ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-560

Budgeted Multi–Armed Bandit in Continuous Action Space

Francesco Trovò, Stefano Paladino, Marcello Restelli, Nicola Gatti[1]

Abstract. Multi–Armed Bandits (MABs) have been widely considered in the last decade to model settings in which an agent wants to learn the action providing the highest expected reward among a fixed set of available actions during the operational life of a system. Classical techniques provide solutions that minimize the regret due to learning in settings where selecting an arm has no cost. Though, in many real world applications the learner has to pay some cost for pulling each arm and the learning process is constrained by a fixed budget B. This problem is addressed in the literature as the Budgeted MAB (BMAB). In this paper, for the first time, we study the problem of Budgeted Continuous–Armed Bandit (BCAB), where the set of the possible actions consists in a continuous set (e.g., a range of prices) and the learner suffers from a random reward and cost at each round. We provide a novel algorithm, named B–Zoom, which suffers a regret of $\tilde{O}(B^{\frac{d+1}{d+2}})$, where d is the Zooming dimension of the problem. Finally, we provide an empirical analysis showing that, despite a lower average performance, the proposed approach is more robust to adverse settings as compared to existing algorithms designed for BMAB.

1 Introduction

In a Multi–Armed Bandit (MAB) problem [3], an agent, called *learner*, is allowed to select a single option, called *arm*, from a finite number of available options and to observe the corresponding stochastic reward. The techniques developed for a MAB problem minimize the loss, called *regret*, incurred during the learning process and provide theoretical guarantees about convergence to the optimal arm. Most regret–minimization algorithms available in the literature provide solutions to the case in which there is a constraint over the maximum number of rounds the agent is allowed to pull arms. However, in many applications, an agent is subject to different constraints. A very common case is when the learner has a fixed budget which she uses to pay a stochastic cost associated with the pulling of a specific arm. Simply, the constraint over the budget reduces to the constraint over the maximum number of rounds when each arm has a fixed unitary cost.

In this setting, known as *Budgeted MAB* (BMAB) [10], the learner is given a fixed budget in advance and she is allowed to pull arms until the budget has been totally spent. The BMAB is able to model a wide range of concrete applications. For instance, bidding in Sponsored Search Auctions (SSA) [6] when an advertiser has no information neither about the probability of being clicked (usually called click–through rate) nor about the cost of being clicked is a BMAB problem. In the same field, the problem of optimizing an advertising campaign presents a similar model. Another application that can be modeled by means of a BMAB problem consists in determining the optimal sensor to interrogate in a wireless sensor network scenario [16, 18]. More precisely, when we retrieve information from a sensor, we gain information about the monitored process and, at the same time, we spend budget in terms of energetic costs. Also the problem of a service provider trying to balance the costs of the employed resources and the revenues gained by the provided services fits the BMAB model [2].

In many applications, the use of a finite set of arms provides an extremely raw model of the situation one studies, potentially forcing the learner to pull only suboptimal arms and thus to suffer a linear regret over time (or budget). Natural examples of spaces of continuous arms are prices and costs. In this paper, to the best of our knowledge, we study the first generalization of the BMAB to continuous arm spaces, named the *Budgeted Continuous–Armed Bandit* (BCAB). In order to cope efficiently with continuous space, we need additional assumptions over the regularity of the average reward and cost functions. In particular, as customary in the literature on Continuous–Armed Bandits (CAB), we assume Lipschitz continuity over the expected reward and cost functions. Such an assumption is largely supported by real–world scenarios, e.g., in SSAs similar bids have similar expected rewards and payments.

Related works A number of recent results on sequential learning settings whose stopping time depends on a fixed budget can be found in the literature. Some of them consider fixed costs [1, 7, 11, 17], while others assume to have stochastic ones [10, 19, 20]. There is a wide literature studying settings in which exploration and exploitation phases are separate and only the exploration phase is subject to costs [1, 7, 11]. Only few works consider settings with deterministic costs without separating exploration and exploitation phases. In particular, in [17], the authors tackle the problem by relying on an approximated optimization technique of the unbounded knapsack problem, which hardly generalizes to the setting with stochastic costs. [5, 10, 19, 20] consider the BMAB problem with stochastic rewards and costs over a discrete space of arms. In [10], the authors propose a frequentist approach that relies on UCB–like bounds [3] achieving an $O(\log B)$ regret for a generic instance of the BMAB having budget B. This approach has been extended to consider also linear bandits in [19].[2] [20] considers the Thompson sampling algorithm to solve the budgeted MAB in the same setting having the same theoretical upper bound by relying on a Bayesian framework. Finally in [5], an algorithm providing a distribution–free bound of $\tilde{O}(\sqrt{B})$ has been

[1] Politecnico di Milano, Italy, email: {francesco1.trovo, stefano.paladino, marcello.restelli, nicola.gatti}@polimi.it.

[2] In linear bandit problems, the reward function is forced to be linear.

proposed.[3]

The literature provides a large number of results analyzing the CAB setting without costs and budget [4, 8, 9, 12, 13, 14, 15]. The CAB problem is arbitrarily hard in the general setting in which the reward function can be arbitrary, presenting $\Theta(T)$ regret over a horizon of T rounds. Positive results can be obtained when the reward function exhibits some structure. Under the assumption of Lipschitzianity of the expected reward functions, the lower bound over the regret is $\Omega(T^{2/3})$ for the one dimensional version of the problem [14]. The former techniques for the CAB problem are based, initially, on the discretization of the action space by exploiting the structure of the problem and, subsequently, on the adoption of MAB techniques over the discretized problem [4, 9, 15] (let us notice that the application of MAB algorithms to any "blind" discretization may lead to a regret $\Omega(T)$). Recently, new techniques adopting a different, more efficient, approach which changes the set of arms during time on the basis of the observed performance of the arms previously chosen have been developed. One of the most promising techniques for this setting is the Zooming algorithm [12, 13]. This algorithm is designed for CAB problems in metric spaces and, differently from most of the previous works, that consider a uniform discretization of the space which is fixed in advance, it starts from a single arm and, if needed, automatically adds arms over time in the domain. Moreover, it provides an upper bound on the regret of $\tilde{O}(T^{\frac{d+1}{d+2}})$, where d is the *Zooming dimension* associated with the reward function (in the single dimension version $d = 1$ and therefore the Zooming algorithm matches the lower bound). Another algorithm designed for the same setting is called HOO [8]. It is based on the idea of using a search tree to find the best arm. Although it assures an upper bound comparable to the Zooming one of $\tilde{O}(T^{\frac{p+1}{p+2}})$, where p is the packing dimension, it may have higher computational complexity.

Original contributions Our original contributions are as follow. We design the first algorithm, named B–Zoom, able to work in the BCAB setting; the B–Zoom algorithm extends the Zooming algorithm to the case with budget. We provide a theoretical regret analysis of the B–Zoom algorithm, showing that it suffers a regret $\tilde{O}(B^{\frac{d+1}{d+2}})$ matching the regret of the Zooming algorithm in the case in which the BCAB setting reduces to the CAB one (i.e., $B = T$ and unitary cost for all the arms). We experimentally evaluate B–Zoom comparing its performance w.r.t. that of a number of frequentist algorithms.

2 Problem formulation

We denote by $\mathcal{A} \subseteq [0, 1]$ the space of the available actions, also called arms, and by x a generic arm. In the BCAB setting, at each round $t \in \mathbb{N}^+$, a learner is allowed to choose an arm $x_t \in \mathcal{A}$. She receives a reward $r_t(x_t)$ and incurs a cost $c_t(x_t)$. Rewards $r_t(x)$ are realizations of i.i.d. random variables $R_t(x) \sim \mathcal{D}_r([0, 1])$, where $\mathcal{D}_r([0, 1])$ is a generic probability density function (pdf) over support $[0, 1]$, and expected value $\mathbb{E}[R_t(x)] = \mu_r(x)$ with $\mu_r : \mathcal{A} \to [0, 1]$. Costs $c_t(x)$ are realizations of i.i.d. random variables $C_t(x) \sim \mathcal{D}_c([0, 1])$, where $\mathcal{D}_c([0, 1])$ is a generic pdf over support $[0, 1]$, and expected value $\mathbb{E}[C_t(x)] = \mu_c(x)$ with $\mu_c : \mathcal{A} \to [\lambda, 1]$. Here $\lambda > 0$ is a known lower bound on the average cost of an action, needed to exclude the case with costless actions.[4]

A fixed budget $B > 0$ is available to the learner at the beginning of the learning process. We denote by $B(t) := B - \sum_{i=1}^{t-1} c_i(x_i)$ the residual budget available at round t due to the costs incurred in having pulled the arms during the previous $t - 1$ rounds. As customary in the previous works on budget, in the case the learner is not able to pay at t for the cost of the chosen arm x_t, she is forced to stop and does not gain any reward due to x_t. Moreover, we assume that the reward function $\mu_r(x)$ and cost function $\mu_c(x)$ are Lipschitz with known constant L_r and L_c, respectively. These assumptions are usual in Lipschitz bandits and here required to solve our problem.

