ϵ–First Policies for Budget–Limited Multi-Armed Bandits

Long Tran-Thanh
Archie Chapman
Enrique Munoz de Cote
Alex Rogers
Nicholas R. Jennings
School of Electronics and Computer Science,
University of Southampton,
Southampton, SO17 1BJ, UK.
{ltt08r,acc,jemc,acr,nrj}@ecs.soton.ac.uk

Abstract

We introduce the budget–limited multi–armed bandit (MAB), which captures situations where a learner's actions are costly and constrained by a fixed budget that is incommensurable with the rewards earned from the bandit machine, and then describe a first algorithm for solving it. Since the learner has a budget, the problem's duration is finite. Consequently an optimal exploitation policy is not to pull the optimal arm repeatedly, but to pull the combination of arms that maximises the agent's total reward within the budget. As such, the rewards for *all* arms must be estimated, because any of them may appear in the optimal combination. This difference from existing MABs means that new approaches to maximising the total reward are required. To this end, we propose an ϵ–first algorithm, in which the first ϵ of the budget is used solely to learn the arms' rewards (exploration), while the remaining $1 - \epsilon$ is used to maximise the received reward based on those estimates (exploitation). We derive bounds on the algorithm's loss for generic and uniform exploration methods, and compare its performance with traditional MAB algorithms under various distributions of rewards and costs, showing that it outperforms the others by up to 50%.

1 Introduction

The multi–armed bandit (MAB) is a classical problem in decision theory (Robbins 1952), and presents one of the clearest examples of the trade–off between *exploration* and *exploitation* in reinforcement learning. It models a machine with k arms, each of which has a different and unknown expected reward. The goal of the agent (player) is to repeatedly pull the optimal arm (i.e. the arm with the highest expected reward) to maximise the expected total reward. However, the agent does not know the rewards for each arm, so it must sample them in order to learn which is the optimal one. In other words, in order to choose the optimal arm (exploitation) the agent first has to estimate the mean rewards of all of the arms (exploration). In the standard MAB, this trade–off has been effectively balanced by decision–making policies such as *upper confidence bound* (UCB) and ϵ_n-greedy (Auer, Cesa-Bianchi, and Fischer 2002).

However, this standard MAB gives an incomplete description of the sequential decision–making problem facing an agent in many real–world scenarios. To this end, a variety of other related models have been studied recently, and, in particular, a number of researchers have focussed on MABs with a limited exploration budget. Specifically, they consider cases where pulling an arm incurs a pulling cost, whose currency is not commensurable with that of their rewards. The agent's exploration budget then limits the number of times it can sample the arms, thus defining an initial exploration phase, during which the agent's sole goal is to determine the optimal arm to pull during the subsequent cost–free exploitation phase. As for the standard MAB, several exploration policies have been developed that maximise the long–run expected reward of an agent faced with this type of limit on its exploration, namely: (i) MAB with pure exploration (Bubeck, Munos, and Stoltz 2009); (ii) budgeted MAB (Guha and Munagala 2007); and (iii) MAB with max–loss value–estimation (Antos, Grover, and Szepesvári 2008).

Building on these works, in this paper, we consider a further extension of the MAB problem, in which pulling an arm is costly, and *both* the exploration and exploitation phases are limited by a single budget. This type of limitation is well motivated by several real–world applications. For example, consider a company that aimes to advertise itself online. I so doing, it has a limited budget for renting online advertising banners on any of a number of web sites, each of which charges a different rental price. The company wishes to maximise the number visitors who click on its banners, however, it does not know the click–through rate for banners on each site. As such the company needs to estimate the click–through rate for each banner (exploration), and then to choose the combination of banners that maximises the sum of clicks (exploitation). In terms of the model described above, the price of renting an advertising banner from a website is the pulling cost of an arm, and the click–through rate of a banner on a particular website is the true reward for pulling that arm, which is unknown at the outset of the problem. Now, because the budget is limited, both the exploration and exploitation phases are budget limited as well. This means previous MAB models cannot efficiently deal with this problem.

In order to address this gap, in this paper we introduce a new version of the MAB, a *budget-limited MAB with an overall budget* (i.e. pulling an arm is costly, and both exploration and exploitation phases are budget–limited). The overall budget limit differentiates it from previous models in that the overall number of arms pulled is *finite*. Consequently, the optimal solution is not to repeatedly pull the optimal arm *ad infinitum*, but to pull the combination of arms that maximises the reward and fully exploits the budget. To see this, first suppose the expected rewards for pulling the arms are known. In this case, a MAB with an overall budget limit reduces to an unbounded knapsack problem (Andonov, Poirriez, and Rajopadhye 2000), as follows. Pulling an arm corresponds to placing an item into the knapsack, with the arm's expected reward equal to the item's value and the pulling cost the item's weight. The overall budget is then the weight capacity of the knapsack. Given this, the optimal combination of items for the knapsack problem is also the optimal combination of pulls for our MAB problem. This difference in desired optimal solution from existing MAB problems means that (now faced with unknown rewards), when defining a decision–making policy for our problem, we must be cognizant of the fact that an optimal policy will involve pulling a combination of arms. As such, it is not sufficient to learn the expected reward of only the highest–value arm; we must also learn the other arms' rewards, because they may appear in the optimal combination. Importantly, we cannot simply import existing policies that are based on upper confidence bound or ϵ_n–greedy arguments, because they concentrate on learning only the value of the highest expected reward arm, and so will not work in this setting.

