A Simple Distribution-Free Approach to the Max k-Armed Bandit Problem

Matthew J. Streeter[1] and Stephen F. Smith[2]

Computer Science Department and
Center for the Neural Basis of Cognition[1] and
The Robotics Institute[2]
Carnegie Mellon University
Pittsburgh, PA 15213
{matts,sfs}@cs.cmu.edu

Abstract. The max k-armed bandit problem is a recently-introduced online optimization problem with practical applications to heuristic search. Given a set of k slot machines, each yielding payoff from a fixed (but unknown) distribution, we wish to allocate trials to the machines so as to maximize the maximum payoff received over a series of n trials. Previous work on the max k-armed bandit problem has assumed that payoffs are drawn from *generalized extreme value* (GEV) distributions. In this paper we present a simple algorithm, based on an algorithm for the classical k-armed bandit problem, that solves the max k-armed bandit problem effectively without making strong distributional assumptions. We demonstrate the effectiveness of our approach by applying it to the task of selecting among priority dispatching rules for the resource-constrained project scheduling problem with maximal time lags (RCPSP/max).

1 Introduction

In the classical k-armed bandit problem one is faced with a set of k slot machines, each having an arm that, when pulled, yields a payoff drawn independently at random from a fixed (but unknown) distribution. The goal is to allocate trials to the arms so as to maximize the cumulative payoff received over a series of n trials. Solving the problem entails striking a balance between exploration (determining which arm yields the highest mean payoff) and exploitation (repeatedly pulling this arm).

In the max k-armed bandit problem, the goal is to maximize the *maximum* (rather than cumulative) payoff. This version of the problem arises in practice when tackling combinatorial optimization problems for which a number of randomized search heuristics exist: given k heuristics, each yielding a stochastic outcome when applied to some particular problem instance, we wish to allocate trials to the heuristics so as to maximize the maximum payoff (e.g., the maximum number of clauses satisfied by any sampled variable assignment, the minimum makespan of any sampled schedule). Cicirello and Smith (2005) show that a max k-armed bandit approach yields good performance on the resource-constrained project scheduling problem with maximum time lags (RCPSP/max).

1.1 Motivations

When solving the classical k-armed bandit problem, one can provide meaningful performance guarantees subject only to the assumption that payoffs are drawn from a bounded interval, for example $[0, 1]$. In the max k-armed bandit problem stronger distributional assumptions are necessary, as illustrated by the following example.

Example 1. There are two arms. One returns payoff $\frac{1}{2}$ with probability 0.999, and payoff 1 with probability 0.001; the other returns payoff $\frac{1}{2}$ with probability 0.995 and payoff 1 with probability 0.005. It is not known which arm is which.

Given a budget of n pulls of the two arms described in Example 1, a variety of techniques are available for (approximately) maximizing the cumulative payoff received. However, any attempt to maximize the *maximum* payoff received over n trials is hopeless. No information is gained about any of the arms until a payoff of 1 is obtained, at which point the maximum payoff cannot be improved.

Previous work on the max k-armed bandit problem has assumed that payoffs are drawn from *generalized extreme value* (GEV) distributions. A random variable Z has a GEV distribution if

$$\mathbb{P}[Z \leq z] = \exp\left(-\left(1 + \frac{\xi(z-\mu)}{\sigma}\right)^{-\frac{1}{\xi}}\right)$$

for some constants μ, $\sigma > 0$, and ξ.

The assumption that payoffs are drawn from a GEV is justified by the Extremal Types Theorem [6], which singles out the GEV as the limiting distribution of the maximum of a large number of independent identically distributed (i.i.d.) random variables. Roughly speaking, one can think of the Extremal Types Theorem as an analogue of the Central Limit Theorem. Just as the Central Limit Theorem states that the sum of a large number of i.i.d. random variables converges in distribution to a Gaussian, the Extremal Types Theorem states that the maximum of a large number of i.i.d. random variables converges in distribution to a GEV. Despite this asymptotic guarantee, we will see in §4 that the GEV is often not even an approximately accurate model of the payoff distributions encountered in practice.

