal asymptotic bound on the regret (in $\mathcal{O}(log(N))$) in the stationary case. The non stationary case has also been studied by Kocsis & Szepesvari

(2005) or Hussain *et al.* (2006), respectively considering the adversarial case or abruptly changing environments. Also, Kocsis & Szepesvari (2006) have extended UCB to the case of tree-structured arms, defining the UCT algorithm.

This paper focuses on the case of *many-armed bandits* (ManAB), when the number of arms is large relatively to the relevant number N of time steps (relevant horizon) (Banks & Sundaram (1992); Agrawal (1995); Dani & Hayes (2006)). Specifically, we assume in the rest of the paper that N is not known in advance and that N is at most $\simeq n^2$. It is claimed that the ManAB setting is relevant to many potential applications of online learning. For instance when Wang & Gelly (2007) adapt UCT to build an automatic Go player, the deep regions of the UCT tree can only be frugally explored while they involve an exponential number of moves. Applications such as labor markets, votes, consumers choice, dating, resource-mining, drug-testing (Berry *et al.* (1997)), feature selection and active learning (Cesa-Bianchi & Lugosi (2006)) also involve a number of options which is large compared to the relevant horizon.

The state of the art does not address the anytime ManAB problem (more on this in section 2). On one hand, UCB algorithms boil down to uniform sampling with no replacement when the number of arms is large comparatively to the number of time steps. On the other hand, the failure-based approaches dealing with the denumerable-armed bandit with good convergence rates (Berry *et al.* (1997), section 2.2) require both: (i) prior knowledge on the reward distribution; (ii) the number N of time steps to be known in advance. They provide highly suboptimal strategies when N is badly guessed or when the p_i are far from 1.

As a first contribution, this paper extends the failure-based algorithms devised for the denumerable-armed bandit by Berry *et al.* (1997) to the anytime framework, preserving their good convergence properties. The resulting algorithm, termed FAILURE, however suffers from the same limitations as all failure-based algorithms when the highest reward probabilities are well below 1. Therefore, two settings are distinguished, the *Easy ManAB* (EManAB) where the reward probabilities p_i are uniformly and independently distributed in $[0, 1]$, and the *Difficult ManAb* (DManAB) where the p_i are uniformly distributed in $[0, \epsilon]$ with $\epsilon < 1$. It must be emphasized that the DManAB setting is relevant to real-world applications; e.g. in the News Recommendation application (Hussain *et al.* (2006)) the optimal reward probabilities might be significantly less than one.

We thus propose a second algorithm, inspired from the Meta-Bandit approach first described by Hartland *et al.* (2006) and referred to as MUCBT. While MUCBT robustly and efficiently deals with all ManAB problems including the difficult ones, it is outperformed by FAILURE on Easy ManAB problems.

This paper is organized as follows. Section 2 briefly introduces the multi-armed bandit background, presenting the UCB algorithms (Auer *et al.* (2001)) and the failure-based algorithms devised for the denumerable-armed bandit in the non-anytime case (Berry *et al.* (1997)). Section 3 extends the failure-based algorithms to the anytime framework, considering both easy and difficult ManAB settings. Section 4 presents the MUCBT algorithms specifically devised for the Difficult ManAB problems. Section 5 reports on the comparative validation of the presented algorithms compared to the state of the art, and discusses which algorithms are best suited to which settings. The paper concludes with some perspectives for further research.

2 State of the art

This section briefly introduces the notations and UCB algorithms, referring the reader to Auer *et al.* (2002) for a comprehensive presentation. The state of the art related to infinitely many-armed bandits is then presented.

2.1 Any-time MAB

A multi-armed bandit involves n arms, where the i-th arm is characterized by its reward probability p_i. In each time step t, the player or the algorithm selects some arm $j = a_t$; with probability p_j it gets reward $r_t = 1$, otherwise $r_t = 0$. The loss after N time steps, or regret, is defined as $Np^* - \sum_{t=1}^{N} r_t$, where p^* is the maximal reward probability among $p_1, \ldots, p_n$.

Two indicators are maintained for each i-th arm: the number of times it has been played up to time t, noted $n_{i,t}$ and the average corresponding reward noted $\hat{p}_{i,t}$. Subscript t is omitted when clear from the context.

The so-called UCB1 algorithm selects in each step t the arm which maximizes an exploration vs exploitation tradeoff

$$\hat{p}_{j,t} + \sqrt{\frac{2 \log \sum_k n_{k,t}}{n_{j,t}}}$$

The first term $\hat{p}_{j,t}$ clearly reflects the exploitation bias (select the arm with optimal average reward); the second term (select arms which have been played exponentially rarely) corresponds to the exploration bias. The asymptotic bound on the regret in UCB1 is $\mathcal{O}(\log(N))$ where N is the number of time steps, which is known to be optimal after Lai & Robbins (1985).

Exploitation		Exploration	
$\hat{p}_j$ +	$\frac{log \sum_k n_{k,t}}{n_{j,t}}$	+ $\sqrt{\frac{2log \sum_k n_{k,t}}{n_{j,t}}}$	UCB1
		$\sqrt{\frac{v_{j,t} log \sum_k n_{k,t}}{n_{j,t}}}$	UCB-Tuned
		$\sqrt{\frac{v_{j,t} log \sum_k n_{k,t}}{n_{j,t}}}$	KUCBT
		$\sqrt{\frac{c log \sum_k n_{k,t}}{n_{j,t}}}$	cUCB

Table 1: UCB Algorithms, where $v_{j,t}$ denotes the maximum between 0.01 and the empirical variance of the reward for the j-th arm. The max with 0.01 is intended to avoid null estimated variance that could lead to reject definitively an arm.