For example, consider a three–armed bandit, with arms X, Y and Z that have true expected reward–pulling cost value pairs $\{9, 4\}$, $\{1.5, 1\}$ and $\{2, 1\}$. Suppose the remaining budget is 15. At this point, the optimal solution is to pull arms X and Z three times each, giving an expected total reward of 33. Now consider the case where the estimates of expected reward for arms Y and Z have been generated using a less efficient exploration policy, and as such are not particularly accurate (whereas the estimate of X is considerably more refined). Specifically, imagine a situation where the estimates are 1.76 and 1.74 for arms Y and Z, respectively. In this case, the proposed best solution would be to pull arms X and Y three times each, which on average returns a suboptimal payment with real expected total reward of 31.5. It is clear from this example that better estimates of the mean reward of all arms would allow the agent to determine the optimal combination of pulls. It is also evident that new techniques must be developed for this new problem, which do consider the combinatorial aspect of the optimal solution to the MAB problem investigated in this paper.

Against this background, we propose a *budgeted ϵ-first algorithm* for our MAB, in which the first ϵ of the overall budget B is dedicated to exploration, and the remaining portion is dedicated to exploitation. In more detail, we split the budget into two parts, ϵB reserved exclusively for exploration, and $(1 - \epsilon)B$ for exploitation. In the exploration phase, the agent estimates the expected rewards by *uniformly* sampling the arms (i.e. the arms are sequentially pulled one after the other). Then, in the exploitation phase, it calculates the optimal combination of pulls based on the estimates generated during the exploration phase. The key benefit of this approach is that we can easily measure the accuracy of the estimates associated with a particular value of ϵ, because all of the arms are sampled the same number of times. Hence, we can control the expected *loss* (i.e. the difference between the optimal and the proposed combination of pulls) as a function of ϵ, which gives us a method of choosing an optimal ϵ for a given scenario. Furthermore, it also gives us a bound on the performance of the ϵ-first approach to the MAB problem with an overall budget.

Given this, the main contributions of this paper are:

- We introduce a new version of MAB, in which each arm has different pulling cost, and both exploration and exploitation phases are limited by a single overall budget.
- We devise the first, theoretically proven upper bound for the loss of a budgeted ϵ–first algorithm that uses any generic exploration method for this problem.
- We improve this upper bound for the case of budgeted ϵ–first approach with uniform pull exploration, in which all of the arms are uniformly pulled in the exploration phase.

The paper is organised as follows: Next we describe the multi-armed bandit with an overall budget limit. We then introduce our ϵ–first algorithm in Section 3. In Section 4 we derive an upper bound on the loss of the algorithm when any generic method is used in the exploration phase, and a refined bound for uniform pull exploration. Then Section 5 presents an empirical comparison of our ϵ–first algorithm using the uniform pull and the UCB–based sampling methods with an ϵ_n–greedy algorithm. Section 6 concludes.

2 Model Description

In this section, we describe the MAB with an overall budget limit. In more detail, we consider a k–armed bandit problem, in which only one arm can be pulled at any time by the agent. Upon pulling arm i of the machine, the agent pays a pulling cost, denoted c_i, and receives a non–negative reward drawn from a distribution associated with that specific arm. Note that in our model, the only restriction on the distribution of these rewards is that they have bounded supports. This assumption is reasonable, since in real–world applications, reward values are typically bounded. Given this, the agent's goal is to maximise the sum of rewards it earns from pulling the arms of the machine. However, initially, the agent has no knowledge about the mean value μ_i of each arm i, so must learn these values in order to deduce a policy that maximises its sum of rewards. Furthermore, in our model, the agent has a cost budget B, which it cannot exceed during its operation time. That is, the total cost of pulling arms cannot exceed this budget limit. Given this, the agent's objective is to find the optimal sequence of pulls, that maximises the expectation of the total reward the agent can achieve, without exceeding the cost budget B.

Formally, let $A = \{i(1), i(2), \dots\}$ be a finite sequence of pulls, where $i(t)$ denotes the arm pulled at time t. The set of possible sequences is limited by the budget. That is, the total cost of the sequence A cannot exceed budget B:

$$\sum_{i(t)\in A} c_{i(t)} \leq B$$

Let $R(A)$ denote the total reward the agent receives by pulling the sequence A. The expectation of $R(A)$ is:

$$\mathbb{E}[R(A)] = \sum_{i(t)\in A} \mu_{i(t)}$$

Then, let A^* denote the optimal sequence that maximises the expected total reward, which can be formulated as follows:

$$A^* = \arg\max_A \mathbb{E}\left[R(A)\right] = \arg\max_A \sum_{i(t)\in A} \mu_{i(t)} \qquad (1)$$

Note that in order to determine A^*, we have to know the value of μ_i in advance, which does not hold in our case. Thus, A^* represents a theoretical optimum value, which is unachievable in general.