In this work, we do not assume that the payoff distributions belong to any specific parametric family. In fact, we will not make any formal assumptions at all about the payoff distributions, although (as shown in Example 1) our approach cannot be expected to work well if the distributions are chosen adversarially. Roughly speaking, our approach will work best when the following two criteria are satisfied.

1. There is a (relatively low) threshold $t_{critical}$ such that, for all $t > t_{critical}$, the arm that is most likely to yield a payoff $> t$ is the same as the arm most likely to yield a payoff $> t_{critical}$. Call this arm i^*.

2. As t increases beyond $t_{critical}$, there is a growing gap between the probability that arm i^* yields a payoff $> t$ and the corresponding probability for other arms. Specifically, if we let $p_i(t)$ denote the probability that the i^{th} arm returns a payoff $> t$, the ratio $\frac{p_{i^*}(t)}{p_i(t)}$ should increase as a function of t for $t > t_{critical}$, for any $i \neq i^*$.

Figure 1 illustrates a set of two payoff distributions that satisfy these assumptions.

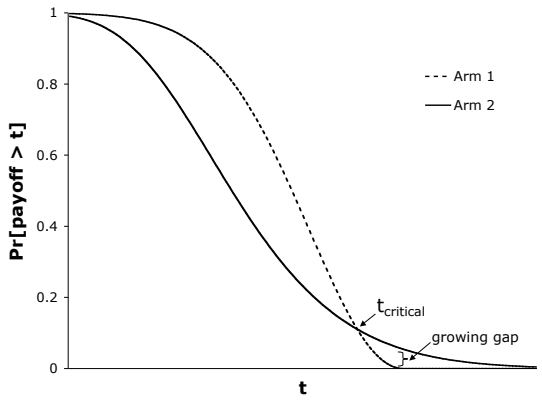

Fig. 1. A max k-armed bandit instance on which Threshold Ascent should perform well.

1.2 Contributions

The primary contributions of this paper are as follows.

1. We present a new algorithm, Chernoff Interval Estimation, for the classical k-armed bandit problem and prove a bound on its regret. Our algorithm is extremely simple and has performance guarantees competitive with the state of the art.
2. Building on Chernoff Interval Estimation, we develop a new algorithm, Threshold Ascent, for solving the max k-armed bandit problem. Our algorithm is designed to work well as long as the two mild distributional assumptions described in §1.1 are satisfied.
3. We evaluate Threshold Ascent experimentally by using it to select among randomized priority dispatching rules for the RCPSP/max. We find that Threshold Ascent (a) performs better than any of the priority rules perform in isolation, and (b) outperforms the recent QD-BEACON max k-armed bandit algorithm of Cicirello and Smith [4, 5].

1.3 Related Work

The classical k-armed bandit problem was first studied by Robbins [11] and has since been the subject of numerous papers; see Berry and Fristedt [3] and Kaelbling [7] for overviews. We give a more detailed discussion of related work on the classical k-armed bandit problem as part of our discussion of Chernoff Interval Estimation in §2.2

The max k-armed bandit problem was introduced by Cicirello and Smith [4, 5], whose experiments with randomized priority dispatching rules for the RCPSP/max form the basis of our experimental evaluation in §4. Cicirello and Smith show that their max k-armed bandit problem yields performance on the RCPSP/max that is competitive with the state of the art. The design of Cicirello and Smith's heuristic is motivated by an analysis of the special case in which each arm's payoff distribution is a GEV distribution with shape parameter $\xi = 0$. Streeter and Smith [13] provide a theoretical treatment of the max k-armed bandit problem under the assumption that each payoff distribution is a GEV.

2 Chernoff Interval Estimation

In this section we present and analyze a simple algorithm, Chernoff Interval Estimation, for the classical k-armed bandit problem. In §3 we use this approach as the basis for Threshold Ascent, an algorithm for the max k-armed bandit problem.