A generic policy $\mathfrak{U}$ for a BCAB problem is an algorithm able to decide the arm x_t to pull at round t, on the basis of the history in terms of previous realizations of the rewards $\{r_1(x_1), \dots, r_{t-1}(x_{t-1})\}$, costs $\{c_1(x_1), \dots, c_{t-1}(x_{t-1})\}$ and pulled arms $\{x_1, \dots, x_{t-1}\}$. We define the *stopping time* t_a of a generic policy $\mathfrak{U}$ which chooses arm x_t at round t the longest t such that $B(t) \geq 0$. Notice that t_a is a random variable depending on the costs $C_t(x)$ and the initial budget B.

In a BCAB problem, a policy should be able to select a sequence of arms that minimizes the amount of budget spent and maximizes the reward collected during the process. The loss of a generic policy $\mathfrak{U}$ in a BCAB problem with budget B is represented by the *pseudo–regret* $\mathcal{R}(B)$:

$$\mathcal{R}(B) = \mathcal{R}^*(B) - \mathbb{E}_{r,c}\left[\sum_{t=1}^{t_a} r_t(x_t)\right], \tag{1}$$

where, $\mathcal{R}^*(B)$ is the optimal expected total reward when the distribution of rewards $\mathcal{D}_r([0, 1])$ and costs $\mathcal{D}_c([0, 1])$ are known, i.e., the one which solves the following stochastic optimization problem:

$$\max_{\mathfrak{U}} \mathbb{E}\left[\sum_{t=1}^{t_a} r_t(x_t)\right], \text{ s.t. } \sum_{t=1}^{t_a} c_t(x_t) \leq B,$$

where the expected value $\mathbb{E}[\cdot]$ is taken w.r.t. the randomness associated to the policy $\mathfrak{U}$, the rewards, and the costs.

3 The proposed method

In what follows, we introduce our algorithm named B–Zoom to tackle the BCAB problem. The B–Zoom algorithm is based on the idea of the Zooming algorithm and is its extension to the case where a fixed budget B is available and the learner incurs a stochastic cost $C_t(x)$ in pulling arm x at round t. After a brief description of its main features, we provide its theoretical analysis, giving an upper bound over the pseudo–regret $\mathcal{R}(B)$.

3.1 The B–Zoom algorithm

Initially, we introduce the following function on which our algorithm is based:

Definition 1. *Given an average reward function $\mu_r(x)$ and an average cost function $\mu_c(x)$, we define the* expected reward–to–cost ratio *function $\mu : \mathcal{A} \to \left[0, \frac{1}{\lambda}\right]$ as:*

$$\mu(x) = \frac{\mu_r(x)}{\mu_c(x)}.$$

[3] We write $u_n = \tilde{O}(v_n)$ when $u_n = O(v_n)$ up to a logarithmic factor.

[4] Without loss of generality, we considered from now on the setting in which the arms are selected in $\mathcal{A} \equiv [0, 1]$ and average reward $\mu_r(x)$ and cost functions $\mu_c(x)$ have images in $[0, 1]$ for each $x \in \mathcal{A}$. In the case that the space and the average functions are over different domains, a rescaling procedure should be performed so they have values and images in $[0, 1]$.

Algorithm 1 The B–Zoom Algorithm

Input: Budget B**, Minimum average cost** λ**, Arm support set** $\mathcal{A}$
$i_{ph} = 0$
$B(0) \leftarrow B$
$t \leftarrow 0$
while $B(t) > 0$ **do**
  $i_{ph} \leftarrow i_{ph} + 1$
  $X(i_{ph}) \leftarrow \emptyset$
  for $t \in \{2^{i_{ph}-1}, \dots, 2^{i_{ph}} - 1\}$ **do**
    if $B(t-1) > 0$ **then**
      $\mathcal{C} \leftarrow \cup_{x \in X(i_{ph})} \mathcal{B}(E_{t-1}(x), x)$
      $\mathcal{NC} \leftarrow \mathcal{A} \setminus \mathcal{C}$
      if $\mathcal{NC} \not\equiv \emptyset$ **then**
        Randomly pick $x \in \mathcal{NC}$
        $X(i_{ph}) \leftarrow X(i_{ph}) \cup \{x\}$
        $\bar{r}_t(x) \leftarrow 0$
        $\bar{c}_t(x) \leftarrow 0$
        $n_t(x) \leftarrow 0$
        $u_t(x) \leftarrow +\infty$
        $E_t(x_t) \leftarrow +\infty$
      Play arm x_t **s.t.:** $x_t = \arg\max_{x \in X(i_{ph})} u_t(x)$
      Suffer cost $c_t(x_t)$
      $B(t) \leftarrow B(t-1) - c_t(x_t)$
      if $B(t) \geq 0$ **then**
        Gain reward $r_t(x_t)$
        $n_t(x_t) \leftarrow n_{t-1}(x_t) + 1$
        $\bar{r}_t(x_t) \leftarrow \frac{(n_t(x_t)-1)\bar{r}_{t-1}(x_t)}{n_t(x_t)} + \frac{r_t(x_t)}{n_t(x_t)}$
        $\bar{c}_t(x_t) \leftarrow \frac{(n_t(x_t)-1)\bar{c}_{t-1}(x_t)}{n_t(x_t)} + \frac{c_t(x_t)}{n_t(x_t)}$
        $E_t(x_t) \leftarrow \frac{1}{\lambda}\left(1 + \frac{1}{\lambda}\right)\sqrt{\frac{8 i_{ph} + \ln(4)}{n_t(x_t)}}$
        $u_t(x) \leftarrow \frac{\bar{r}_t(x)}{\max\{\lambda, \bar{c}_t(x)\}} + 2E_t(x)$

We can show that function $\mu(x)$ is Lipschitz when both $\mu_r(x)$ and $\mu_c(x)$ are Lipschitz.

Lemma 1. *Given an average reward function $\mu_r : \mathcal{A} \to [0,1]$, L_r–Lipschitz, and an average cost function $\mu_c : \mathcal{A} \to [\lambda, 1]$, $\lambda > 0$, L_c–Lipschitz, the average reward–to–cost ratio function $\mu(x)$ is L'–Lipschitz with $L' \leq \frac{L_c + L_r}{\lambda^2}$.*

Proof. Thanks to the Lipschitz assumption over functions $\mu_r(x)$ and $\mu_c(x)$, we have:

$$|\mu_r(x_1) - \mu_r(x_2)| \leq L_r|x_1 - x_2| \qquad \forall x_1, x_2 \in [0,1]$$
$$|\mu_c(x_1) - \mu_c(x_2)| \leq L_c|x_1 - x_2| \qquad \forall x_1, x_2 \in [0,1]$$

thus:

$$\begin{aligned}
&|\mu(x_1) - \mu(x_2)| = \\
&= \left|\frac{\mu_r(x_1)}{\mu_c(x_1)} - \frac{\mu_r(x_2)}{\mu_c(x_2)}\right| = \frac{|\mu_r(x_1)\mu_c(x_2) - \mu_r(x_2)\mu_c(x_1)|}{\mu_c(x_1)\mu_c(x_2)} \\
&\leq \frac{1}{\lambda^2}|\mu_r(x_1)\mu_c(x_2) - \mu_r(x_1)\mu_c(x_1) + \\
&\qquad + \mu_r(x_1)\mu_c(x_1) - \mu_r(x_2)\mu_c(x_1)| \\
&\leq \frac{1}{\lambda^2}\left(\mu_r(x_1)|\mu_c(x_2) - \mu_c(x_1)| + |\mu_r(x_1) - \mu_r(x_2)|\mu_c(x_1)\right) \\
&\leq \frac{L_c + L_r}{\lambda^2}|x_1 - x_2| \leq L'|x_1 - x_2|.
\end{aligned}$$

□

From now on, without loss of generality, we assume $L' = 1$ Lipschitz constant for the function $\mu(\cdot)$. Indeed, a scaling procedure can be always performed to obtain $L' = 1$.