However, for any sequence A, we can define the loss (or regret) for A as the difference between the expected cumulative reward for A and the theoretical optimum A^*. More precisely, letting $L(A)$ denote the loss function, we have:

$$L(A) = \mathbb{E}\left[R(A^*)\right] - \mathbb{E}\left[R(A)\right] \qquad (2)$$

Given this, our objective is to derive a method of generating a sequence, A, that minimises this loss function for the class of MAB problems defined above.

3 Budgeted ϵ–first Approach

In this section we introduce our ϵ–first approach for the MAB with budget limits problem. The structure of the ϵ–first approach is such that the exploration and exploitation phases are independent of each other, so in what follows, the methods used in each phase are discussed separately. Now, several combinations of methods can be used for both exploration and exploitation. However, due to space constraints, in this work we investigate only one exploration and one exploitation method. For the former, we use a *uniform pull sampling method*, in which all of the arms are uniformly pulled until the exploration budget is exceeded, because we can put a higher bound for the loss of an ϵ–first algorithm using this exploration method than for other methods. In the exploitation phase, due to the similarities of our MAB to unbounded knapsack problems when the rewards are known, we use the *reward–cost ratio ordered greedy* method, a variant of an efficient greedy approximation algorithm for knapsack problems.

3.1 Uniform Pull Exploration

In this phase, we uniformly pull the arms, with respect to the exploration budget ϵB. That is, we sequentially pull all of the arms, one after the other, until the budget is exceeded. Letting n_i denote the number of pulls of arm i, we have:

$$\left\lfloor \frac{\epsilon B}{\sum_{j=1}^{k} c_j} \right\rfloor \leq n_i \leq \left\lfloor \frac{\epsilon B}{\sum_{j=1}^{k} c_j} \right\rfloor + 1 \qquad (3)$$

The reason of choosing this method is that, in order to bound the loss of the algorithm, since we do not know which arms will be pulled in the exploitation phase, we need to treat the arms equally in the exploration phase. Hereafter we refer to the sequence sampled by the uniform algorithm as A_{uni}.

3.2 Reward–Cost Ratio Ordered Exploitation

We first introduce the foundation of the method used in this phase of our algorithm, the unbounded knapsack problem. We then describe an efficient approximation method for solving this knapsack problem, which we subsequently use in the second phase of our budgeted ϵ-first algorithm.

The unbounded knapsack problem is formulated as follows. Given k types of items, each type i has a corresponding value v_i, and weight w_i. In addition, there is also a knapsack with weight capacity B. The unbounded knapsack problem selects integer units of those types that maximise the total value of items in the knapsack, such that the total weight of the items does not exceed the knapsack weight capacity. That is, the goal is to find the non–negative integers $x_1, x_2, \ldots, x_k$ that

$$\max \sum_{i=1}^{k} x_i v_i \quad \text{s.t.} \quad \sum_{i=1}^{k} x_i w_i \leq C$$

This problem is *NP*–hard, however, near–optimal approximation methods have been proposed to solve this problem.[1]

In particular, here we make use of a simple, but efficient approximation method, the *density ordered greedy* algorithm, which has $O(k \log k)$ computational complexity, where k is the number of item types (Kohli, Krishnamurti, and Mirchandani 2004). This algorithm works as follows: Let $^{v_i}/_{w_i}$ denote the *density* of type i. At the beginning, we sort the item types in order of the value of their density. This needs $O(k \log k)$ computational complexity. Then in the first round of this algorithm, we identify the item type with the highest density and select as many units of this item as are feasible, without exceeding the knapsack capacity. Then, in the second round, we identify the densest item among the remaining feasible items (i.e. items that still fit into the residual capacity of the knapsack), and again select as many units as are feasible. We repeat this step in each subsequent round, until there is no feasible item left. Clearly, the maximal number of rounds is k.

Now, we reduce the problem faced by an agent in the exploitation phase to the unbounded knapsack problem. Recall that in the exploitation phase, the agent makes use of the expected reward estimates from the exploration phase. Let $\hat{\mu}_i$ denote the estimate of μ_i after the first phase. Given this we aim to solve the following integer program:

$$\max \sum_{i=1}^{k} x_i \hat{\mu}_i \quad \text{s.t.} \quad \sum_{i=1}^{k} x_i c_i \leq (1 - \epsilon) B$$

where x_i is the number of pulls of arm i in the exploitation phase. In this case, the ratio of an arm's reward estimate to its pulling cost, $^{\hat{\mu}_i}/_{c_i}$, is analogous to the "density" of an item, because it represents the reward for consuming one unit of the budget, or one unit of the carrying capacity of the knapsack. As such, the problem is equivalent to the knapsack problem above, and in order to solve it, we can use a *reward–cost ratio* (density) ordered greedy algorithm. Hereafter we refer to the sequence pulled by this algorithm as A_{greedy}.