In the classical k-armed bandit problem one is faced with a set of k arms. The i^{th} arm, when pulled, returns a payoff drawn independently at random from a fixed (but unknown) distribution. All payoffs are real numbers between 0 and 1. We denote by μ_i the expected payoff obtained from a single pull of arm i, and define $\mu^* = \max_{1 \leq i \leq k} \mu_i$. We consider the finite-time version of the problem, in which our goal is to maximize the cumulative payoff received using a fixed budget of n pulls. The *regret* of an algorithm (on a particular instance of the classical k-armed bandit problem) is the difference between the cumulative payoff the algorithm would have received by pulling the single best arm n times and the cumulative payoff the algorithm actually received.

Chernoff Interval Estimation is simply the well-known interval estimation algorithm [7, 8] with confidence intervals derived using Chernoff's inequality. Although various interval estimation algorithms have been analyzed in the literature and a variety of guarantees have been proved, both (a) our use of Chernoff's inequality in an interval estimation algorithm and (b) our analysis appear to be novel. In particular, when the mean payoff returned by each arm is small (relative to the maximum possible payoff) our algorithm has much better performance than the recent algorithm of [1], which is identical to our algorithm except that confidence intervals are derived using Hoeffding's inequality. We give further discussion of related work in §2.2.

Procedure **ChernoffIntervalEstimation**$(n,\ \delta)$:

1. Initialize $x_i \leftarrow 0$, $n_i \leftarrow 0\ \forall i \in \{1, 2, \ldots, k\}$.
2. Repeat n times:
   (a) $\hat{i} \leftarrow \arg\max_i U(\bar{\mu}_i, n_i)$, where $\bar{\mu}_i = \frac{x_i}{n_i}$ and

$$U(\mu_0, n_0) = \begin{cases} \mu_0 + \frac{\alpha + \sqrt{2n_0\mu_0\alpha + \alpha^2}}{n_0} & \text{if } n_0 > 0 \\ \infty & \text{otherwise} \end{cases}$$

   where $\alpha = \ln\left(\frac{2nk}{\delta}\right)$.
   (b) Pull arm $\hat{i}$, receive payoff R, set $x_{\hat{i}} \leftarrow x_{\hat{i}} + R$, and set $n_i \leftarrow n_i + 1$.

2.1 Analysis

In this section we put a bound on the expected regret of Chernoff Interval Estimation. Our analysis proceeds as follows. Lemma 1 shows that (with a certain minimum probability) the value $U(\bar{\mu}_i, n_i)$ is always an upper bound on μ_i. Lemma 2 then places a bound on the number of times the algorithm will sample an arm whose mean payoff is suboptimal. Theorem 1 puts these results together to obtain a bound on the algorithm's expected regret.

We will make use of the following inequality.

Chernoff's inequality. *Let $X = \sum_{i=1}^{n} X_i$ be the sum of n independent identically distributed random variables with $X_i \in [0, 1]$ and $\mu = \mathbb{E}[X_i]$. Then for $\beta > 0$,*

$$\mathbb{P}\left[\frac{X}{n} < (1-\beta)\mu\right] < \exp\left(-\frac{n\mu\beta^2}{2}\right)$$

and

$$\mathbb{P}\left[\frac{X}{n} > (1+\beta)\mu\right] < \exp\left(-\frac{n\mu\beta^2}{3}\right) \ .$$

We will also use the following easily-verified algebraic fact.

Fact 1 *If $U = U(\mu_0, n_0)$ then*

$$Un_0\left(1 - \frac{\mu_0}{U}\right)^2 = 2\alpha \ .$$

Lemma 1. *During a run of ChernoffIntervalEstimation(n, δ) it holds with probability at least $1 - \frac{\delta}{2}$ that for all arms $i \in \{1, 2, \ldots, k\}$ and for all n repetitions of the loop, $U(\bar{\mu}_i, n_i) \geq \mu_i$.*