The key point is to adjust the exploration strength. Several variants have been proposed, summarized in Table 1 :
Based on intuitively satisfactory ideas and taking into account the empirical variance of the reward associated to each arm, the UCB-Tuned (UCBT) proposed by Auer *et al.* (2002) often outperforms UCB1, though with no formal proof of improvement.
KUCBT is similar to UCBT but without the non-variance-based term

$\log(\sum_k n_{k,t})/n_{j,t}$. We additionally consider the cUCB variant, using an explicit constant c to bias the selection toward exploitation as being more appropriate when the number of arms increases and the time horizon decreases.

2.2 Denumerable-Armed Bandits

The case of denumerable-armed bandits (DAB) have been studied by Berry *et al.* (1997), establishing an upper bound $2\sqrt{N}$ on the regret when the number N of time steps is known (non any-time setting).

Berry *et al.* (1997) introduce several algorithms:

- **The k-failure strategy**. When the current arm fails for k successive time steps, this arm is never tested again and a new arm is selected. While this strategy converges toward the optimal success rate, the regret decreases slowly with N ($\mathcal{O}(N/logN)$).
- **The α-rate strategy**. When the average reward of the current arm falls below some $\alpha < 1$ threshold, a new arm is selected. This strategy does not necessarily converge toward the optimal success rate.
- **The m-run strategy**. This strategy firstly runs the 1-failure strategy until either selecting the m-th arm, or until a sequence of m wins occurs; at this point, the m-run strategy plays the arm with best average reward until the end of the N steps. When m is of the order of $\sqrt{N}$, the m-run strategy reaches the optimal success rate and the regret decreases with N as $2\sqrt{N}$; otherwise, the m-strategy does not necessarily converge to the optimal success rate as N increases, as it almost surely stops exploration after having tested finitely many arms.
- **The non-recalling m-run strategy**. This strategy likewise runs the 1-failure strategy until a sequence of m wins occurs, and it thereafter plays the current arm until the end. Like the m-run strategy, the non-recalling m-run strategy reaches the optimal success rate with regret $2\sqrt{N}$ for $m \simeq \sqrt{N}$.
- **The m-learning strategy**. This strategy uses 1-Failure during the first m steps, and then uses the empirically best arm during the remaining $N - m$ steps.

3 Any-time Denumerable-Armed Bandits

This section extends the approaches presented by Berry *et al.* (1997) to the anytime setting, where the algorithm does not depend on the number N of time steps.

3.1 The Easy ManAB setting

Let us first consider the easy setting where the reward probabilities p_i are independently uniformly distributed in $[0, 1]$. Then we show:

Theorem 1: EManAB setting. *There exists an any-time strategy for the denumerable-armed bandit with expected failure rate bounded by* $O(1/\sqrt{N})$.

Proof: Let $\alpha > 1$ be a parameter and let us define the family of time intervals $I_i = [i^\alpha, (i+1)^\alpha[$. Let us consider the strategy defined by playing the m-run strategy with $m = \sqrt{i^\alpha}$ on interval I_i (independently from what has been done during the previous time intervals).

By construction this strategy is any-time; it does not depend on N. For a given N, let k be such that $N \in I_k$. On each interval I_i, the expected number of failures incurred by the $\sqrt{i^\alpha}$ strategy is $\mathcal{O}(\sqrt{i^\alpha})$ after Berry *et al.* (1997), Thm 4. Therefore, the expected number of failures until the end of I_k is at most $\mathcal{O}(\sum_{i=1}^{k} \sqrt{i^\alpha})$.

It comes that the number of failures of the considered strategy up to time step N is upper bounded by $k \times \sqrt{k^\alpha} = k^{1+\alpha/2}$. The failure rate thus is upper bounded by $k^{1+\alpha/2}/N \leq k^{1+\alpha/2}/k^\alpha = \mathcal{O}(1/\sqrt{N})$. □

We point out that another algorithm can be used for the same result. With the same proof as above using the properties of m-learn strategies instead of the properties of m-run strategy (Berry *et al.* (1997)), we show the same result for the following algorithm at step t:

- if $\lfloor \sqrt{t}/\log(t) \rfloor > \lfloor \sqrt{t-1}/\log(t-1) \rfloor$, choose the arm with lowest index which has never failed (this is the FAILURE algorithm);
- otherwise, use the arm which has the best empirical success rate among all arms that have been rejected by FAILURE.

This algorithm, termed "MLEARN" in the rest of this paper, is nicer as it has no free parameter; it will be used in experiments.

3.2 The Difficult ManAB setting

Let us consider the difficult ManAB setting, where the reward probabilities p_i are uniformly distributed in $[0, \epsilon]$ for $\epsilon < 1$. As shown by Berry *et al.* (1997), for some given m depending on ϵ and N, the m-run strategy reaches an expected failure rate $\mathcal{O}(\sqrt{\epsilon/N})$.