Now, let x_i^{greedy} denote the solution for arm i given by the value density ordered algorithm. The expected total reward is:

$$\mathbb{E}\left[R\left(A_{\epsilon-\text{first}}\right)\right] = \sum_{i=1}^{k} \left(n_i + x_i^{\text{greedy}}\right) \mu_i$$

where $A_{\epsilon-\text{first}} = A_{\text{uni}} + A_{\text{greedy}}$ denotes the sequence of pulls resulted by our proposed algorithm.

4 An Upper Bound for the Loss Function

In this section, we first derive an upper bound for *any* exploration policy and the reward–cost ratio ordered greedy

[1] A detailed survey of these algorithms can be found in Andonov, Poirriez, and Rajopadhye (2000).

exploitation algorithm (i.e. the upper bound is independent of the choice of the exploration algorithm). We then refine this bound for the specific case of uniform pull exploration, in order to have a more practical and efficient bound.

Recall that both A_{uni} and A_{greedy} together form sequence $A_{\epsilon-\text{first}}$, which is the policy generated by our ϵ-first algorithm. The expected reward for this policy is then:

$$\mathbb{E}\left[R\left(A_{\epsilon-\text{first}}\right)\right] = \mathbb{E}\left[R\left(A_{\text{uni}}\right)\right] + \mathbb{E}\left[R\left(A_{\text{greedy}}\right)\right], \tag{4}$$

which is the expected reward of the exploration phase plus the expected reward of the exploitation phase.

In what follows, we derive lower bounds for $\mathbb{E}\left[R\left(A_{\text{uni}}\right)\right]$ and $\mathbb{E}\left[R\left(A_{\text{greedy}}\right)\right]$ independently. Then, putting these together, we have a lower bound for the expected reward of $A_{\epsilon-\text{first}}$. Following this, we derive an upper bound for the expected reward of the optimal sequence A^*. The difference between the lower bound of $\mathbb{E}\left[R\left(A_{\epsilon-\text{first}}\right)\right]$ and the upper bound of $\mathbb{E}\left[R\left(A^*\right)\right]$ then gives us the upper bound of the loss function of our proposed algorithm. However, this bound is constructed using Hoeffding's inequality, so it is correct only with a certain probability. Specifically, it is correct with probability $(1-\beta)^k$, where $\beta \in (0,1)$ is a predefined confidence parameter (i.e. the confidence with which we want the upper bound to hold) and k is the number of arms.

Given this, without loss of generality, for ease of exposition we assume that the reward distribution of each arm has support in $[0,1]$, and the pulling cost $c_i > 1$ for each i (our result can be scaled for different size supports and costs as appropriate). We now define some terms. Let $I^{\max} = \arg\max_i \frac{\hat{\mu}_i}{c_i}$ denote the arm with the highest mean value density estimate. Similarly, let $i^{\max}$ and $i^{\min}$ denote the arm with the highest and the lowest real mean value densities, respectively. Then, let $D_{\max}$ denote the difference between the highest and the lowest mean value densities, i.e. $D_{\max} = \frac{\mu_{i^{\max}}}{c_{i^{\max}}} - \frac{\mu_{i^{\min}}}{c_{i^{\min}}}$. Finally, let A_{arb} be an arbitrary exploration sequence of pulls. We say that A_{arb} exploits the budget dedicated to the exploration if and only if after A_{arb} stops, none of the arms can be additionally pulled without exceeding the exploration budget. Given this, we now state the main contribution of the paper.

Theorem 1 *Suppose that A_{arb} is an arbitrary exploration sequence, that exploits the given exploration budget, and that within this sequence, each arm i is pulled n_i times. Let A denote the joint sequence of A_{arb} and A_{greedy}, where A_{greedy} is as defined in Section 3.2. Then, for any $k > 1$, $B > 0$, and $0 < \epsilon, \beta < 1$, with probability $(1-\beta)^k$, the loss function of sequence A is at most:*

$$2 + \epsilon B D_{\max} + B\left(\sqrt{\frac{-\ln\beta}{n_{i^{\max}}}} + \sqrt{\frac{-\ln\beta}{n_{I^{\max}}}}\right)$$

where $n_{i^{\max}}$ and $n_{I^{\max}}$ are the number of pulls of arms $i^{\max}$ and $I^{\max}$, respectively.

A sketch of the proof is provided in the Appendix. Note that this theorem is of no practical use, since it assumes knowledge of $i^{\max}$ and $i^{\min}$, which is unlikely in real applications. However, it gives us a generic upper bound for the loss function that can be refined for specific exploration methods.