Proof. It suffices to show that for any arm i and any particular repetition of the loop, $\mathbb{P}[U(\bar{\mu}_i, n_i) < \mu_i] < \frac{\delta}{2nk}$. Consider some particular fixed values of μ_i, α, and n_i, and let μ_c be the largest solution to the equation

$$U(\mu_c, n_i) = \mu_i \tag{1}$$

By inspection, $U(\mu_c, n_i)$ is strictly increasing as a function of μ_c. Thus $U(\bar{\mu}_i, n_i) < \mu_i$ if and only if $\bar{\mu}_i < \mu_c$, so $\mathbb{P}[U(\bar{\mu}_i, n_i) < \mu_i] = \mathbb{P}[\bar{\mu}_i < \mu_c]$. Thus

$$\begin{aligned}
\mathbb{P}\left[U(\bar{\mu}_i, n_i) < \mu_i\right] &= \mathbb{P}\left[\bar{\mu}_i < \mu_c\right] \\
&= \mathbb{P}\left[\bar{\mu}_i < \mu_i\left(1 - \left(1 - \frac{\mu_c}{\mu_i}\right)\right)\right] \\
&< \exp\left(-\frac{\mu_i n_i}{2}\left(1 - \frac{\mu_c}{\mu_i}\right)^2\right) \\
&= \exp\left(-\alpha\right) \\
&= \frac{\delta}{2nk}
\end{aligned}$$

where on the third line we have used Chernoff's inequality, and on the fourth line we have used Fact 1 in conjunction with (1). □

Lemma 2. *During a run of ChernoffIntervalEstimation(n, δ) it holds with probability at least $1 - \delta$ that each suboptimal arm i (i.e., each arm i with $\mu_i < \mu^*$) is pulled at most $\frac{3\alpha}{\mu^*}\frac{1}{(1-\sqrt{y_i})^2}$ times, where $y_i = \frac{\mu_i}{\mu^*}$.*

Proof. Let i^* be some optimal arm (i.e., $\mu_{i^*} = \mu^*$) and assume that $U(\bar{\mu}_{i^*}, n_{i^*}) \geq \mu^*$ for all n repetitions of the loop. By Lemma 1, this assumption is valid with probability at least $1 - \frac{\delta}{2}$. Consider some particular suboptimal arm i. By inspection, we will stop sampling arm i once $U(\bar{\mu}_i, n_i) < \mu^*$. So it suffices to show that if $n_i \geq \frac{3\alpha}{\mu^*}\frac{1}{(1-\sqrt{y_i})^2}$, then $U(\bar{\mu}_i, n_i) < \mu^*$ with probability at least $1 - \frac{\delta}{2k}$ (then the probability that any of our assumptions fail is at most $\frac{\delta}{2} + k\frac{\delta}{2k} = \delta$). To show this we will prove two claims.

Claim. If $n_i \geq \frac{3\alpha}{\mu^*}\frac{1}{(1-\sqrt{y_i})^2}$, then with probability at least $1 - \frac{\delta}{2k}$, $\bar{\mu}_i < \sqrt{y_i^{-1}}\mu_i$.

Proof (of Claim 1).

$$\begin{aligned}
\mathbb{P}\left[\bar{\mu}_i > \sqrt{y_i^{-1}}\mu_i\right] &= \mathbb{P}\left[\bar{\mu}_i > \left(1 + \frac{1 - \sqrt{y_i}}{\sqrt{y_i}}\right)\mu_i\right] \\
&< \exp\left(-\frac{n_i\mu_i}{3}\frac{(1-\sqrt{y_i})^2}{y_i}\right) \\
&= \exp\left(-\frac{n_i\mu^*}{3}(1-\sqrt{y_i})^2\right) \\
&\leq \exp\left(-\alpha\right) \\
&= \frac{\delta}{2nk} < \frac{\delta}{2k}\ .
\end{aligned}$$

□

Claim. If $n_i \geq \frac{3\alpha}{\mu^*}\frac{1}{(1-\sqrt{y_i})^2}$ and $\bar{\mu}_i < \sqrt{y_i^{-1}}\mu_i$, then $U(\bar{\mu}_i, n_i) < \mu^*$.

