# Mortal Multi-Armed Bandits

Deepayan Chakrabarti, Yahoo! Research
Ravi Kumar, Yahoo! Research
Filip Radlinski, Microsoft Research
Eli Upfal, Brown University

## Abstract

We study a new variant of the k-armed bandit problem, motivated by e-commerce applications. In our model, arms have a lifetime, after which they expire.

- **The search algorithm needs to continuously explore new arms,**
  Contrasts with standard k-armed bandit settings, where exploration is reduced once search narrows to good arms.
- **The algorithm needs to choose among a large collection of arms,**
  - More than can be fully explored within the typical arm lifetime.

We present:
- An optimal algorithm for the deterministic reward case,
- Obtain a number of algorithms for the stochastic reward case.
- Show that the proposed algorithms significantly outperform standard multi-armed bandit approaches given various reward distributions.

## Introduction

- In online advertising, ad brokers select ads to display from a large corpus, with the goal to generate the most ad clicks and revenue.
- Previous work has suggested considering this as a multi-armed bandit problem. [Pandey et al, 2007].

### Multi-Armed Bandits

- Models a casino with k slot machines (one-armed bandits).
- Each machine has an unknown expected payoff.
- The goal is to select the optimal sequence of slot machines to play to maximize the expected total reward, or minimize regret: How much we could have made but didn't.

### How is this like advertising?

- Show ads is like pulling arms: It has a cost, and a possible reward.
- We want an algorithm to select the best sequence of ads to show to maximize the (expected) financial reward.

### How is advertising harder?

- A standard assumption is that arms exists perpetually.
- The expect payoff is allowed to change, but only slowly.
- Ads, on the other hand, are constantly being created and removed from circulation: budgets run out, seasons change, etc.
- There are too many ads to explore in a typical ad lifetime.

Arm with expected payoff $\mu_i$ provides a reward when pulled:
**Deterministic setting:** *reward*($\mu_i$) = $\mu_i$
**Stochastic setting:** *reward*($\mu_i$) = *1* with prob. $\mu_i$, *0* otherwise.

Two forms of death are studied:
**Budgeted**: lifetime $L_i$ of arms is known to alg., only pulls count.
**Timed**: each arm has probability *p* of dying each time step.

### Related approaches

- Restless Bandits [e.g. Whittle; Bertsimas; Nino-Mora; Slivkins & Upfal]: Arms rewards change over time.
- Sleeping bandits / experts [e.g. Freund et al.; Blum & Mansour; Kleinberg et al]: A subset of arms is available at each time step.
- New arms appearing [e.g. Whittle]: There is an optimal index policy.
- Infinite arm supply [e.g. Berry et al.; Teytaud et al.; Kleinberg; Krause & Guestrin]: Too many arms to explore completely.

## Upper Bound on Mortal Reward

Consider the deterministic reward, budgeted death case. Assume fresh arms are always available.

Let $\bar{\mu}(t)$ denote the maximum mean reward that any algorithm for this case can obtain in t steps. Then $\lim_{t\to\infty} \bar{\mu}(t) \le \max_\mu \Gamma(\mu)$ where

$$\Gamma(\mu) = \frac{E[X] + (1-F(\mu))(L-1)E[X \mid X \ge \mu]}{1+(1-F(\mu))(L-1)}$$

and L is the expected arm lifetime and $F(\mu)$ is the cumulative distribution of arm payoffs.

In the stochastic reward, and timed death cases, we can do no better.

Example cases:
**1.** Say arm payoff is 1 with probability p<0.5, 1-δ otherwise. Say arms have probability p of dying each time step. The mean reward per step is at most 1- δ+ δp, while maximum reward is 1. Hence regret per step is $\Omega(1)$.
**2.** Suppose F(x) = x with x∊[0,1]. Suppose arms have probability *p* of dying each time step. The mean reward per step is bounded by $(1-\sqrt{p})/(1-p)$, expected regret of any algorithm is $\Omega(\sqrt{p})$.

## Bandit Algorithms for Mortal Arms

### DetOpt: Optimal for the deterministic reward case

In the deterministic case, we can try new arms once until we find a good one:

**Algorithm** DETOPT
**input:** Distribution $F(\mu)$, expected lifetime $L$
$\mu^* \leftarrow \operatorname{argmax}_\mu \Gamma(\mu)$ [*$\Gamma$ is defined in (1)*]
**while** we keep playing
  $i \leftarrow$ random new arm
  Pull arm $i$; $R \leftarrow R(\mu_i) = \mu_i$
  **if** $R > \mu^*$ [*If arm is good, stay with it*]
    Pull arm $i$ every turn until it expires
  **end if**
**end while**

Let DEPOPT(t) denote the mean reward per turn obtained by DetOpt after running for t steps with $\mu^* = \operatorname{argmax}_\mu \Gamma(\mu)$. Then $\lim_{t\to\infty}$ DEPOPT(t) $= \max_\mu \Gamma(\mu)$

### DetOpt for stochastic reward case, with early stopping:

In the stochastic case, we can just try new arms up to n times before deciding if to move on:

**Algorithm** STOCH. WITH EARLY STOPPING
**input:** Distribution $F(\mu)$, expected lifetime $L$
$\mu^* \leftarrow \operatorname{argmax}_\mu \Gamma(\mu)$ [*$\Gamma$ is defined in (1)*]
**while** we keep playing
  [*Play random arm as long as necessary*]
  $i \leftarrow$ random new arm; $r \leftarrow 0$; $d \leftarrow 0$
  **while** $d < n$ **and** $n - d \ge n\mu^* - r$
    Pull arm $i$; $r \leftarrow r + R(\mu_i)$; $d \leftarrow d + 1$
  **end while**
  **if** $r > n\mu^*$ [*If it is good, stay with it forever*]
    Pull arm $i$ every turn until it dies
  **end if**
**end while**

For $n = O(\log L/\varepsilon^2)$, STOCHASTIC(without early stopping) gets an expected reward per step of $\Gamma(\mu^* - \varepsilon)$

## Subset Heuristics & Greedy

Standard Multi-Armed Bandit algorithms trade off exploration and exploitation well. The problem with mortal arms is that there are too many options. Can we avoid that?

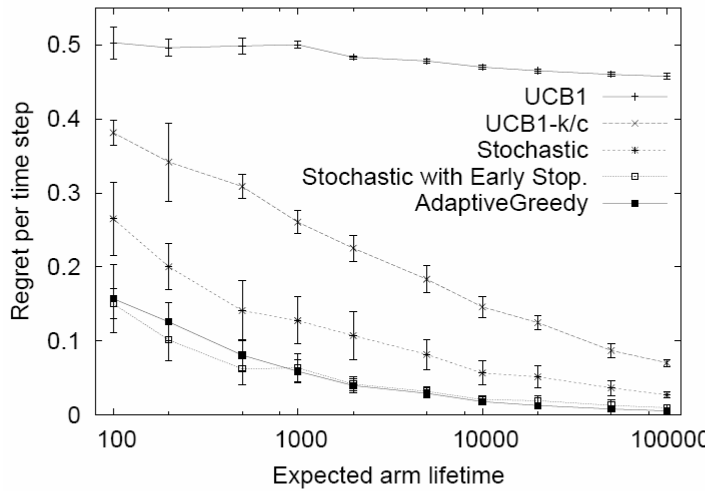
**Algorithm** UCB1K/C
**input:** $k$-armed bandit, $c$
**while** we keep playing
  $S \leftarrow k/c$ random arms
  $dead \leftarrow 0$
  $A^{UCB1}(S) \leftarrow$ Initialize UCB1 over arms $S$
  **repeat**
    $i \leftarrow$ arm selected by $A^{UCB1}(S)$
    Pull arm $i$, provide reward to $A^{UCB1}(S)$
    $x \leftarrow$ total arms that died this turn
    Check for newly dead arms in $S$, remove any
    $dead \leftarrow dead + x$
  **until** $dead \ge k/2$ **or** $|S| = 0$
**end while**

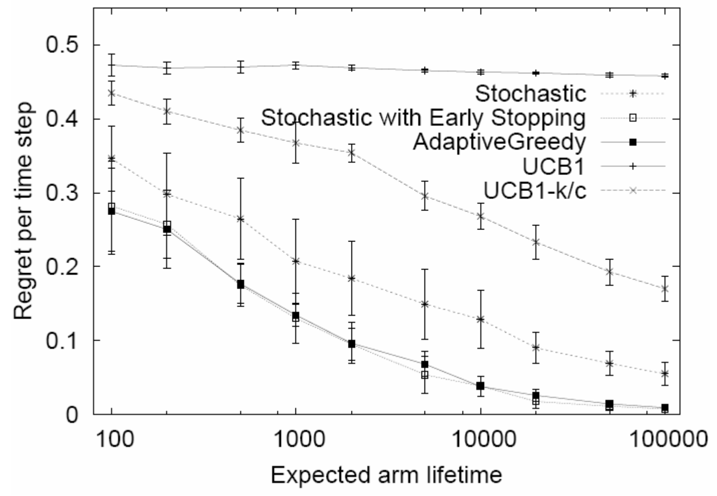
**Algorithm** ADAPTIVEGREEDY
**input:** $k$-armed bandit, $c$
Initialization: $\forall i \in [1,k], r_i, n_i \leftarrow 0$
**while** we keep playing
  $m \leftarrow \operatorname{argmax}_i r_i/n_i$ [*Find best arm so far*]
  $p_m \leftarrow r_m/n_m$
  With probability $\min(1, c \cdot p_m)$
    $j \leftarrow m$
  Otherwise [*Pull a random arm*]
    $j \leftarrow \text{uniform}(1,k)$
  $r \leftarrow R(j)$
  $r_j \leftarrow r_j + r$ [*Update the observed rewards*]
  $n_j \leftarrow n_j + 1$
**end while**

Picking the theoretically best subset size and epoch length is still an open problem.

In many empirical studies, greedy algorithms also perform well on average due to the lack of exploration that is needed for worst-case performance guarantees. AdaptiveGreedy is one such algorithm.

## Empicial Evaluation




Simulated with k=1000 arms, for time duration 10 times the expected lifetime of each arm. Simulating k=100,000 arms gives similar results.

With F(x) = x (top):
- UCB1 performs poorly
- Subset heuristic helps
- Stochastic with early stopping performs equally best with Adaptive Greedy.

We see a similar picture with F(x) matching real advertisements (bottom).

Similar performance is seen whenF(X) is distributed as beta(1,3).

Mortal Multi-Armed Bandits model the realistic case when strategies are sometimes permanently removed.

- Sublinear regret is impossible.
- We presented algorithms and analysis for this setting.