In particular, we now refine this upper bound for the case of uniform pull exploration A_{uni}. Since we uniformly pull the arms, A_{uni} exploits the exploration budget. Furthermore, according to Equation 3, each arm is pulled at least $\lfloor {}^{\epsilon B}/{\sum_{j=1}^{k} c_j} \rfloor$ times. Thus, we can state the following:

Corollary 2 *Let $A_{\epsilon-\text{first}}$ denote the pulling sequence of the ϵ-first algorithm with uniform pull exploration and density value–ordered greedy exploitation. For any $k > 1$, $B > 0$, and $0 < \epsilon, \beta < 1$, with probability $(1-\beta)^k$, loss function of $A_{\epsilon-\text{first}}$ is at most*

$$L\left(A_{\epsilon-\text{first}}\right) \leq 2 + \epsilon B D_{\max} + 2B\left(\sqrt{\frac{(-\ln\beta)}{\lfloor {}^{\epsilon B}/{\sum_{j=1}^{k} c_j} \rfloor}}\right)$$

Apart from the uniform pull algorithm, other exploration policies, such as UCB, or Boltzmann exploration (Cicirello and Smith 2005), can be applied within our ϵ-first approach. The upper bound given in Theorem 1 also holds for those policies. However, refining this upper bound, as we did in the case of the uniform pull algorithm, is a challenge. In particular, estimating the number of times each arm is pulled is difficult in both UCB and Boltzmann explorations. Thus, refined upper bounds cannot be derived for these policies using the same technique as for the uniform pull policy. As such, we must rely on empirical comparisons of ϵ–first with uniform and non–uniform exploration policies.

5 Performance Evaluation

In this section, we evaluate our ϵ–first algorithm with uniform exploration and with UCB–based exploration, and compare them to a modified version of the ϵ_n*–greedy* algorithm. The latter was originally proposed to solve the standard MAB problem, while UCB–exploration was developed to solve the MAB with pure exploration. We compare these algorithms in three scenarios, which highlight the benefits of using an algorithm that is tailored to the budget–limited MAB. However, before discussing the experiments, we first describe the benchmark algorithms and explain why they were chosen.

The ϵ_n–greedy algorithm is designed for standard MABs. It learns the values of all arms while still selecting the arm with the highest reward infinitely more frequently than any other, and can be described as follows: At each time–step n, the arm with the *highest expected reward estimate* is pulled with probability $1-\epsilon_n$, otherwise a randomly selected arm is pulled. The value of ϵ_n is decreased over time, proportional to $O\left({}^1/{n}\right)$, starting from an initial value ϵ_0. However, since ϵ_n–greedy does not take pulling cost into account, we modify it to fit our model's cost constraints. Specifically, the arm with the highest reward value–cost ratio is pulled with probability $1-\epsilon_n$, with any other randomly chosen arm pulled with probability ϵ. Now, consider the case of a budget–limited MAB when the budget B is considerably larger than the pulling cost of each arm (e.g. $B \to \infty$). In this situation, the difference between the pulling cost of the arms are not significant, and thus, this version of our problem could be approximated by a standard MAB (without pulling cost). However, when B is comparable to the arms' pulling costs, it is not obvious how traditional MAB algorithms, such as ϵ_n-greedy, behave.

Conversely, since our proposed ϵ-first approach relies significantly on the efficiency of the exploration phase, it is not obvious that uniformly pulling the arms is the best choice for

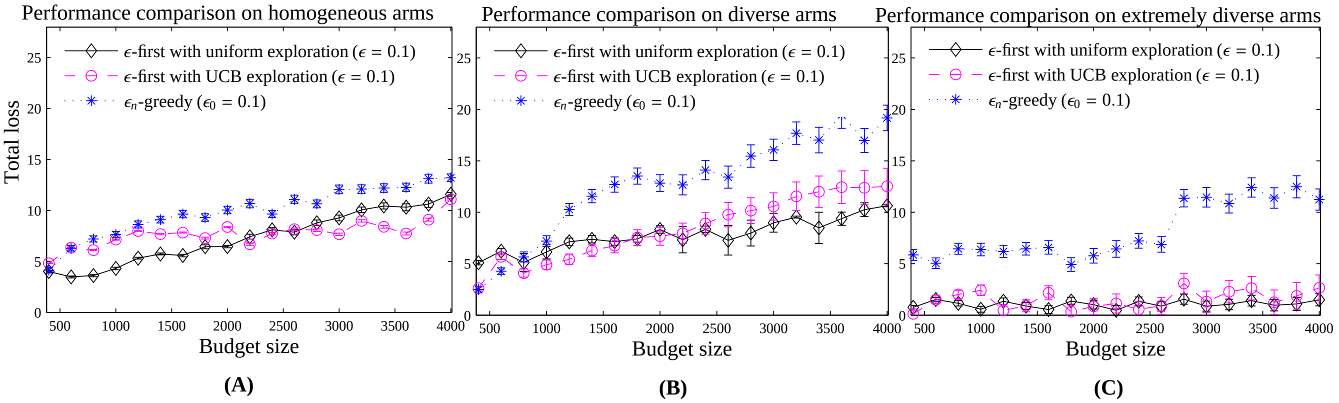

Figure 1: *Total loss of the algorithms in the case of 6-armed bandit machine, which has: (A) homogeneous arms with expected reward value–cost parameter pairs* $\{4, 4\}, \{4, 4\}, \{4, 4\}, \{2.7, 3\}, \{2.7, 3\}, \{2.7, 3\}$*; (B) diverse arms with parameter pairs* $\{3, 4\}, \{2, 4\}, \{0.2, 2\}, \{0.16, 2\}, \{18, 20\}, \{18, 16\}$*; and (C) extremely diverse arms with with parameter pairs* $\{0.44, 5\}, \{0.4, 4\}, \{0.2, 3\}, \{0.08, 1\}, \{14, 120\}, \{18, 150\}$.