Proof (of Claim 2). Let $U_i = U(\bar{\mu}_i, n_i)$, and suppose for contradiction that $U_i \geq \mu^*$. Then by Fact 1,

$$n_i = \frac{2\alpha}{U_i}\left(1 - \frac{\bar{\mu}_i}{U_i}\right)^{-2} .$$

The right hand side increases as a function of $\bar{\mu}_i$ (assuming $\bar{\mu}_i < U_i$, which is true by definition). So if $\bar{\mu}_i < \sqrt{y_i^{-1}}\mu_i$ then replacing $\bar{\mu}_i$ with $\sqrt{y_i^{-1}}\mu_i$ only increases the value of the right hand side. Similarly, the right hand side decreases as a function of U_i, so if $U_i \geq \mu^*$ then replacing replacing U_i with μ^* only increases the value of the right hand side. Thus

$$n_i < \frac{2\alpha}{\mu^*}\left(1 - \frac{\sqrt{y_i^{-1}}\mu_i}{\mu^*}\right)^{-2} = \frac{2\alpha}{\mu^*}\left(1 - \sqrt{y_i}\right)^{-2}$$

which is a contradiction. □

Putting Claims 1 and 2 together, once $n_i \geq \frac{3\alpha}{\mu^*}\frac{1}{(1-\sqrt{y_i})^2}$ it holds with probability at least $1 - \frac{\delta}{2k}$ that $U(\bar{\mu}_i, n_i) < \mu^*$, so arm i will no longer be pulled. □

The following theorem shows that when n is large (and the parameter δ is small), the total payoff obtained by Chernoff Interval Estimation over n trials is almost as high as what would be obtained by pulling the single best arm for all n trials.

Theorem 1. *The expected regret incurred by ChernoffIntervalEstimation*(n, δ) *is at most*

$$(1-\delta)2\sqrt{3\mu^* n(k-1)\alpha} + \delta\mu^* n$$

where $\alpha = \ln\left(\frac{2nk}{\delta}\right)$.

Proof. We confine our attention to the special case $k = 2$. The proof for general k is similar.

First, note that the conclusion of Lemma 2 fails to hold with probability at most δ. Because expected regret cannot exceed $\mu^* n$, this scenario contributes at most $\delta\mu^* n$ to overall expected regret. Thus it remains to show that, conditioned on the event that the conclusion of Lemma 2 holds, expected regret is at most $2\sqrt{3\mu^* n(k-1)\alpha}$.

Assume $\mu^* = \mu_1 > \mu_2$ and let $y = \frac{\mu_2}{\mu^*}$. By Lemma 2, we sample arm 2 at most $\min\{n, \frac{3\alpha}{\mu^*}\frac{1}{(1-\sqrt{y})^2}\}$ times. Each sample of arm 2 incurs expected regret $\mu^* - \mu_2 = \mu^*(1-y)$. Thus expected total regret is at most

$$\mu^*(1-y)\min\left\{n, \frac{3\alpha}{\mu^*}\frac{1}{(1-\sqrt{y})^2}\right\} . \tag{2}$$

Using the fact that $y < 1$,

$$\begin{aligned}\frac{1-y}{(1-\sqrt{y})^2} &= \frac{1-y}{(1-\sqrt{y})^2} \cdot \frac{(1+\sqrt{y})^2}{(1+\sqrt{y})^2} \\ &= \frac{(1+\sqrt{y})^2}{1-y} \\ &< \frac{4}{1-y} \ .\end{aligned}$$

Plugging this value into (2), the expected total regret is at most

$$\min\left\{\mu^* \Delta n, \frac{12\alpha}{\Delta}\right\}$$

where $\Delta = 1 - y$. Setting these two expressions equal gives $\bar{\Delta} = 2\sqrt{\frac{3\alpha}{n\mu^*}}$ as the value of Δ that maximizes expected regret. Thus the expected regret is at most $\mu^* \bar{\Delta} n = 2\sqrt{3\mu^* n\alpha} = 2\sqrt{3\mu^* n(k-1)\alpha}$.

□

2.2 Discussion and Related Work

Types of Regret Bounds In comparing the regret bound of Theorem 1 to previous work, we must distinguish between two different types of regret bounds. The first type of bound describes the asymptotic behavior of regret (as $n \to \infty$) on a *fixed* problem instance (i.e., with all k payoff distributions held constant). In this framework, a lower bound of $\Omega(\ln(n))$ has been proved, and algorithms exist that achieve regret $O(\ln(n))$ [1]. Though we do not prove it here, Chernoff Interval Estimation achieves $O(\ln(n))$ regret in this framework when δ is set appropriately.