The B–Zoom algorithm pseudo–code is presented in Algorithm 1. The functioning of the algorithm is split into temporal phases, where the i–th phase is denoted by i_{ph} and the length of phase i_{ph} is $2^{i_{ph}-1}$ rounds. Each phase i_{ph} is associated with a (potentially different) subset of arms $X(i_{ph}) \subset \mathcal{A}$ named *active arms*, which is initially empty and is incrementally populated over the phase i_{ph}. Furthermore, each active arm $x \in X(i_{ph})$ is associated with an open ball $\mathcal{B}(E_t(x), x)$ with radius $E_t(x)$ and centered in x, where $E_t(x)$ is a confidence radius defined as follows:

$$E_t(x) = \begin{cases} \frac{1}{\lambda}\left(1 + \frac{1}{\lambda}\right)\sqrt{\frac{8 i_{ph} + \ln(4)}{n_t(x)}} & \text{if } n_t(x) > 0 \\ +\infty & \text{otherwise} \end{cases},$$

where $n_t(x) = \sum_{i=1}^{t} I\{x_i = x\}$ is the number of rounds an arm x has been pulled up to round t and $I\{\cdot\}$ is the indicator function. The confidence radius $E_t(x)$ varies over time, reducing as the number of rounds an arm x has been pulled increases. Notice that, if an active arm x has never been pulled, its ball $\mathcal{B}(E_t(x), x)$ contains entirely $\mathcal{A}$, the radius $E_t(x)$ being infinite independently of t.

At time t, we define the *covering set* of the active arms $\mathcal{C} = \cup_{x \in X(i_{ph})} \mathcal{B}(E_{t-1}(x), x)$ as the union of the balls of all the active arms. We say that a set $\mathcal{A}$ is *covered* by $\mathcal{C}$ if and only if $\mathcal{C} \supseteq \mathcal{A}$. The covering of a set $\mathcal{A}$ by $\mathcal{C}$ can be easily checked by means of a *covering oracle* (as we discuss below for the sake of presentation); we denote by $\mathcal{NC} := \mathcal{A} \setminus \mathcal{C}$ the subset of $\mathcal{A}$ that is not covered by $\mathcal{C}$.

At each round t, the first task accomplished by the B–Zoom algorithm is to decide whether or not to add new active arms. The *rationale* whereby such a decision is taken follows. If at round t the arm space $\mathcal{A}$ is not covered by the covering set $\mathcal{C}$ of active arms $X(i_{ph})$, the algorithm randomly draws an arm $x \in \mathcal{NC}$ with an arbitrary probability distribution and add it to the active arm set $X(i_{ph})$. Notice that, independently of the shape of $\mathcal{NC}$, no more than one active arm is added at each round t. Indeed, once an active arm has been added, the radius of its ball is, by definition, infinite and therefore the new covering set $\mathcal{C}$ covers the whole arm space $\mathcal{A}$.

At each round t, once the coverage of $\mathcal{A}$ by $\mathcal{C}$ has been evaluated and, potentially, a new active arm has been introduced in $X(i_{ph})$, the B–Zoom algorithm plays the arm $x_t \in X(i_{ph})$ having the maximum upper bound $u_t(x_t)$ defined as:

$$u_t(x) = \frac{\bar{r}_t(x)}{\max\{\lambda, \bar{c}_t(x)\}} + 2E_t(x), \tag{2}$$

where $\bar{r}_t(x) = \frac{\sum_{i=1}^{t} r_i(x) I\{x_i = x\}}{n_t(x)}$ and $\bar{c}_t(x) = \frac{\sum_{i=1}^{t} c_i(x) I\{x_i = x\}}{n_t(x)}$ are the estimated average reward and cost for arm x, respectively. The idea behind the computation of $u_t(x)$ is that we want to upper bound (in high probability) the average reward–to–cost ratio function $\mu(x)$ with the first term in the r.h.s. of Equation 2 plus a radius $E_t(x)$ and we consider another radius $E_t(x)$ to be able to bound $\mu(\tilde{x})$ for all arms $\tilde{x} \in \mathcal{B}(E_t(x), x)$ by relying on the Lipschitzianity of the function $\mu(x)$. Once the arm x_t has been played, the B–Zoom algorithm pays a cost $c_t(x_t)$ and, in the case there is still enough budget remaining $B(t) > 0$, it gains reward $r_t(x_t)$ and updates the necessary statistics $\bar{r}_t(x_t)$, $\bar{c}_t(x_t)$ and $n_t(x_t)$ corresponding to arm x_t, otherwise the algorithm stops. Notice that, in $u_t(x)$, we use $\max\{\lambda, \bar{c}_t(x)\}$ in place of the unbiased estimator $\bar{c}_t(x)$ since for some realizations it could happen that $\bar{c}_t(x) < \lambda$, but we *a priori* know that $\mu_c(x) \geq \lambda$.

The B–Zoom algorithm does not require the setting of any parameter, but it requires information about the Lipschitz constant L' (or equivalently of the constants L_r and L_c related to the average rewards and costs, respectively) and of the minimum average cost λ. Moreover, it requires also a covering oracle. In the case the arm space $\mathcal{A}$ is one dimensional, we can state the following (the complexity in higher dimensional spaces might be higher [8]).

Theorem 1. *A covering oracle for the B–Zoom algorithm over $\mathcal{A} \subset \mathbb{R}$ has computational complexity $O(n)$, where $n = |X(i_{ph})|$ and $|\cdot|$ is the cardinality operator.*

Proof. Let us suppose that at a given round t we are storing in a list s for each arm $x \in X(i_{ph})$ an interval $[p_t(x), q_t(x)] = [x - E_{t-1}(x), x + E_{t-1}(x)]$ and we ordered them w.r.t. ascending values of $p_t(x)$. At each new round, we have either to insert a new arm or modify the confidence radius of an existing one. In the case we introduce a new arm x_j in the set of active arms $X(i_{ph})$, we need to insert the interval $[p_t(x_j), q_t(x_j)]$ in the ordered list s. This operation requires a computational cost of $O(\log_2(n))$ (binary search). Otherwise, if an existing arm x is selected, we have to delete the old interval $[p_{t-1}(x), q_{t-1}(x)]$ from the list s and insert the new one $[p_t(x), q_t(x)]$, which has a total computational cost of $O(\log_2(n))$.

After that, we need to form the covering set $\mathcal{C}$. Let us assume that the list of intervals is $s = \{I_1, \dots, I_n\} = \{[p_t(x_1), q_t(x_1)], \dots, [p_t(x_n), q_t(x_n)]\}$ (with $p_t(x_1) \leq \dots \leq p_t(x_n)$). At first, we have $\mathcal{C}_1 = I_1$. For each $i \in \{2, \dots, n\}$ we perform $\mathcal{C}_i = \mathcal{C}_{i-1} \cup I_i$. If $\mathcal{C}_i$ is still an interval, i.e., if we have $c_M \geq p_t(x_i)$ with $\mathcal{C}_{i-1} = [c_m, c_M]$, we continue the procedure, otherwise we can say that the set $\mathcal{A}$ is not covered by $\mathcal{C}$. If we reached the n–th interval and $\mathcal{C}_n = \mathcal{C} \supseteq \mathcal{A}$, then $\mathcal{A}$ is covered by $\mathcal{C}$. This procedure consists in a maximum of n interval union operations, whose cost is constant. Thus, the computation of the covering set has requires a computational cost of $O(n)$. By considering an empty list s at the first round and by using an inductive argument, we complete the proof. □

At each round t, the B–Zoom algorithm has a computational complexity of $O(n)$ with $n \leq t$ due to the complexity of the covering oracle. Moreover, notice that n is upper bounded by $O\left(\frac{\lambda^4}{(\lambda+1)^2}\frac{t}{\ln(t)}\right)$ and therefore in practice $n \ll t$. For comparison, the HOO algorithm [8], in its general formulation requires $O(t)$ at turn t.

3.2 Theoretical analysis

Considering the problem formulation described in Section 2, we can show that:

Theorem 2. *The regret $\mathcal{R}(B)$ over a generic BCAB problem of the B–Zoom algorithm is:*

$$\mathcal{R}(B) \leq \tilde{C} \cdot (\ln(B))^{\frac{1}{d+2}} \cdot B^{\frac{d+1}{d+2}},$$

where d is the Zooming dimension of the Lipschitz MAB problem $(\mathcal{A}, l, \mu)$, with $l(x, y) = L'|x-y|$, and $\tilde{C}$ is an appropriately defined constant.