exploration. Thus, there is a need to compare our proposed algorithm with ϵ-first approaches that use different exploration techniques. Given this, we choose an *ϵ-first approach with UCB–based exploration* as the second benchmark algorithm. The UCB exploration phase pulls the arm with the highest upper confidence bound on the arm's expected value, in order to determine the arm with the highest expected reward. Since UCB was proposed for MAB with non–costly arms, we also have to modify its goal, similarly to the case of ϵ_n-greedy. Note that, according to Bubeck, Munos, and Stoltz (2009), UCB typically results in better performance in exploration than the uniform pull approach.

Now, recall that due to the limited budget, the optimal solution of our MAB problem is not to repeatedly pull the optimal arm, but to pull the optimal combination of arms. However, our proposed exploitation algorithm, the reward–cost ratio ordered greedy method, does not always return an optimal combination, because it is an approximate algorithm. Specifically, if the arms are homogeneous, that is, the pulling costs do not significantly differ from each other, then the reward–cost ratio ordered greedy typically results in solely pulling the arm with highest reward estimate–cost ratio. Consequently, it shows similarities to the ϵ_n–greedy policy in this case. On the other hand, when the arms are more diverse, that is, the pulling costs are significantly different, the algorithm pulls more arms (for more details see Kohli, Krishnamurti, and Mirchandani (2004)).

Given this, in order to measure the efficiency of the reward–cost ratio ordered greedy in different situations, we set three test cases, each of which considers a 6-armed bandit machine which has: (i) homogenous arms; (ii) moderately diverse arms; and (iii) extremely diverse arms. In the moderately and extremely cases, two out of six arms have moderately/extremely more expensive pulling cost than the others. Given this, the expected reward and cost values were randomly chosen, with regard to these diversity properties. Specifically, the distribution of the rewards are Gaussian, with means given as the first element of the expected reward value–cost pairs in the caption of Figure 1, with 0.1 variance value , and supports $[0, 20]$. The cost of each arm is given as the second element of the expected reward value–cost pairs. Finally, for all test cases, we run our simulation with different budget values, from 400 to 4000. The numerical results are depicted in Figure 1. Within this figure, the error bars represent the 95% confidence intervals.

From Figure 1, we can see that in the homogeneous case, both ϵ_n-greedy and the ϵ-first with uniform exploration approach achieve similar efficiency. However, our algorithm results in better performance when the arms are more diverse. In particular, the total loss of our algorithm is 50% less in the diverse case, and 80% less in the extremely diverse case, than that of the ϵ_n-greedy. In addition, despite the better performance of UCB in exploration, the performance of our algorithm surprisingly does not significantly differ from that of the ϵ-first with UCB-based exploration, since both uniform and UCB–based explorations can efficiently determine those arms which are pulled in the reward–cost ratio ordered greedy algorithm. In order to show this, and to highlight the differences of each algorithm's behaviour as well, consider a single run of the case of extremely diverse arms, with budget $B = 3500$. Figure 2 depicts the number of pulls of each particular arm. Here, the optimal solution is to pull arm 2 twelve times, arm 4 two times, and arm 6 twenty–three times. We can see that both ϵ-first with uniform and with UCB-based exploration show similar behaviour to the optimal solution, however, ϵ_n-greedy focuses on arms 2 and 5 instead, while arm 6 is not sampled at all. The reason is, due to its nature, if ϵ_n-greedy starts with a wrong arm (i.e. not the best arm), it will stick with it for a while. Therefore, in order to learn the best combination, it needs more time. However, due to the limited budget, it does not have enough time to do this.

6 Conclusions and Future Work

In this paper, we have introduced a new MAB model, the budget-limited MAB with an overall budget, in which the number of pulls is limited. Thus, different pulling behaviour is needed within this model, in order to achieve maximal expected total reward. Given this, we have proposed an ϵ-first based approach that splits the overall budget into exploration and exploitation budgets. Our algorithm uses a uniform pull policy for exploration, and the reward–cost ratio ordered greedy algorithm for exploitation. Beside this, we have devised a theoretical upper bound for the loss func-

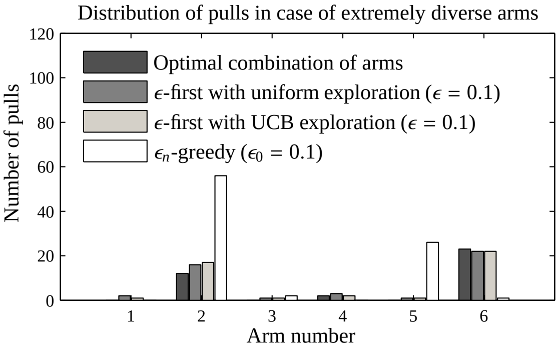

Figure 2: *Number of pulls of each arm in the case of 6–armed bandit with extremely diverse arms, and budget* $B = 3500$.

tion of generic ϵ-first approaches with arbitrary exploration policies. We also have refined this upper bound to the specific case of uniform pull exploration. Finally, through simulation, we have demonstrated that the ϵ-first approaches outperform the ϵ_n-greedy method, an efficient algorithm for standard MAB problems.