The second type of bound concerns the maximum, over all possible instances, of the expected regret incurred by the algorithm when run on that instance for n pulls. In this setting, a lower bound of $\Omega(\sqrt{kn})$ has been proved [2]. It is this second form of bound that Theorem 1 provides. In what follows, we will only consider bounds of this second form.

The Classical *k*-Armed Bandit Problem We are not aware of any work on the classical k-armed bandit problem that offers a better regret bound (of the second form) than the one proved in Theorem 1. Auer et al. [1] analyze an algorithm that is identical to ours except that the confidence intervals are derived from Hoeffding's inequality rather than Chernoff's inequality. An analysis analogous to the one in this paper shows that their algorithm has worst-case regret $O(\sqrt{nk\ln(n)})$ when the instance is chosen adversarially as a function of n. Plugging $\delta = \frac{1}{n^2}$ into Theorem 1 gives a bound of $O(\sqrt{n\mu^* k\ln(n)})$, which is never any worse than the latter bound (because $\mu^* \leq 1$) and is much better when μ^* is small.

The Nonstochastic Multiarmed Bandit Problem In a different paper, Auer et al. [2] consider a variant of the classical k-armed bandit problem in which the sequence of payoffs returned by each arm is determined adversarially in advance. For this more difficult problem, they present an algorithm called **Exp3.1** with expected regret

$$8\sqrt{(e-1)G_{\max}k\ln(k)} + 8(e-1)k + 2k\ln(k)$$

where $G_{\max}$ is the maximum, over all k arms, of the total payoff that would be obtained by pulling that arm for all n trials. If we plug in $G_{\max} = \mu^* n$, this bound is sometimes better than the one given by Theorem 1 and sometimes not, depending on the values of n, k, and μ^*, as well as the choice of the parameter δ.

3 Threshold Ascent

To solve the max k-armed bandit problem, we use Chernoff Interval Estimation to maximize the number of payoffs that exceed a threshold T that varies over time. Initially, we set T to zero. Whenever s or more payoffs $> T$ have been received so far, we increment T. We refer to the resulting algorithm as Threshold Ascent. The code for Threshold Ascent is given below. For simplicity, we assume that all payoffs are integer multiples of some known constant Δ.

Procedure **ThresholdAscent**(s, n, δ):

1. Initialize $T \leftarrow 0$ and $n_i^R = 0$, $\forall i \in \{1, 2, \ldots, k\}, R \in \{0, \Delta, 2\Delta, \ldots, 1-\Delta, 1\}$.
2. Repeat n times:
   (a) While $\sum_{i=1}^{k} S_i(T) \geq s$ do:

$$T \leftarrow T + \Delta$$

   where $S_i(t) = \sum_{R>t} n_i^R$ is the number of payoffs $> t$ received so far from arm i.
   (b) $\hat{i} \leftarrow \arg\max_i U\left(\frac{S_i(T)}{n_i}, n_i\right)$, where $n_i = \sum_R n_i^R$ is the number of times arm i has been pulled and

$$U(\mu_0, n_0) = \begin{cases} \mu_0 + \frac{\alpha + \sqrt{2n_0\mu_0\alpha + \alpha^2}}{n_0} & \text{if } n_0 > 0 \\ \infty & \text{otherwise} \end{cases}$$

   where $\alpha = \ln\left(\frac{2nk}{\delta}\right)$.
   (c) Pull arm $\hat{i}$, receive payoff R, and set $n_i^R \leftarrow n_i^R + 1$.

The parameter s controls the tradeoff between exploration and exploitation. To understand this tradeoff, it is helpful to consider two extreme cases.

Case $s = 1$. ThresholdAscent$(1, n, \delta)$ is equivalent to round-robin sampling. When $s = 1$, the threshold T is incremented whenever a payoff $> T$ is obtained. Thus the value $\frac{S_i(T)}{n_i}$ calculated in 2 (b) is always 0, so the value of $U\left(\frac{S_i(T)}{n_i}, n_i\right)$ is determined strictly by n_i. Because U is a decreasing function of n_i, the algorithm simply samples whatever arm has been sampled the smallest number of times so far.