Proof. At first, by defining the arm with largest expected reward–to–cost ratio $x^* \in \mathcal{A}$ as:

$$x^* := \arg\max_{x \in \mathcal{A}} \mu(x) = \arg\max_{x \in \mathcal{A}} \frac{\mu_r(x)}{\mu_c(x)},$$

we are able to decompose regret $\mathcal{R}(B)$ defined in Equation (1) into two parts:

$$\mathcal{R}(B) = \underbrace{\mathcal{R}^*(B) - \mathbb{E}_{r,c}\left[\sum_{t=1}^{t_a^*} r_t(x^*)\right]}_{\mathcal{R}_1} + \underbrace{\mathbb{E}_{r,c}\left[\sum_{t=1}^{t_a^*} r_t(x^*)\right] - \mathbb{E}_{r,c}\left[\sum_{t=1}^{t_a} r_t(x_t)\right]}_{\mathcal{R}_2},$$

where $\mathcal{R}_1$ is the component considering that the best possible strategy is not the one choosing always x^* until t_a^* (the stopping round of action x^*), and $\mathcal{R}_2$ is the component considering the loss due to the process of finding the arm x^*.

Regret $\mathcal{R}_1$ can be bounded by trivially extending the result discussed in [20] for BMAB to the case of BCAB:

Lemma 2. *Given any instance of the BCAB problem we have:*

$$\mathcal{R}_1 \leq 2\mu(x^*) = 2\frac{\mu_r(x^*)}{\mu_c(x^*)} \leq \frac{2}{\lambda}.$$

Instead, bounding $\mathcal{R}_2$ is not trivial. For sake of clarity, we divide the proof into three steps. In the first step, we define an auxiliary Lipschitz CAB problem $(\mathcal{A}, l, \mu)$, i.e., a CAB problem without budget or, equivalently, in which each arm has unitary cost and the budget corresponds to a temporal deadline. We show that the execution of the B–Zoom algorithm up to $t = t_a$ to problem $(\mathcal{A}, l, \mu)$ is equivalent to the execution of a modified version of the Zooming algorithm. We use this relation to bound the regret of this problem with $\mathcal{R}_\Delta(t)$ over a generic horizon $t \leq t_a$. In the second step, we show that $\mathcal{R}_2$ is bounded by $\mathcal{R}_\Delta(t_a)$. In the third step, we derive the relationship between the stopping round t_a and the budget B of a BCAB problem and use it to formulate the bound over $\mathcal{R}_2$ in terms of the budget B.

Step 1. Since Lemma 1 holds, the instance of the CAB problem $(\mathcal{A}, l, \mu)$, with $l(x, y) = L'|x - y|$, is a Lipschitz MAB problem [13]. The regret of the B–Zoom algorithm executed over the Lipschitz problem $(\mathcal{A}, l, \mu)$ at round $t \leq t_a$ is defined as:

$$\begin{aligned}\mathcal{R}_\Delta(t) &:= \sum_{i_{ph}=1}^{\log_2(t)} \sum_{x \in X(i_{ph})} \left(\frac{\mu_r(x^*)}{\mu_c(x^*)} - \frac{\mu_r(x)}{\mu_c(x)}\right) n_t(x) \\ &= \sum_{i_{ph}=1}^{\log_2(t)} \sum_{x \in X(i_{ph})} (\mu(x^*) - \mu(x))\, n_t(x).\end{aligned}$$

By verifying that the bounds used in the B–Zoom algorithm satisfy the properties required in Lemma 4.15 in [13] we are able to resort on the regret bound results presented in the same work. More specifically we require that the following two properties are satisfied by the B–Zoom algorithm:

Property 1. *Consider an instance of the Lipschitz CAB problem $(\mathcal{A}, l, \mu)$ and a generic algorithm $\mathfrak{U}$ considering estimates $\hat{\mu}(x)$ and confidence radius $b_t(x)$ for the arm x in phase i_{ph}. A phase i_{ph} is* clean *with probability δ if for each t s.t. $2^{i_{ph}} \leq t \leq 2^{i_{ph}+1} - 1$ and for each arm $x \in \mathcal{A}$:*

$$|\hat{\mu}(x) - \mu(x)| < r_t(x)$$

holds with probability at least $1 - \delta$.

Property 2. *Consider the instance of the Lipschitz CAB problem $(\mathcal{A}, l, \mu)$ and a generic algorithm $\mathfrak{U}$ considering estimates $\hat{\mu}(x)$ and confidence radius $b_t(x)$ for the arm x in phase i_{ph}. The radius $b_t(x)$ is (c_0, β)–*good *if there exist $c_0 > 0$ and $\beta > 0$, at a given phase i_{ph} s.t. for all $x \in X(i_{ph})$ if $\mu(x^*) - \mu(x) < E_t(x)$ then $n_t(x) \leq c_0(\mu(x^*) - \mu(x))^{-\beta} i_{ph}$.*

At first, we want to show that both these properties are satisfied by the B–Zoom algorithm when applied to problem $(\mathcal{A}, l, \mu)$.

Lemma 3. *Each phase i_{ph} of the B–Zoom algorithm applied to Lipschitz CAB problem $(\mathcal{A}, l, \mu)$ is clean with probability at least $1 - 4^{-i_{ph}}$.*

Proof. The proof will show that the probability of the phase i_{ph} of not being clean is smaller than t^{-4}. Since the B–Zoom algorithm considers estimates $\hat{\mu}(x) := \frac{\bar{r}_t(x)}{\bar{c}_t(x)}$ and radius $b_t(x) := E_t(x)$, we have that:

$$\begin{aligned}
&\mathbb{P}\left[\left|\frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \mu(x)\right| > E_t(x)\middle| x = x_j\right] \\
&= \mathbb{P}\left[\left|\frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \frac{\mu_r(x)}{\mu_c(x)}\right| > E_t(x)\middle| x = x_j\right] \\
&= \mathbb{P}\left[\underbrace{\frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \frac{\mu_r(x)}{\mu_c(x)} > E_t(x)}_{e_1}\middle| x = x_j\right] + \\
&\qquad + \mathbb{P}\left[\underbrace{\frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \frac{\mu_r(x)}{\mu_c(x)} < -E_t(x)}_{e_2}\middle| x = x_j\right].
\end{aligned}$$

The event e_1 implies that at least one of the following two inequalities holds:

- $\bar{r}_t(x) \geq \mu_r(x) + \varepsilon_t(x)$,
- $\bar{c}_t(x) \leq \mu_c(x) - \varepsilon_t(x)$,

where $\varepsilon_t(x) = \sqrt{\frac{8 i_{ph} + \ln 4}{n_t(x)}}$. In fact, if $\bar{r}_t(x) \leq \mu_r(x) + \varepsilon_t(x) \wedge \bar{c}_t(x) \geq \mu_c(x) - \varepsilon_t(x)$ and since $\bar{c}_t(x) \geq \lambda$, $\forall x \in \mathcal{A}$, we have:

$$\begin{aligned}
&\frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \frac{\mu_r(x)}{\mu_c(x)} = \frac{\bar{r}_t(x)\mu_c(x) - \mu_r(x)\bar{c}_t(x)}{\bar{c}_t(x)\mu_c(x)} \\
&\overset{\pm\mu_c(x)\mu_r(x)}{=} \frac{[\bar{r}_t(x) - \mu_r(x)]\mu_c(x) + [\mu_c(x) - \bar{c}_t(x)]\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} \\
&\leq \frac{\varepsilon_t(x)\mu_c(x) + \varepsilon_t(x)\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} = \\
&= \frac{\varepsilon_t(x)}{\bar{c}_t(x)} + \frac{\varepsilon_t(x)\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} \leq \frac{\varepsilon_t(x)}{\lambda} + \frac{\varepsilon_t(x)}{\lambda^2} \\
&= \frac{1}{\lambda}\left(1 + \frac{1}{\lambda}\right)\varepsilon_t(x) = E_t(x).
\end{aligned}$$

The event e_2 implies that at least one of the following two inequalities holds:

- $\bar{r}_t(x) \leq \mu_r(x) - \varepsilon_t(x)$,
- $\bar{c}_t(x) \geq \mu_c(x) + \varepsilon_t(x)$.