However, since it is difficult to estimate the number of pulls of each arm, it is not trivial to refine the proposed generic upper bound to other non–uniform exploration policies. Thus, new techniques are needed to provide bounds for those methods. Therefore, as future work, we intend to extend the analysis to other ϵ-first approaches with different exploration and exploitation algorithms.

References

Andonov, R.; Poirriez, V.; and Rajopadhye, S. 2000. Unbounded knapsack problem: dynamic programming revisited. *European Journal of Operational Research* 123(2):394–407.

Antos, A.; Grover, V.; and Szepesvári, C. 2008. Active learning in multi-armed bandits. *In Proc. of 19th Int. Conf. on Algorithmic Learning Theory* 287–302.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47:235–256.

Bubeck, S.; Munos, R.; and Stoltz, G. 2009. Pure exploration for multi-armed bandit problems. *Algorithmic Learning Theory* 23–37.

Cicirello, V. A., and Smith, S. F. 2005. The max k-armed bandit: a new model of exploration applied to search heuristic selection. *In Proc. of AAAI'05* 1355–1361.

Guha, S., and Munagala, K. 2007. Approximation algorithms for budgeted learning problems. *Proc. of STOC'07* 104–113.

Kohli, R.; Krishnamurti, R.; and Mirchandani, P. 2004. Average performance of greedy heuristics for the integer knapsack problem. *European Journal of Operational Research* 154(1):36–45.

Padhy, P.; Dash, R. K.; Martinez, K.; and Jennings, N. R. 2006. A utility-based sensing and communication model for a glacial sensor network. *In Proc. of AAMAS'06* 1353–1360.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bulletin of the AMS* 55:527–535.

Appendix: Proof of Theorem 1

Before we prove it, let us prove the following auxiliary lemmas.

Lemma 3 *Let A_{arb} denote an arbitrary exploration sequence that exploits its exploration budget. Within this sequence, each arm i is pulled n_i times. Thus, we have* $\sum_{i=1}^{k} n_i\mu_i \geq \epsilon B\left({}^{\mu_{i^{\min}}}/_{c_{i^{\min}}}\right) - 1$

Lemma 4 *If A_{greedy} is the density-ordered greedy exploitation algorithm, then* $\mathbb{E}\left[R\left(A_{\mathrm{greedy}}\right)\right] \geq (1-\epsilon)\,B\left({}^{\mu_{I^{\max}}}/_{c_{I^{\max}}}\right) - 1$

Lemma 5 *If A^* is the optimal sequence defined in equation (1), then* $\mathbb{E}\left[R\left(A^*\right)\right] \leq B\left({}^{\mu_{i^{max}}}/_{c_{i^{max}}}\right)$

Lemma 6 *If $|a-b| \leq \delta_1$, $|c-d| \leq \delta_2$, $a \geq c$, then $d \leq b + \delta_1 + \delta_2$*

Proof of lemma 3. If A_{arb} exploits the budget for exploration, it is true that for any c_j, $\sum_{i=1}^{k} n_i c_i \geq \epsilon B - c_j$, since none of the arms can be pulled after the stop of A_{arb}, without exceeding ϵB. Furthermore, $\mu_i = c_i\left({}^{\mu_i}/_{c_i}\right) \geq c_i\left({}^{\mu_{i^{\min}}}/_{c_{i^{\min}}}\right)$. Since $\mu_i \geq 1$, we have:

$$\sum_{i=1}^{k} n_i\mu_i \geq \left(\sum_{i=1}^{k} n_i c_i\right)\frac{\mu_{i^{\min}}}{c_{i^{\min}}} \geq \left(\epsilon B - c_{i^{\min}}\right)\frac{\mu_{i^{\min}}}{c_{i^{\min}}} \geq \frac{\epsilon B\mu_{i^{\min}}}{c_{i^{\min}}} - 1$$

Proof of lemma 4. By just pulling arm $I^{\max}$ in the exploitation phase, which is the first round of A_{greedy}, the expected reward we can get there is $\lfloor \frac{(1-\epsilon)B}{c_{I^{\max}}} \rfloor \mu_{I^{\max}}$. Since $\lfloor \frac{(1-\epsilon)B}{c_{I^{\max}}} \rfloor > \left(\frac{(1-\epsilon)B}{c_{I^{\max}}} - 1\right)$, we have:

$$\mathbb{E}\left[R\left(A_{\mathrm{greedy}}\right)\right] \quad \geq \quad \left(\frac{(1-\epsilon)\,B}{c_{I^{\max}}} - 1\right)\mu_{I^{\max}} \geq (1-\epsilon)\,\frac{B\mu_{I^{\max}}}{c_{I^{\max}}} - 1$$