In this section, the above result is extended to the case where N and ϵ are unknown.

Theorem 2: DManAB setting. *Let us assume that the reward distribution is such that there exists a constant* $C > 0$ *which satisfies*

$$\forall \epsilon \in]0, 1[, P(p_1 > \sup p_i - \epsilon) \geq \min(1, C\epsilon) \tag{1}$$

Then there exists an any-time strategy for the denumerable armed bandit with expected failure rate bounded by $\tilde{\mathcal{O}}(N^{-\frac{1}{4}}/C)$ *(with* $a = \tilde{\mathcal{O}}(b)$ *the notation for* $\exists k > 0; a = \mathcal{O}(b(\log(b))^k)$*).*

Note that the bound is uniform in the distribution (i.e. all constants hidden in the $O(.)$ are independent of the distribution) under assumption (1). Assumption (1) typically holds when the reward probabilities are uniformly distributed in $[0, \epsilon]$.

Proof: The proof is constructive, based on the algorithm described in Table 2. Indices $n_{i,t}$ and $w_{i,t}$ respectively stand for the number of times the i-th arm is played (resp.,

wins) up to time t. Two sequences $(s_n)_{n\in\mathbb{N}}$ and $(k_n)_{n\in\mathbb{N}}$, with s increasing and k non-decreasing are used.

1. Init : $n_{i,0} = w_{i,0} = 0$ forall i.
2. Loop : For $t = 1;\ true;\ t = t + 1;$
   If $t = s_i$ for some i, Exploration(t).
   Else ($t \in]s_i, s_{i+1}[$), Exploitation(t).

Exploration(t) $\qquad t = s_i$
   Select $j = argmin\{n_\ell, \ell \in [1, k_i[\}$ In case of ties, prefer smallest j.
   Receive r_t
   $n_{j,t+1} = n_{j,t} + 1 \quad ; \quad w_{j,t+1} = w_{j,t} + r_t$
   $n_{i,t+1} = n_{i,t} \quad ; \quad w_{i,t+1} = w_{i,t}\ forall\ i \neq j$

Exploitation(t) $\qquad t \in]s_i, s_{i+1}[$
   Select $j = argmax\{w_\ell/n_\ell, \ell \in [1, k_i[\}$ In case of ties, prefer smallest j.
   Receive r_t
   $n_{j,t+1} = n_{j,t} + 1 \quad ; \quad w_{j,t+1} = w_{j,t} + r_t$
   $n_{i,t+1} = n_{i,t} \quad ; \quad w_{i,t+1} = w_{i,t}\ forall\ i \neq j$

Table 2: DManAB Algorithm

Let us define ϵ_i the maximal reward estimation error after i exploration steps:

$$\epsilon_i = argmax\{|\frac{w_{j,t}}{n_{j,t}} - p_j|, j \in [1, k_i], t = s_i + 1\}$$

Let t be a time step in the i-th epoch ($t \in [s_i, s_{i+1}[$). Let ϵ_t define the maximal ϵ_i such that $t < s_{i+1}$. Up to time t, i) the number of exploration steps so far is i; ii) the arms which have been played are included in $[1, k_i]$; iii) the maximal estimation error so far is ϵ_i.

For the particular two sequences below, we shall show that the algorithm is efficient, i.e. ϵ_t goes to 0. Let

$$k_n = \lfloor n^\alpha \rfloor, \ \ \alpha = \frac{1}{3} \tag{2}$$
$$s_n = \sum_{i=1}^{n} \lfloor 1 + i^\gamma \rfloor, \ \ \gamma = \frac{1}{3}$$

Step 1: Fast convergence of ϵ_t to 0.
Let t be the current time step, belonging to the i-th epoch ($t \in [s_i, s_{i+1}[$). Let j be an arm belonging to the set of arms explored up to now ($j \in [1, k_i[$). Then, as all arms have been played an equal number of times during the i exploration steps:

$$n_{j,t} \geq \lfloor i/k_i \rfloor \tag{3}$$

After the Hoeffding bound, for all arms $j \in [1, k_i[$ and $t \geq s_i$, it comes

$$P(|\frac{w_{j,t}}{n_{j,t}} - p_j| > \epsilon) \leq \exp(-2\lfloor i/k_i \rfloor \epsilon^2) \tag{4}$$

$$P(\sup_{j<k_i} |\frac{w_{j,t}}{n_{j,t}} - p_j| > \epsilon) \leq k_i \exp(-2\lfloor i/k_i \rfloor \epsilon^2)$$

$$\text{and therefore } \mathbb{E} \sup_{j<k_i} |\frac{w_{j,t}}{n_{j,t}} - p_j| = O(\sqrt{k_i \log(k_i)/i}) \tag{5}$$

$$\text{and therefore } \mathbb{E} \sup_{j<k_i} |\frac{w_{j,t}}{n_{j,t}} - p_j| = \tilde{O}(i^{(\alpha-1)/2}) \tag{6}$$

Eq. (5) follows from the lemma below ((Devroye *et al.*, 1997, chap 12, p 208)):

$$P(Z < 0) = 0 \wedge P(Z \geq \epsilon) \leq c \exp(-2m\epsilon^2) \Rightarrow EZ \leq \sqrt{\log(ce)/2m}$$

Eq. (6) states that ϵ_t converges to 0 like $\mathcal{O}(i^{-1/3}log(i))$ ie like $\tilde{\mathcal{O}}(i^{-1/3})$.