In fact, if $\bar{r}_t(x) \geq \mu_r(x) - \varepsilon_t(x) \wedge \bar{c}_t(x) \leq \mu_c(x) + \varepsilon_t(x)$ we have:

$$\begin{aligned}
&\frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \frac{\mu_r(x)}{\mu_c(x)} = \frac{\bar{r}_t(x)\mu_c(x) - \mu_r(x)\bar{c}_t(x)}{\bar{c}_t(x)\mu_c(x)} \\
&\overset{\pm\mu_c(x)\mu_r(x)}{=} \frac{[\bar{r}_t(x) - \mu_r(x)]\mu_c(x) + [\mu_c(x) - \bar{c}_t(x)]\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} \\
&\geq \frac{-\varepsilon_t(x)\mu_c(x) - \varepsilon_t(x)\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} = \\
&= -\frac{\varepsilon_t(x)}{\bar{c}_t(x)} - \frac{\varepsilon_t(x)\mu_r(x)}{\bar{c}_t(x)\mu_c(x)} \geq -\frac{\varepsilon_t(x)}{\lambda} - \frac{\varepsilon_t(x)}{\lambda^2} \\
&= -\frac{1}{\lambda}\left(1 + \frac{1}{\lambda}\right)\varepsilon_t(x) = -E_t(x)
\end{aligned}$$

Thus, we can write:

$$\begin{aligned}
&\mathbb{P}\left[\left|\frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \mu(x)\right| > E_t(x)\middle| x = x_j\right] \leq \\
&\mathbb{P}\left[\bar{r}_t(x) \geq \mu_r(x) + \varepsilon_t(x)\right] + \mathbb{P}\left[\bar{c}_t(x) \leq \mu_c(x) - \varepsilon_t(x)\right] + \\
&+ \mathbb{P}\left[\bar{r}_t(x) \leq \mu_r(x) - \varepsilon_t(x)\right] + \mathbb{P}\left[\bar{c}_t(x) \geq \mu_c(x) + \varepsilon_t(x)\right].
\end{aligned}$$

We can provide a bound to each single term in the r.h.s. of the previous inequality by means of the Hoeffding's bound:

$$\begin{aligned}
&\mathbb{P}\left[\left|\frac{\bar{r}_t(x)}{\bar{c}_t(x)} - \frac{r_t(x)}{c_t(x)}\right| > E_t(x)\middle| x = x_j\right] \\
&\leq \frac{t^{-4}}{4} + \frac{t^{-4}}{4} + \frac{t^{-4}}{4} + \frac{t^{-4}}{4} = t^{-4}
\end{aligned}$$

with $\varepsilon_t(x) = \sqrt{\frac{8 i_{ph} + \ln 4}{n_t(x)}}$.

Taking the union bound over all the $n_t(x) < t$, integrating over $x_j \in [0, 1]$ and taking the union bound over $i \in [0, t]$ concludes the proof. □

Lemma 4. *The radius $E_t(x)$ of the B–Zoom algorithm applied to Lipschitz CAB problem $(\mathcal{A}, l, \mu)$ is (c_0, β)–good with $c_0 = \frac{10(1+\lambda)^2}{\lambda^4}$ and $\beta = 2$.*

Proof. By using the definition of $E_t(x)$ in the B–Zoom algorithm and by defining $\Delta := \mu(x^*) - \mu(x)$:

$$\begin{aligned}
&\Delta < E_t(x) \\
&\Delta < \frac{1}{\lambda}\left(1 + \frac{1}{\lambda}\right)\sqrt{\frac{8 i_{ph} + \ln 4}{n_t(x)}} \\
&\sqrt{\frac{8 i_{ph} + \ln 4}{n_t(x)}} > \frac{\lambda^2 \Delta}{1 + \lambda} \\
&\frac{10 i_{ph}}{n_t(x)} > \frac{\lambda^4 \Delta^2}{(1 + \lambda)^2} \\
&n_t(x) < \frac{10(1 + \lambda)^2}{\lambda^4}\Delta^{-2} i_{ph}
\end{aligned}$$

thus, taking $c_0 = \frac{10(1+\lambda)^2}{\lambda^4}$ and $\beta = 2$ concludes the proof. □

Since both Lemmas 3 and 4 hold, it is possible to use Lemma 4.15 in [13] to bound the regret $\mathcal{R}_\Delta(t)$ of the B–Zoom algorithm applied to the Lipschitz CAB problem $(\mathcal{A}, l, \mu)$.

Theorem 3. *Consider the instance of the Lipschitz MAB problem $(\mathcal{A}, l, \mu)$. Fix any $c > 0$ and let d be the Zooming dimension with multiplier c [13]. The regret $\mathcal{R}_\Delta(t)$ of the B–Zoom algorithm satisfies:*

$$\mathcal{R}_\Delta(t) \leq \bar{C}(\ln(t))^{\frac{1}{d+2}} \cdot t^{\frac{d+1}{d+2}},$$

for any $t > 0$, where $\bar{C}$ is an appropriate constant (depending on c).

Step 2. In what follows, we bound $\mathcal{R}_2$ in terms of $\mathcal{R}_\Delta(t_a)$. It can be observed that, by considering the arm x_t selected by the B–Zoom algorithm at round t, we have the guarantee that at each round t it holds $\mu(x^*) - \mu(x_t) \leq 3E_t(x_t)$. Notice that this inequality does not represent a bound on the instantaneous regret $\mu_r(x^*) - \mu_r(x_t)$. Indeed, the limit of the difference $\mu_r(x^*) - \mu_r(x_t)$ as $3E_t(x_t)$ goes to zero may be a constant $1-\lambda$ (e.g., consider the case: $\mu_r(x^*) = 1$, $\mu_c(x^*) = 1$, $\mu_r(x_t) = \lambda - \varepsilon$ and $\mu_c(x_t) = \lambda$ with $\varepsilon \ll \lambda$; we have $\mu(x^*) - \mu(x_t) = \frac{\varepsilon}{\lambda}$ while $\mu_r(x^*) - \mu_r(x_t) = 1 - \lambda + \varepsilon$), and therefore the results described in [13] cannot be directly applied to bound $\mathcal{R}_2$. Instead, to bound $\mathcal{R}_2$, we restate $\mathcal{R}_2$ as:

$$\begin{aligned}\mathcal{R}_2 &= \mathbb{E}_{r,c}\left[\sum_{t=1}^{t_a^*} r_t(x^*)\right] - \mathbb{E}_{r,c}\left[\sum_{t=1}^{t_a} r_t(x_t)\right] = \\ &= \mathbb{E}_c\left[\mathbb{E}_r\left[\sum_{t=1}^{t_a^*} r_t(x^*)\right] - \mathbb{E}_r\left[\sum_{t=1}^{t_a} r_t(x_t)\right]\right] = \\ &= \mathbb{E}_c\underbrace{\left[\mu_r(x^*)t_a^* - \sum_{i_{ph}}^{\log_2(t_a)} \sum_{x\in X(i_{ph})} \mu_r(x) n_{i_{ph}}(x)\right]}_{\overline{\mathcal{R}}_2},\end{aligned}$$

where $X(i_{ph})$ is the set of active arms in phase i_{ph} and $n_{i_{ph}}(x)$ is the number of rounds we pull x during phase i_{ph}. Let us consider a generic round $t \leq t_a$, with residual budget $B(t) = B - \bar{B} \geq 0$. With notation overload, we denote by $B(x, i_{ph})$ the amount of budget spent by arm x in phase i_{ph}. Each arm $x \in X(i_{ph})$ spent $B(x, i_{ph}) = \mu_c(x) n_{i_{ph}}(x)$ in the phase i_{ph} (in expectation w.r.t. the reward) and $\sum_{i_{ph}=1}^{\log_2(t)} \sum_{x\in X(i_{ph})} B(x, i_{ph}) = \bar{B}$. Let us define $t^*(x) := \frac{\mu_c(x) n_{i_{ph}}(x)}{\mu_c(x^*)}$ for every $x \in X(i_{ph})$ and for every i_{ph}, i.e., the amount of rounds the arm x^* should be pulled to spend a budget of $B(x, i_{ph}) = \mu_c(x) n_{i_{ph}}(x)$. To bound $\mathcal{R}_2$, we need to consider $t = t_a$. It is easy to show that $\sum_{i_{ph}=1}^{\log_2(t_a)} \sum_{x\in X(i_{ph})} t^*(x) = t^*$. Thus, we have:

$$\begin{aligned}\overline{\mathcal{R}}_2 &= \sum_{i_{ph}=1}^{\log_2(t_a)} \sum_{x\in X(i_{ph})} \Big(\mu_r(x^*)t^*(x) - \mu_r(x) n_{i_{ph}}(x)\Big) \\ &= \sum_{i_{ph}=1}^{\log_2(t_a)} \sum_{x\in X(i_{ph})} \left(\mu_r(x^*)\frac{\mu_c(x)}{\mu_c(x^*)} n_{i_{ph}}(x) - \mu_r(x)\frac{\mu_c(x)}{\mu_c(x)} n_{i_{ph}}(x)\right) \\ &= \sum_{i_{ph}=1}^{\log_2(t_a)} \sum_{x\in X(i_{ph})} \left(\frac{\mu_r(x^*)}{\mu_c(x^*)} - \frac{\mu_r(x)}{\mu_c(x)}\right) n_{i_{ph}}(x)\mu_c(x) \\ &\leq \sum_{i_{ph}=1}^{\log_2(t_a)} \sum_{x\in X(i_{ph})} \left(\frac{\mu_r(x^*)}{\mu_c(x^*)} - \frac{\mu_r(x)}{\mu_c(x)}\right) n_{i_{ph}}(x) = \mathcal{R}_\Delta(t_a),\end{aligned}$$

where the last inequality holds since $\mu_c(x) \leq 1$ for every $x \in \mathcal{A}$.