Proof of lemma 5. Suppose that in the optimal sequence, N_i is the total number of pulls of arm i. Thus, the cost constraint can be formulated as $\sum_{i=1}^{k} N_i c_i \leq B$. Given this, we have:

$$\mathbb{E}\left[R\left(A^*\right)\right] = \sum_{i=1}^{k} N_i\mu_i = \sum_{i=1}^{k} N_i c_i \frac{\mu_i}{c_i} \leq \left(\sum_{i=1}^{k} N_i c_i\right)\frac{\mu_{i^{\max}}}{c_{i^{\max}}} \leq \frac{B\mu_{i^{\max}}}{c_{i^{\max}}}$$

Proof of lemma 6. Since $|a-b| \leq \delta_1$, we have $a \leq b + \delta_1$. Similarly, we have $d \leq c + \delta_2$. Since $a \geq c$, we have the following: $d \leq c + \delta_2 \leq a + \delta_2 \leq b + \delta_1 + \delta_2$.

Proof sketch of theorem 1. Let n_i denote the number of pulls of arm i in the exploration phase. Using Hoeffding's inequality for each arm i, and for any positive δ_i, we have:

$$P\left(\left|\frac{\hat{\mu}_i}{c_i} - \frac{\mu_i}{c_i}\right| \geq \delta_i\right) \leq \exp\left\{-n_i\delta_i^2 c_i^2\right\} \tag{5}$$

By setting $\delta_i = \sqrt{\frac{-\ln\beta}{n_i c_i^2}}$, it is easy to prove, that with probability $(1-\beta)^k$, $\left|\frac{\hat{\mu}_i}{c_i} - \frac{\mu_i}{c_i}\right| \leq \delta_i$ holds for each arm i. Hereafter, we stricly focus on this case. Given this, the reward collected in the exploration phase can be calculated as follows:

$$\mathbb{E}\left[R\left(A_{\mathrm{arb}}\right)\right] = \sum_{i=1}^{k} n_i\mu_i \geq \frac{\epsilon B\mu_{i^{\min}}}{c_{i^{\min}}} - 1 \tag{6}$$

The right side of equation (6) holds, due to lemma 3. Using lemma 4 and equation (6), we get the following:

$$\begin{aligned}\mathbb{E}\left[R\left(A\right)\right] \quad &= \quad \mathbb{E}\left[R\left(A_{\mathrm{arb}}\right)\right] + \mathbb{E}\left[R\left(A_{\mathrm{greedy}}\right)\right] \\ &\geq \quad \frac{\epsilon B\mu_{i^{\min}}}{c_{i^{\min}}} + (1-\epsilon)\,\frac{B\mu_{I^{\max}}}{c_{I^{\max}}} - 2 \qquad (7)\end{aligned}$$

By denifition, we have ${}^{\mu_{I^{\max}}}/_{c_{I^{\max}}} \geq {}^{\mu_{i^{\max}}}/_{c_{i^{\max}}}$, and ${}^{\hat{\mu}_{i^{\max}}}/_{c_{i^{\max}}} \geq {}^{\hat{\mu}_{I^{\max}}}/_{c_{I^{\max}}}$. Furthermore, $\left|\frac{\hat{\mu}_i}{c_i} - \frac{\mu_i}{c_i}\right| \leq \delta_i$ holds for each arm i. Thus, according to lemma 6, we have ${}^{\mu_{I^{\max}}}/_{c_{I^{\max}}} \geq {}^{\mu_{i^{\max}}}/_{c_{i^{\max}}} - \delta_{i^{\max}} - \delta_{I^{\max}}$. Substituting this into equation (7), we have:

$$\begin{aligned}\mathbb{E}\left[R\left(A\right)\right] \quad \geq \quad &\frac{\epsilon B\mu_{i^{\min}}}{c_{i^{\min}}} + B\frac{\mu_{i^{\max}}}{c_{i^{\max}}} - \frac{\epsilon B\mu_{i^{\max}}}{c_{i^{\max}}} - \\ &- (1-\epsilon)\,B\left(\delta_{i^{\max}} + \delta_{I^{\max}}\right) - 2 \qquad (8)\end{aligned}$$

According to lemma 5, $\mathbb{E}\left[R\left(A^*\right)\right] \leq \frac{B\mu_{i^{\max}}}{c_{i^{\max}}}$. Thus, by substituting it into equation (8), and using the definition of loss function in equation (2), we have:

$$L\left(A\right) \leq 2 + \epsilon BD_{\max} + B\left(\delta_{i^{\max}} + \delta_{I^{\max}}\right)$$

where $D_{\max} = \left({}^{\mu_{i^{\max}}}/_{c_{i^{\max}}}\right) - \left({}^{\mu_{i^{\min}}}/_{c_{i^{\min}}}\right)$. Note that here we used the fact that $(1-\epsilon) < 1$. Thus, by replacing $\delta_i = \sqrt{\frac{-\ln\beta}{n_i c_i^2}}$ for $i = I^{\max}$ and $i = i^{\max}$, and using the fact that $c_i \geq 1$ for each i, we get the requested formula.