Step 2: exploitation is efficient.
Let R_N denote the sum of all rewards up to time step N, and let us consider the expectation of R_N. Let us assume further that N belongs to the n-th epoch ($N \in [s_n, s_{n+1}[$). It comes:

$$\mathbb{E}R_N \geq \mathbb{E} \sum_{i=1}^{n-1} \left(\underbrace{r_{s_i}}_{exploration} + \underbrace{\sum_{t=s_i+1}^{s_{i+1}-1} r_t}_{exploitation} \right) \tag{7}$$

Let p_i^* denote the reward probability of the arm selected during the i-th exploitation epoch (being reminded that a single arm is played during $]s_i, s_{i+1}[$), and let p_i^{**} denote the maximal reward probability p_j for j in $[1, k_i[$. Note $\mathbb{E}_n$ the expectation operator, conditionally to[1] the $(p_i)_{i\in\mathbb{N}}$ and to the exploration (formally, conditionally to all the p_i for $i \in \mathbb{N}$ and to all the r_t for $t = s_1, \ldots, s_n$).

$$\begin{aligned} \frac{1}{N}\mathbb{E}_n R_N &\geq \frac{1}{N} \sum_{i=1}^{n-1} (i^\gamma {p^*}_i) \\ &\geq \frac{1}{N} \sum_{i=1}^{n-1} (i^\gamma (p_i^{**} - 2\epsilon_i)) \end{aligned}$$

by definition of ϵ_i. Let us note $S_n = \frac{1}{N} \sum_{i=1}^{n-1} (1 + i^\gamma) p_i^{**}$.

$$\begin{aligned} \mathbb{E}_n S_n - \frac{1}{N}\mathbb{E}_n R_N &= O(\frac{1}{N} \sum_{i=1}^{n-1} (1 + i^\gamma \epsilon_i)) \\ &= O(n/N + \frac{1}{N}. \sum_{i=1}^{n-1} (i^\gamma . i^{(\alpha-1)/2} . log(i)) \qquad (8) \\ &= \tilde{O}(n/N) \qquad (9) \end{aligned}$$

[1] Recall that the p_i are i.i.d random variables.

almost surely thanks to step 1. Let $\mathbb{E}_p$ denote the conditional expectation with respect to $p_i, i \in \{1, 2, 3, \dots\}$; as the constants in the $O(.)$ notation are universal constants, we can therefore take the expectation of eq. (9) with respect to the exploration (keeping the conditioning with respect to the p_i), and get:

$$\begin{array}{ll} \mathbb{E}_p[\mathbb{E}_n S_n - \frac{1}{N}\mathbb{E}_n R_N] & = E_{explor}[\mathbb{E}_n S_n - \frac{1}{N}\mathbb{E}_n R_N] = \mathbb{E}_{explor}[\tilde{\mathcal{O}}(n/N)] = \tilde{\mathcal{O}}(n/N) \\ \text{hence } \frac{1}{N}\mathbb{E}_p R_N & \geq p_n^{**} - \tilde{O}(n/N) \end{array} \tag{10}$$

since $\mathbb{E}_p S_n \leq \frac{1}{N}\sum_{i=1}^{n-1}(1+i^\gamma)p_i^{**} \leq p_n^{**}$ by construction.

Step 3: exploration is sufficient. It remains to lower-bound S_n, which depends on the expectation of the maximum p_j for $j \in [1, k_i[$, where p_j are iid random variables such that eq. (1) holds. Noting as above $p_i^{**} = max\{p_j,\ j \in [1, k_i[\}$, and letting $p_* = \sup p_i$, after eq. (1) it comes:

$$\begin{array}{ll} \mathbb{E}[p_* - p_i^{**}] & = \int P(p_* - p_i^{**}) > t)dt = \int \Pi_{j=1}^{k_i-1} P(p_* - p_j > t)dt \\ & = \int P(p_* - p_1 > t)^{k_i-1}dt \\ & = \int (1 - P(p_* - p_1 < t))^{k_i-1}dt \\ & < \int_0^{1/C}(1 - Ct)^{k_i}dt \end{array} \tag{11}$$

hence

$$\mathbb{E}p_i^{**} \geq p_* - O(1/(C.k_i)). \tag{12}$$

Summing eq. (12) for $i \in [1, n]$ leads to

$$S_n \geq p_* - \frac{1}{NC}O(\sum_{i=1}^{n} i^\gamma/k_i) \geq p_* - O(\frac{n}{NC}) \tag{13}$$

Eqs (13) and (10) together lead to

$$\begin{array}{rcl} \frac{1}{N}\mathbb{E}_p R_N & \geq & p_* - O(1/n^{(\alpha-1)/2}) - O(n/CN) \\ & \geq & p_* - O(N^{-1/4}/C) \end{array}$$

which concludes the proof. □

4 Algorithms For Many-Armed Bandits

The theoretical analysis in the previous section suggests that the easy and difficult ManAB settings should be handled through different algorithms. Accordingly, this section presents two FAILURE variants adapted from the failure algorithms introduced in section 2.2. The FPU algorithm inspired by Wang & Gelly (2007) and the MUCBT algorithm inspired by Hartland *et al.* (2006) are last presented.