Step 3. In this step we formulate the regret for the Lipschitz CAB problem $(\mathcal{A}, l, \mu)$, previously defined on the basis of t_a, as depending on B. Initially, we state:

Lemma 5. *For each $\delta \in (0,1)$, with probability at least $1-\delta$, the number of rounds t used to spend a budget of $\bar{B}$ is:*

$$t \leq \frac{\bar{B}}{\lambda} + \frac{2(1-\lambda)}{\lambda\sqrt{\lambda}}\sqrt{\bar{B}\ln(1/\delta)} + \left(\frac{1-\lambda}{\lambda}\right)^2 \ln(1/\delta).$$

Proof. Consider the unbiased estimator of the average cost $\frac{\sum_{i=1}^t c_i(x_i)}{t}$ until round t, we have:

$$\begin{aligned}&\mathbb{P}\left(\frac{\sum_{i=1}^t c_i(x_i)}{t} \leq \lambda - \varepsilon\right) \\ &\leq \mathbb{P}\left(\frac{\sum_{i=1}^t c_i(x_i)}{t} \leq \frac{\sum_{i=1}^t \mu_c(x_i)}{t} - \varepsilon\right)\end{aligned}$$

since $\mu_c(x) > \lambda$ for every $x \in \mathcal{A}$. Thus, we can bound the r.h.s. of the previous equation by using the Hoeffding's bound:

$$\mathbb{P}\left(\frac{\sum_{i=1}^t c_i(x_i)}{t} \leq \lambda - \varepsilon\right) \leq \delta \to \varepsilon = \sqrt{\frac{\ln(1/\delta)(1-\lambda)^2}{2t}}$$

At round t and with probability at least $1-\delta$ the budget spent is:

$$\begin{aligned}&\bar{B} \geq t\left(\lambda - \sqrt{\frac{\ln(1/\delta)(1-\lambda)^2}{2t}}\right) \\ &2\lambda t - \sqrt{2\ln(1/\delta)(1-\lambda)^2}\, t^{1/2} - 2\bar{B} \leq 0 \\ &t^{1/2} \leq \frac{\sqrt{2\ln(1/\delta)(1-\lambda)^2} + \sqrt{2\ln(1/\delta)(1-\lambda)^2 + 16\lambda\bar{B}}}{4\lambda} \\ &t^{1/2} \leq \frac{\sqrt{2\ln(1/\delta)(1-\lambda)^2} + \sqrt{2\ln(1/\delta)(1-\lambda)^2} + \sqrt{16\lambda\bar{B}}}{4\lambda} \\ &t \leq \left(\frac{4(1-\lambda)\sqrt{\ln(1/\delta)} + 4\sqrt{\lambda\bar{B}}}{4\lambda}\right)^2 \\ &t \leq \left(\sqrt{\frac{\bar{B}}{\lambda}} + \frac{(1-\lambda)\sqrt{\ln(1/\delta)}}{\lambda}\right)^2 \\ &t \leq \frac{\bar{B}}{\lambda} + \frac{2(1-\lambda)}{\lambda\sqrt{\lambda}}\sqrt{\bar{B}\ln(1/\delta)} + \left(\frac{1-\lambda}{\lambda}\right)^2 \ln(1/\delta)\end{aligned}$$

which concludes the proof. □

Since the bound in Lemma 5 holds also for the stopping round t_a (when the budget spent is B), by considering $\delta = B^{-\frac{1}{d+2}}$ and $\ln(B) \leq B$ we have:

$$\begin{aligned}&t_a \leq \frac{B}{\lambda} + \frac{2(1-\lambda)}{\lambda\sqrt{\lambda}}\sqrt{B\ln(1/\delta)} + \left(\frac{1-\lambda}{\lambda}\right)^2 \ln(1/\delta) \\ &t_a \leq \frac{B}{\lambda} + \frac{2(1-\lambda)}{\lambda\sqrt{\lambda}\sqrt{d+2}} B + \left(\frac{1-\lambda}{\lambda}\right)^2 \frac{B}{d+2} \\ &t_a \leq \frac{6}{\lambda^2(d+2)} B\end{aligned}$$

Finally, by taking the expectation over time (or over costs equivalently), we have that there exists constant $\tilde{C}$ (depending on λ, d and c) s.t.:

$$\begin{aligned}\mathcal{R}(B) &\leq \mathcal{R}_1 + (1-\delta)\overline{\mathcal{R}}_2 + \delta B \\ &\leq \mathcal{R}_1 + (1-\delta)\mathcal{R}_\Delta(t_a) + \delta B \\ &\leq \frac{2}{\lambda} + (1-\delta)\bar{C}(\ln(t_a))^{\frac{1}{d+2}} \cdot t_a^{\frac{d+1}{d+2}} + \delta B \\ &\leq \frac{2}{\lambda} + \bar{C}(\ln(B))^{\frac{1}{d+2}} \cdot B^{\frac{d+1}{d+2}}) + \delta B \\ &\leq \tilde{C}(\ln(B))^{\frac{1}{d+2}} \cdot B^{\frac{d+1}{d+2}}\end{aligned}$$

which concludes the proof. □

4 Experimental analysis

In this section, we evaluate the empirical performance of the B–Zoom algorithm. Our evaluation is twofold and it is based on the comparison with other frequentist algorithms, ours being of this class. In particular, we compare the performance of the B–Zoom algorithm (denoted BZ for short) w.r.t. the one of the Zooming algorithm [12] (denoted Z for short) suited for CAB problems, analyzing empirically the impact of taking into account explicitly information about budget and costs. We recall indeed that, in the worst–case analysis, the Zooming algorithm performs arbitrarily worse than the B–Zoom one if a budget constraint is present, since it is assured to find the arm maximizing the expected reward, which in general is different from the optimal reward–to–cost ratio optimal arm x^*. Furthermore, we empirically analyze the impact of exploiting information about the continuous structure of the arm space by comparing the performance of the B–Zoom algorithm w.r.t. the UCBBV1 algorithm [10], designed for BMAB problems, and the UCB1 [3] one, designed for MAB problems, both applied to a finite set of arms obtained by some discretization of the arm space $\mathcal{A}$ and kept fixed for all the rounds. We recall that in the worst–case analysis the UCBBV1 and UCB1 algorithms applied to a finite set of arms randomly drawn from the arm space perform arbitrarily worse than the B–Zoom algorithm when expected reward and costs are Lipschitz. Indeed, consider the case the reward–to–cost ratio $\mu(x)$ is flat except for a small–supported peak in which there is the optimum. The UCBBV1 algorithm might perform as good as a random choice when the finite set of arms is such that all the arms are positioned in the flat part of $\mu(x)$.