4.1 The FAILURE and FAILUCB Algorithms

The 1-failure algorithm previously defined for the denumerable-armed bandit is adapted to the ManAB setting in two ways, respectively referred to as FAILURE and FAILUCB.

In both cases, the algorithm plays the current arm until it fails; on failure, one selects the first arm which has never been tested if it exists.

After all arms have been played at least once, FAILURE greedily selects the arm with best estimated reward; FAILUCB uses KBCBT (Table 1). Indeed, FAILURE offers no guarantee to converge to the best success rate as N goes to infinity; however, such a poor asymptotic behaviour is irrelevant in the considered framework since N remains comparable to n or n^2.

4.2 The First Play Urgency Algorithm

The First Play Urgency (FPU) algorithm was first defined in MoGo by Wang & Gelly (2007), to handle a large number of tree-structured arms. Formally, the selection criterion used in UCT (Table 1) is replaced by:

$$V_j = \begin{cases} \hat{p}_{j,t} + \sqrt{2 f_{FPU} \log(\sum_k n_{k,t})/n_{j,t}} & \text{if } n_{j,t} > 0 \\ c_{FPU} & \text{otherwise} \end{cases}$$

(other formula, taking into account variance-terms, are proposed in Wang & Gelly (2007)) It is worth noting that for $f_{FPU} = 0$ and $c_{FPU} = 1$, the FPU algorithm coincides with the FAILURE one.

4.3 The Meta-UCBT Algorithm

We last define the meta-bandit algorithm MUCBT to deal with the ManAB setting. MUCBT is inspired from the meta-bandit algorithm devised by Hartland *et al.* (2006), which won the Exploration vs Exploitation Challenge defined by Hussain *et al.* (2006). However, the EE Challenge focuses on the extension of the many-armed bandit to non-stationary environments, where the meta-bandit was in charge of handling the change point detection epochs.

Quite the opposite, MUCBT is a recursive meta-bandit, where the first meta-bandit decides between the best empirical arm and all other arms, the second meta-bandit decides between the second best arm and all other arms, and so forth (Fig. 1, left).

A variant of the MUCBT algorithm, referred to as MUCBT-k, uses the first meta-bandit to decide between the first best $k-1$ arms, and the others, the second meta-bandit to decide between the next best $k-1$ arms, and the remaining arms, and so forth (Fig 1, right).

Formally, w_i (respectively ℓ_i) denotes the number of wins (resp. losses) with the i-th arm up to the current time step. Algorithms MUCBT and MUCBT-k are specified above, (Algs. 1 & 2), where each algorithm chooses arm a_t at time step t, and t_i is the number of time steps (previous to t) where the chosen arm is greater than i ($t_i = |\{t' \leq t; a_{t'} \geq i\}|$.

Algorithm 1 MUCBT

Input: a (possibly infinite) number of arms.
Initialize $w_j = 0$, $\ell_j = 0$ and $t_j = 0$ for all j.
for $t = 1;\quad true;\quad t \leftarrow t + 1$ **do**
  Sort arms by decreasing $w_j/(w_j + \ell_j)$ (with $0/0 = -\infty$ by convention).
  for $i = 1;\quad true;\quad i \leftarrow i + 1$ **do**
    Compute $w'_i = \sum_{j>i} w_j$ and $\ell'_i = \sum_{j>i} \ell_j$.
    $V_i = w_i/(w_i + \ell_i) + \sqrt{2\log(t_i)/(w_i + \ell_i)}$
    $V'_i = w'_i/(w'_i + \ell'_i) + \sqrt{2\log(t_i)/(w'_i + \ell'_i)}$
    if $V_i > V'_i$ **then**
      break
    end if
  end for
  Play arm i ($a_t = i$).
  If win $w_i \leftarrow w_i + 1$ else $l_i \leftarrow l_i + 1$
  $\forall j \leq i, t_j \leftarrow t_j + 1$.
end for

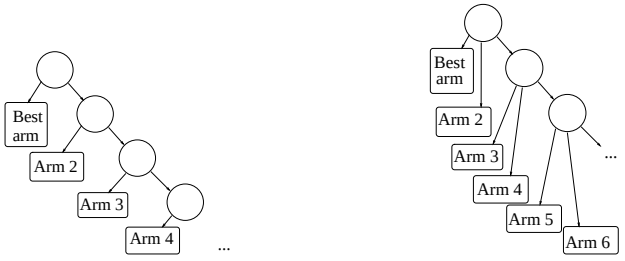

Figure 1: MUCBT algorithm (left) and MUCBT-3 as an example of the MUCBT-k algorithm (right).