In what follows, we consider the *cumulative profit* of a policy $\mathfrak{U}$ over a fixed budget B as figure of merit, defined as:

$$P_{\mathfrak{U}}(B) = \sum_{t=1}^{t_a} r_t(x_t).$$

Experimental setting To provide a thorough experimental evaluation of each algorithm, we consider different settings with budget $B \in \{50,000; 100,000; 500,000\}$ and with $\lambda \in \{0.05; 0.10; 0.25; 0.50\}$. We select the average reward–to–cost ratio functions $\mu(x)$ s.t. the average reward $\mu_r(x)$ and cost $\mu_c(x)$ are:

$$\mu_r(x) = 0.05x + 0.95,$$
$$\mu_c(x) = \lambda + \frac{1}{5}\left(1 - e^{-500(x-\tilde{x})^2}\right),$$

where $\tilde{x}$ is the arm with the lowest expected cost. The value for the arm $\tilde{x}$ is sampled for each of the experiments from a uniform distribution over $[0, 1]$. The instantaneous reward $r_t(x)$ and cost $c_t(x)$ for pulling an arm x are sampled from Bernoulli distributions, i.e., $R_t(x) \sim Be(\mu_r(x))$ and $C_t(x) \sim Be(\mu_c(x))$, respectively. The above functions are modeling a setting where the reward is linearly increasing in the value $x \in \mathcal{A}$ of the arms and the cost is constant except for a small area around $\tilde{x}$, where it is sensibly lower, which generates a unimodal reward–to–cost ratio function. For each pair of values of B and λ, we generate 100 different instances characterized by potentially different reward–to–cost ratio functions. Given the empirical profit obtained by an algorithm over the 100 instances, we compute the empirical mean $\bar{P}$, the minimum m, the 25–th percentile Q_1 (first quartile) and the 50–th percentile Q_2 (second quartile or median value).

For the UCBBV1 and UCB1 algorithms we use different numbers of arms $K \in \{5, 10, 15\}$ randomly placed over the arm space $\mathcal{A}$. For sake of comparison, we here adopt a version of the B–Zoom and Zooming algorithms that do not consider exponentially long phases i_{ph}, but we use a single phase and a confidence radius equal to $E_t(x) = \frac{1}{\lambda}\left(1 + \frac{1}{\lambda}\right)\sqrt{\frac{8(\ln B - \ln\lambda) + \ln 4}{n_t(x)}}$, where the term $\frac{B}{\lambda}$ is a rough estimation of the average stopping time of the learning process. The experiments here reported have been performed in MATLAB.

Table 1 Results for the cumulative profit provided in thousand of reward units. The highest cumulative profit for each row is highlighted in bold.

		P_{BZ}	P_Z	P_{UCBBV1}			P_{UCB1}		
K		–	–	5	10	15	5	10	15
$\lambda = 0.05$	m	**203**	199	198	199	199	198	199	198
	Q_1	**212**	200	200	203	204	201	207	203
	Q_2	215	201	204	215	213	206	**216**	209
	$\bar{P}$	217	212	**266**	236	216	215	216	211
$\lambda = 0.1$	m	**170**	165	165	165	166	165	166	166
	Q_1	**175**	166	167	168	172	167	168	169
	Q_2	180	167	173	184	**192**	171	174	174
	$\bar{P}$	181	173	**230**	224	219	178	176	176
$\lambda = 0.25$	m	**113**	110	110	111	110	110	110	111
	Q_1	117	111	111	112	**118**	111	112	112
	Q_2	119	111	114	129	**144**	112	114	115
	$\bar{P}$	119	115	134	139	**142**	115	115	115
$\lambda = 0.5$	m	**72**	71	71	71	71	71	71	71
	Q_1	73	71	71	72	**74**	71	72	72
	Q_2	74	71	72	77	**85**	72	73	73
	$\bar{P}$	74	74	78	80	**83**	73	73	73
$B = 50,000$									
$\lambda = 0.05$	m	**413**	397	398	398	400	397	397	398
	Q_1	**428**	400	400	405	414	400	410	411
	Q_2	442	401	408	429	**459**	413	432	423
	$\bar{P}$	442	426	542	**554**	520	427	436	426
$\lambda = 0.1$	m	**347**	331	330	332	332	332	332	332
	Q_1	**362**	333	334	336	342	334	337	341
	Q_2	372	334	345	386	**430**	341	353	348
	$\bar{P}$	374	351	463	476	**494**	353	353	352
$\lambda = 0.25$	m	**232**	221	221	221	221	221	222	221
	Q_1	**242**	222	223	224	236	223	225	226
	Q_2	246	222	231	263	**311**	225	231	230
	$\bar{P}$	247	230	275	284	**301**	230	231	231
$\lambda = 0.5$	m	**145**	142	142	143	143	142	142	142
	Q_1	**150**	143	143	143	147	143	143	144
	Q_2	152	143	144	161	**171**	144	146	145
	$\bar{P}$	152	147	156	164	**168**	146	146	146
$B = 100,000$									
$\lambda = 0.05$	m	**2129**	1995	1995	1997	1998	1996	1994	1995
	Q_1	**2249**	2001	2002	2037	2083	2001	2042	2060
	Q_2	2368	2005	2050	**2999**	2654	2088	2155	2123
	$\bar{P}$	2376	2111	3383	**4372**	4022	2171	2162	2146
$\lambda = 0.1$	m	**1923**	1661	1662	1664	1666	1662	1664	1664
	Q_1	**2120**	1666	1667	1682	1788	1667	1687	1707
	Q_2	2308	1669	1700	2201	**3076**	1675	1757	1769
	$\bar{P}$	2267	1725	2457	2758	**3082**	1756	1769	1768
$\lambda = 0.25$	m	**1360**	1107	1109	1110	1109	1108	1110	1109
	Q_1	**1432**	1111	1111	1187	1248	1112	1115	1124
	Q_2	1471	1112	1142	1599	**1743**	1123	1145	1142
	$\bar{P}$	1461	1142	1343	1554	**1608**	1153	1152	1149
$\lambda = 0.5$	m	**784**	712	713	713	714	713	713	712
	Q_1	**807**	714	715	718	800	715	716	719
	Q_2	821	715	730	874	**928**	722	729	729
	$\bar{P}$	816	724	808	845	**890**	735	730	731
$B = 500,000$									

Results We report in Table 1 the results of our experiments. Initially, we focus on the empirical mean $\bar{P}$. The B–Zoom algorithm outperforms the Zooming one for all the configurations, providing a larger profit up to about 30%. This was expected since the B–Zoom algorithm exploits more information than the Zooming algorithm. Similar results can be observed when compared with the UCB1. Unexpectedly, the UCBBV1 applied to a randomly gener-

ated set of arms outperforms the B–Zoom algorithm for all the configurations, providing a larger profit up to about 40%. In order to understand the reasons behind such a behavior, we need to focus on the other indices: m, Q_1, and Q_2. The B–Zoom algorithm in most of the cases outperforms the other algorithms for the indices m and Q_1. More specifically, with $B = 50,000$ all the minimum values m provided by the B–Zoom algorithm are higher than the one achieved by other algorithms and the difference in terms of cumulative profit increases as the minimum average cost λ decreases. In this setting, when $\lambda = 0.05$ and $\lambda = 0.10$ even the first quartile Q_1 has higher values. Conversely, in terms of median (Q_2), the UCBBV1 algorithm with $K = 15$ arms provides the largest profit in most of the settings, despite being less robust to unfavorable cases. Similar results also hold for the settings with $B = 100,000$ and $B = 500,000$. These results suggest that the B–Zoom algorithm is the most robust algorithm in the worst case, assuring the best performance for m and Q_1, but, in order to be robust, it must pay a cost in the average case (and in some situations in the median case). This suggests also that, in our experiments, the worst case occurs with low probability, otherwise the B–Zoom algorithm would provide good performance also for the empirical mean $\bar{P}$. This is clear by observing Figure 1, where we report the boxplot for the case with $B = 100,000$ and $\lambda = 0.10$: the B–Zoom algorithm provides the best performance at m and Q_1, but it also presents a compact distribution with an extremely low variance. Instead, the performance of the UCBBV1 algorithm presents a very large variance that allows it to have both poor and excellent profits.

By analyzing how different values of initial budgets B affect the performance of the algorithms, we can observe how the B–Zoom algorithm is able to improve the minimum cumulative profit from approximately 1% in the case $B = 50,000$ scenario to more than 10% in the case $B = 500,000$. This behavior was expected since we have assurance of convergence to the optimal solution for the B–Zoom algorithm, while an algorithm relying on a fixed discretization of the space or considering a different minimization objective function (loss of cumulative reward) might not converge to the optimal arm.