Algorithm 2 MUCBT-k

Input: a (possibly infinite) number of arms; parameter $k > 1$.
Initialize $w_j = 0$, $\ell_j = 0$ and $t_j = 0$ for all j.
for $t = 1; \quad true; \quad t \leftarrow t + 1$ **do**
  Sort arms by decreasing $w_j/(w_j + \ell_j)$ (with $0/0 = -\infty$).
  for $i = 1; true; i \leftarrow i + k - 1$ **do**
    Compute $w'_i = \sum_{j>i} w_j$ and $\ell'_i = \sum_{j>i} \ell_j$.
    for $j = i; j \leq i + k - 2; j \leftarrow j + 1$ **do**
      $W_j = w_j/(w_j + \ell_j) + \sqrt{2\log(t_i)/(w_j + \ell_j)}$
    end for
    $u = \mathrm{argmax}_{j \in [[i, i+k-2]]} W_j$
    $V_i = W_u$
    $V'_i = w'_i/(w'_i + \ell'_i) + \sqrt{2\log(t_i)/(w'_i + \ell'_i)}$
    if $V_i > V'_i$ **then**
      break
    end if
  end for
  Play arm u ($a_t = u$).
  If win $w_u \leftarrow w_u + 1$ else $l_u \leftarrow l_u + 1$
  $\forall j \leq u, t_j \leftarrow t_j + 1$.
end for

5 Experimental Validation

This section reports on the experimental validation of the presented algorithms and discusses which algorithm is best suited to the different settings considered.

5.1 Experimental Setting

Considering a number of bandit arms n ranging in $[20, 200]$, artificial bandit problems are generated by drawing iid reward probabilities p_i i) in $[0, 1]$ (Easy ManAB setting) and in $[0, \epsilon]$ for some ϵ (Difficult ManAB setting).

All defined algorithms, including the FAILURE and MUCBT algorithms presented in the paper and the baseline UCB variants, are tested against these problems.

The comparisons refer to three main regimes, depending on the relationship of the number N of time steps and the number n of bandit arms. The standard multi-armed bandit case, referred to as long-run regime, is when $N >> n$; the medium regime is when $N \simeq n^2$; and the short-run regime is when N is circa 2 ou 3 times n.

The performance of each algorithm is given by its average regret per time step.

5.2 Experimental Results

Fig. 2 displays the regret per time step against the number of time step, comparing MUCBT-k for various values of k to the UCB variants (UCB1, UCBT, KUCBT, cUCBT) in an Easy ManAB setting. Note that all UCB variants behave identically in

the considered framework; indeed, for $N \leq n$, UCB-variants receive a reward $r_t = 1$ with probability exactly $\frac{1}{2}$.

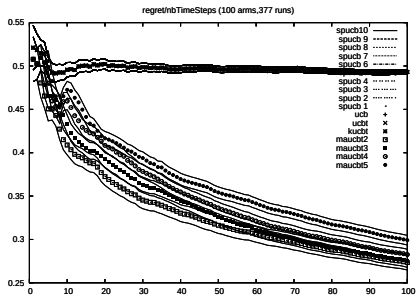

Figure 2: Easy ManAB setting: Comparing UCB variants with MUCBT-k with $n = 100$ arms and $N = 100$ time steps. The average regret per time step is plotted against the total number of time steps. All UCB variants get a reward with probability $\frac{1}{2}$, and are significantly outperformed by MUCBT-k. No significant differences among the various values of k is found in this setting.

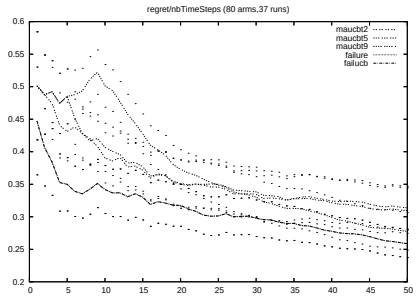

Figure 3: Easy ManAB setting. Experimental results with $N = 50$, for $n > N$ (infinitely many arms). Dotted lines around curves are $\pm$ standard deviations. Algorithms are ordered wrt their horizon performance (best strategies at the bottom). Only the five best algorithms are presented.All UCB variants are outperformed by MUCBT and FAILURE.

A first remark is that, while standard UCB algorithms are optimally suited to the long-run regime ($N >> n^2$), they do not handle efficiently the non-asymptotic cases including the medium-run ($N \simeq n^2$) and short-run $N \simeq 2n, 3n$) regimes.

In the medium-run regime, the failure-based algorithms are optimal in the Easy ManAB case, when the reward probabilities are uniform in $[0, 1]$; in the Difficult ManAB case, the MUCBT algorithms empirically outperform the failure algorithms.

Lastly, the MUCBT algorithms are well suited to the short-run regime.

5.3 Discussion

Failure algorithms, based on theoretical investigations in Berry *et al.* (1997), are very efficient for EManAB. Some variants proposed in this paper work very efficiently also

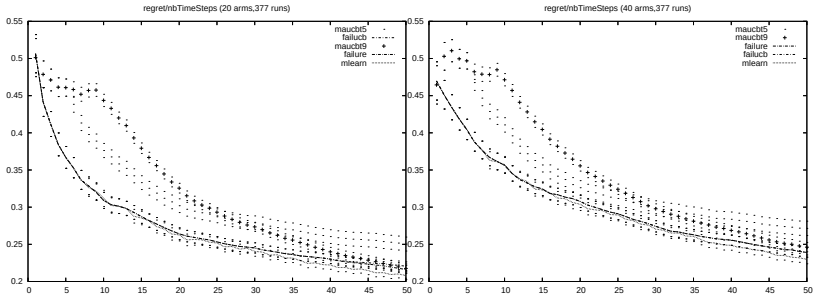

Figure 4: *Easy ManAB setting,* $N \lesssim 3n$. Experimental results with $n = 20$ and $n = 40$ arms respectively, both with $N = 1, \ldots, 50$. Dotted lines around curves are $\pm$ standard deviations. Algorithms are ordered wrt their horizon performance (best strategies at the bottom). Only the five best algorithms are presented.MUCBT variants have a very similar behavior. FAILURE and FAILUCB are exactly equal and all variants of UCB (UCB, KUCB, cUCB) are exactly equal when $N \leq n$. For moderately large N ($N \geq 3n$) the best algorithms are firstly the FAILUREvariants and secondly MUCBT.

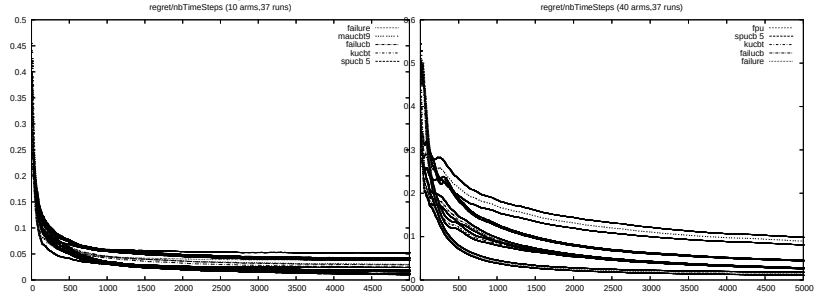

Figure 5: *Easy ManAB setting,* larger horizons. Experimental results with $n = 10, n = 40$ arms respectively, and $N = 5000$. Legends and set-up as in figure 4 (only the best five algorithms wrt horizon performance are presented). We see the strong influence of n; the number of time steps for which FAILURE algorithms outperform baseline UCB-variants is at least $\Omega(n^2)$, in agreement with theory.

in the anytime case and for n finite with $N \simeq n^2$. FAILURE and its variants are experimentally very impressive in experiments, until at least 5000 time-steps, when $n \geq 40$. FAILUCB combines (i) the asymptotic optimality of UCB1 (Auer *et al.* (2001)) when the number of arms is finite and small in front of the horizon; (ii) the non-asymptotic very good behavior of failure-based algorithms.

Also, **FPU**, in particular in its optimal parametrization is close to FAILURE (converging to FAILUCB and UCBT when N increases). Therefore, the mathematical analysis of FPU is very related to the joint analysis of FAILURE and of UCB.

Meta-bandit algorithms MUCBT inspired by Hartland *et al.* (2006) outperform baseline UCB-variants and FPU in all ManAB settings considered here; they also strongly outperform FAILURE and FAILUCB in the DManAB case (figure 6, with reward probabilities in $[0, \frac{1}{4}]$). For cases like news-selection (for which many news are not interesting for the reader) or game-tree-search (for which many moves are stupid moves), small probabilities are a natural case. As for FAILUCB, we point out that FPU

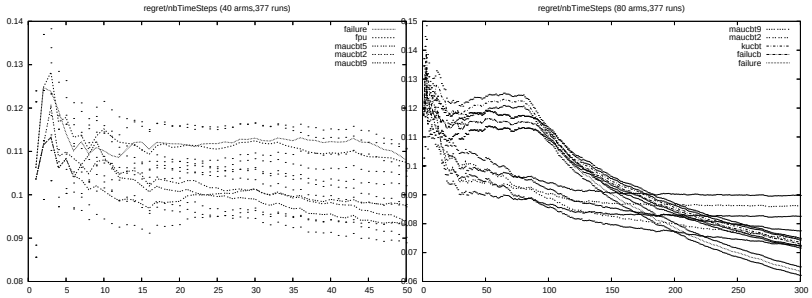

Figure 6: *Difficult ManAB setting.* Experimental results with $n = 40$ arms and $N = 50$ time steps (left), $n = 80$ arms and $N = 300$ time steps (right); legends and set-up are as in other figure, but here probabilities p_i of reward are uniform in $[0, \frac{1}{4}]$ instead of $[0, 1]$. Within this less favorable framework, MUCBT is significantly more robust than FAILURE and variants (MLEARN, FAILUCB); while FPU outperforms FAILURE, it is still dominated by MUCBT. MUCBT-2 (usually the best MUCBT) dominates the other algorithms until $N \simeq 90$ time steps for $n = 40$ (only 50 time-steps presented) and until $N \simeq 190$ time steps for $n = 80$.

was designed for specific purposes (UCT-adaptation) and the efficiency of MUCBT in front of FPU is not ensured for these specific cases.

UCBT and other variance-based extensions of UCB1 (Auer *et al.* (2001); Audibert *et al.* (2006)) significantly improve on the baseline UCB1 in the considered setting. This holds for algorithms considered in isolation or as subroutines for MUCBTor FAILUCB.

6 Conclusion

In summary, some recommendations based on theoretical and practical arguments can be formulated regarding anytime many-armed bandit-problems:

- Use FAILURE when the reward probability distribution is easy and if $N \lesssim n^2$.
- Use MUCBT when the reward probability distribution might be very bad and if $N \lesssim 3n$.
- Always use variance-based UCB-algorithms instead of baseline UCB-algorithms.
- FPU (mainly used for UCT) is a trade-off between FAILURE and MUCBT (Wang & Gelly (2007)).