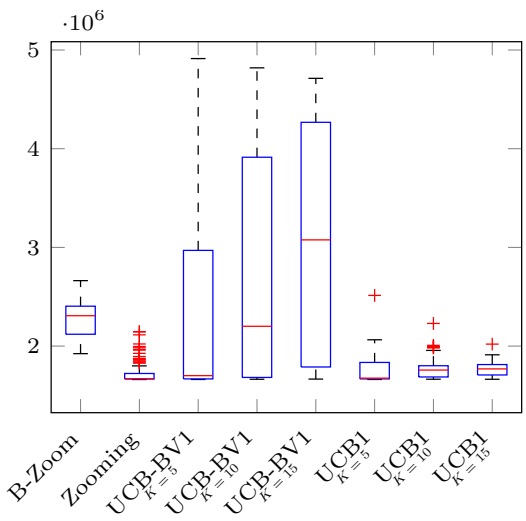

Figure 1. Boxplots of $P_{\mathfrak{U}}(500,000)$ for different algorithms with $\lambda = 0.1$

Summarily, the theoretical guarantee over the regret minimization of the B–Zoom algorithm represents an intrinsic limit to outperform other algorithms that do not have any theoretical guarantee in terms of mean empiric profit. However, the B–Zoom algorithm is more robust w.r.t. the other algorithms in terms of the minimum m and first quartile Q_1 of cumulative profit. In principle, this makes the B–Zoom algorithm more suitable for situations in which the learner is risk averse.

Table 2 Results for the cumulative profit with fixed cost $\lambda = 0.5$, provided in thousands of reward units. The highest cumulative profit for each budget and row is highlighted in bold.

	P_Z	P_{UCB1}			P_Z	P_{UCB1}			P_Z	P_{UCB1}		
K	–	5	10	15	–	5	10	15	–	5	10	15
m	**0.7**	0.0	0.0	0.0	**2.6**	0.0	0.0	0.0	**43.3**	0.0	0.0	0.0
Q_1	1.4	0.0	0.0	**1.5**	**4.0**	0.0	0.4	2.4	**48.6**	0.0	8.0	27.2
Q_2	2.0	0.1	2.9	**5.7**	4.7	0.3	5.7	**11.7**	53.2	14.7	46.3	**70.7**
$\bar{P}$	1.9	2.2	3.4	**4.7**	5.0	4.6	7.2	**9.3**	51.9	32.3	48.6	**60.1**
	$B = 50,000$				$B = 100,000$				$B = 500,000$			

Additional results On the basis of the results described above, we investigate whether the poor performance in terms of empiric mean profit of the B–Zoom algorithm is intrinsic in the need for being robust to the worst case with continuous arm space independently of the presence of budget constraints or it is due exclusively to the presence of budget constraints in continuous arm space. To evaluate this issue, we compare the performance of an algorithm suited for the CAB case, i.e., the Zooming algorithm, versus the one provided by a discrete MAB, i.e., the UCB1, once a random discretization of the space is applied reducing the budget constraint to a time horizon constraint. We consider a setting with fixed costs for all the arms $C_t(x) = \lambda, \forall x, t$ and rewards $R_t(x)$ drawn from Bernoulli distributions with expected value $\mu_r(x) = \frac{1}{5} e^{-500(x-\tilde{x})^2}$ and $\tilde{x}$ is uniformly drawn from $[0,1]$, where the optimal arm $\tilde{x}$ for the reward–to–cost ratio function coincides with the optimal arm for the reward function. In this way, the BCAB problem reduces to a CAB problem with $T = B/\lambda$. We repeat the experiments for 100 independent runs.

We report our experimental results in Table 2. Even in these experiments the continuous approach is able to provide a risk–averse alternative to the discretized ones, at the expense of loss in terms of average performance. Again, the values for the minimum m is always higher and the first quartile Q_1 is higher in the case we have a larger budget. This behaviour is explained by the fact that the Zooming algorithm always adds arms over the whole space to cope with possible worst–case settings, which decreases its average performance. The loss due to the introduction of such arms is balanced by the convergence to the optimal arm, which asympotically provides higher profits and at the same time is able not to reduce the losses in unfavorable settings.

5 Conclusions and future works

In this paper, we present a new problem, the Budgeted Continuous–Armed Bandit (BCAB), and an algorithm, the B–Zoom, specifically suited for this setting. We study the proposed algorithm both in terms of theoretical properties and empirical performances. While it suffers a regret of $\tilde{O}(B^{\frac{d+1}{d+2}})$, it is able to provide empirical evidence that it is more risk averse than the algorithms present in the literature of BMAB.

Some of the most promising works for future research are: introducing a vector of costs and a stopping round dependent on a combination of these costs. Moreover, we may explore the problem of having a search space $\mathcal{A}$ with more than one dimension. Finally, it could be interesting to extend other existing algorithms for the CAB setting to the BCAB problem.

REFERENCES

[1] András Antos, Varun Grover, and Csaba Szepesvári, 'Active learning in multi-armed bandits', in *Proceedings of the International Conference on Algorithmic Learning Theory, ALT*, pp. 287–302. Springer, (2008).

[2] Danilo Ardagna, Barbara Panicucci, and Mauro Passacantando, 'A game theoretic formulation of the service provisioning problem in cloud systems', in *Proceedings of the International Conference on World Wide Web, WWW*, pp. 177–186, (2011).

[3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer, 'Finite-time analysis of the multiarmed bandit problem', *Machine learning*, **47**(2-3), 235–256, (2002).

[4] Peter Auer, Ronald Ortner, and Csaba Szepesvári, 'Improved rates for the stochastic continuum-armed bandit problem', in *Proceedings of the Annual Conference on Learning Theory, COLT*, pp. 454–468, (2007).

[5] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins, 'Bandits with knapsacks', in *Proceedings of the Annual Symposium on Foundations of Computer Science, FOCS*, pp. 207–216, (2013).

[6] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Kamal Jain, Omid Etesami, and Mohammad Mahdian, 'Dynamics of bid optimization in online advertisement auctions', in *Proceedings of the International Conference on World Wide Web, WWW*, pp. 531–540, (2007).

[7] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz, 'Pure exploration in multi-armed bandits problems', in *Proceedings of the International Conference on Algorithmic Learning Theory, ALT*, pp. 23–37, (2009).

[8] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvari, 'X-armed bandits', *The Journal of Machine Learning Research*, **12**, 1655–1695, (2011).

[9] Sébastien Bubeck, Gilles Stoltz, and Jia Yuan Yu, 'Lipschitz bandits without the lipschitz constant', in *Proceedings of the International Conference on Algorithmic Learning Theory, ALT*, pp. 144–158, (2011).

[10] Wenkui Ding, Tao Qin, Xu-Dong Zhang, and Tie-Yan Liu, 'Multi-armed bandit with budget constraint and variable costs', in *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI*, pp. 232–238, (2013).

[11] Sudipto Guha and Kamesh Munagala, 'Approximation algorithms for budgeted learning problems', in *Proceedings of the Symposium on Theory of Computing, STOC*, pp. 104–113. ACM, (2007).

[12] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal, 'Multi-armed bandits in metric spaces', in *Proceedings of the Symposium on Theory of Computing, STOC*, pp. 681–690, (2008).

[13] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal, 'Bandits and experts in metric spaces', *arXiv preprint arXiv:1312.1277*, (2013).

[14] Robert D. Kleinberg, 'Nearly tight bounds for the continuum-armed bandit problem', in *Proceedings of Neural Information Processing Systems, NIPS*, pp. 697–704, (2004).

[15] Stefan Magureanu, Richard Combes, and Alexandre Proutiere, 'Lipschitz bandits: Regret lower bound and optimal algorithms', in *Proceedings of the Conference on Learning Theory, COLT*, pp. 975–999, (2014).

[16] Paritosh Padhy, Rajdeep K Dash, Kirk Martinez, and Nicholas R Jennings, 'A utility-based adaptive sensing and multihop communication protocol for wireless sensor networks', *Transactions on Sensor Networks*, **6**(3), 27:1–27:39, (2010).

[17] Long Tran-Thanh, Archie Chapman, Alex Rogers, and Nicholas R Jennings, 'Knapsack based optimal policies for budget–limited multi–armed bandits', in *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI*, pp. 1134–1140, (2012).

[18] Long Tran-Thanh, Alex Rogers, and Nicholas R Jennings, 'Long-term information collection with energy harvesting wireless sensors: a multi-armed bandit based approach', in *Proceedings of the Autonomous Agents and Multi-Agent Systems, AAMAS*, volume 25, pp. 352–394, (2012).

[19] Yingce Xia, Wenkui Ding, Xu-Dong Zhang, Nenghai Yu, and Tao Qin, 'Budgeted bandit problems with continuous random costs', in *Proceedings of the Asian Conference on Machine Learning, ACML*, pp. 317–332, (2015).

[20] Yingce Xia, Haifang Li, Tao Qin, Nenghai Yu, and Tie-Yan Liu, 'Thompson sampling for budgeted multi-armed bandits', in *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*, pp. 3960–3966, (2015).