From the mathematical point of view, we conclude that the algorithms that are proved optimal (within $\sqrt{2}$-factor) for infinitely many arms and finite horizon, namely m-run strategies, can be extended to an anytime algorithm that is also proved optimal (within a multiplicative factor) in the EManAB case. Additionally, we show that a $N^{-1/4}$ rate can be achieved in the DManAB case without knowledge of the distribution.

Still, both proposed algorithms are somewhat unsmooth: m-run strategies switch from a failure-based exploration to exploitation; and their anytime extensions (section

3.1 and 3.2) switch infinitely often between both behaviours. It is likely that algorithms learning the distribution of reward-probability could be nicer and more efficient.

Acknowledgements

We wish to thank the Z. Hussain, P. Auer, N. Cesa-Bianchi, J. Shawe-Taylor and the Touch-Clarity company for the organization of the Tradeoff between Exploitation and Exploration workshop, and C. Szevespary for many fruitful discussion. We also thank the reviewers for helpful comments. The authors gratefully acknowledge the support of the `Pascal` Network of Excellence IST-2002-506 778.

References

AGRAWAL R. (1995). The continuum-armed bandit problem. *SIAM J. Control Optim.*, **33**(6), 1926–1951.

AUDIBERT J.-Y., MUNOS R. & SZEPESVARI C. (2006). Use of variance estimation in the multi-armed bandit problem. In *NIPS 2006 Workshop on On-line Trading of Exploration and Exploitation*.

AUER P., CESA-BIANCHI N. & FISCHER P. (2002). Finite-time analysis of the multi-armed bandit problem. *Machine Learning*, **47**(2/3), 235–256.

AUER P., CESA-BIANCHI N. & GENTILE C. (2001). Adaptive and self-confident on-line learning algorithms. *Machine Learning Journal*.

BANKS J. S. & SUNDARAM R. K. (1992). Denumerable-armed bandits. *Econometrica*, **60**(5), 1071–96. available at http://ideas.repec.org/a/ecm/emetrp/v60y1992i5p1071-96.html.

BELLMAN R. (1957). *Dynamic Programming*. Princeton Univ. Press.

BERRY D. A., CHEN R. W., ZAME A., HEATH D. C. & SHEPP L. A. (1997). Bandit problems with infinitely many arms. *Ann. Statist.*, **25**(5), 2103–2116.

BERTSEKAS D. (1995). *Dynamic Programming and Optimal Control, vols I and II*. Athena Scientific.

CESA-BIANCHI N. & LUGOSI G. (2006). *Prediction, learning, and games*. Cambridge University Press.

DANI V. & HAYES T. P. (2006). Robbing the bandit: less regret in online geometric optimization against an adaptive adversary. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, p. 937–943, New York, NY, USA: ACM Press.

DEVROYE L., GYÖRFI L. & LUGOSI G. (1997). *A probabilistic Theory of Pattern Recognition*. Springer.

HARTLAND C., GELLY S., BASKIOTIS N., TEYTAUD O. & SEBAG M. (2006). Multi-armed bandits, dynamic environments and meta-bandits. In *NIPS Workshop "online trading of exploration and exploitation*.

HUSSAIN Z., AUER P., CESA-BIANCHI N., NEWNHAM L. & SHAWE-TAYLOR J. (2006). Exploration vs. exploitation challenge. *Pascal Network of Excellence*.

KOCSIS L. & SZEPESVARI C. (2005). Reduced-variance payoff estimation in adversarial bandit problems. In *Proceedings of the ECML-2005 Workshop on Reinforcement Learning in Non-Stationary Environments*.

KOCSIS L. & SZEPESVARI C. (2006). Bandit-based monte-carlo planning. *ECML'06*.
LAI T. & ROBBINS H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, **6**, 4–22.
WANG Y. & GELLY S. (2007). Modification of uct with patterns in monte-carlo go. In *Proceedings of ADPRL'07*.

Appendix

This appendix focuses on the First Play Urgency algorithm, examining the impact of the c_{FPU} constant in the particular context of the Go-program Mogo devised by Wang & Gelly (2007). As noted above, the so-called UCB1 algorithm corresponds to the particular case $c_{FPU} = \infty$. The best constant in the considered framework is $c_{FPU} = 1$.

Table 3: The effect of the constant c_{FPU} in FPU from Wang & Gelly (2007). Experiments with 70000 simulations/move in a UCT-based Monte-Carlo-Go, distinguishing the winning rate with white, with black, and the average winning rate.

FPU constant	Winning Rate for Black Games	Winning rate for White Games	Total Winning Rate
1.4	37% $\pm$ 4.5%	38% $\pm$ 5%	37.5% $\pm$ 3.5%
1.2	46% $\pm$ 5%	36% $\pm$ 5%	41% $\pm$ 3.5%
1.1	45% $\pm$ 3%	41% $\pm$ 3%	43.4% $\pm$ 2.2%
1.0	49% $\pm$ 3%	42% $\pm$ 3%	45% $\pm$ 2%
0.9	47% $\pm$ 4%	32% $\pm$ 4%	40% $\pm$ 2.8%
0.8	40% $\pm$ 7%	32% $\pm$ 6.5%	36% $\pm$ 4.8%