Case $s = \infty$. ThresholdAscent(∞, n, δ) is equivalent to ChernoffIntervalEstimation (n, δ) running on a k-armed bandit instance where payoffs $> T$ are mapped to 1 and payoffs $\leq T$ are mapped to 0.

4 Evaluation on the RCPSP/max

Following Cicirello and Smith [4, 5], we evaluate our algorithm for the max k-armed bandit problem by using it to select among randomized priority dispatching rules for the resource-constrained project scheduling problem with maximal time lags (RCPSP/max). Cicirello and Smith's work showed that a max k-armed bandit approach yields good performance on benchmark instances of this problem.

Briefly, in the RCPSP/max one must assign start times to each of a number of activities in such a way that certain temporal and resource constraints are satisfied. Such an assignment of start times is called a *feasible schedule*. The goal is to find a feasible schedule whose makespan is as small as possible, where makespan is defined as the maximum completion time of any activity.

Even without maximal time lags (which make the problem more difficult), the RCPSP is NP-hard and is "one of the most intractable problems in operations research" [9]. When maximal time lags are included, the feasibility problem (i.e., deciding whether a feasible schedule exists) is also NP-hard.

4.1 The RCPSP/max

Formally, an instance of the RCPSP/max is a tuple $\mathcal{I} = (\mathcal{A}, R, \mathcal{T})$, where $\mathcal{A}$ is a set of activities, R is a vector of resource capacities, and $\mathcal{T}$ is a list of temporal constraints. Each activity $a_i \in \mathcal{A}$ has a *processing time* p_i, and a resource demand $r_{i,k}$ for each $k \in \{1, 2, \ldots, |R|\}$. Each temporal constraint $T \in \mathcal{T}$ is a triple $T = (i, j, \delta)$, where i and j are activity indices and δ is an integer. The constraint $T = (i, j, \delta)$ indicates that activity a_j cannot start until δ time units after activity a_i has started.

A schedule S assigns a *start time* $S(a)$ to each activity $a \in \mathcal{A}$. S is feasible if

$$S(a_j) - S(a_i) \geq \delta \quad \forall (i, j, \delta) \in \mathcal{T}$$

(i.e., all temporal constraints are satisfied), and

$$\sum_{a_i \in A(S,t)} r_{i,k} \leq R_k \quad \forall\, t \geq 0, k \in \{1, 2, \ldots, |R|\}$$

where $A(S,t) = \{a_i \in \mathcal{A} \mid S(a_i) \leq t < S(a_i) + p_i\}$ the set of activities that are in progress at time t. The latter equation ensures that no resource capacity is ever exceeded.

4.2 Randomized Priority Dispatching Rules

A priority dispatching rule for the RCPSP/max is a procedure that assigns start times to activities one at a time, in a greedy fashion. The order in which start times are assigned is determined by a rule that assigns priorities to each activity. As noted above, it is NP-hard to generate a feasible schedule for the RCPSP/max. Priority dispatching rules are therefore augmented to perform a limited amount of backtracking in order to increase the odds of producing a feasible schedule. For more details, see [10].

Cicirello and Smith describe experiments with randomized priority dispatching rules, in which the next activity to schedule is chosen from a probability distribution, with the probability assigned to an activity being proportional to its priority. Cicirello and Smith consider the five randomized priority dispatching rules in the set $\mathcal{H} = \{LPF, LST, MST, MTS, RSM\}$. See Cicirello and Smith [4, 5] for a complete description of these heuristics. We use the same five heuristics as Cicirello and Smith, with two modifications: (1) we added a form of intelligent backtracking to the procedure of [10] in order to increase the odds of generating a feasible schedule and (2) we modified the RSM heuristic to improve its performance.

4.3 Instances

We evaluate our approach on a set $\mathcal{I}$ of 169 RCPSP/max instances from the ProGen/max library [12]. These instances were selected as follows. We first ran the heuristic LPF (the heuristic identified by Cicirello and Smith as having the best performance) 10,000 times on all 540 instances from the `TESTSETC` data set. For many of these instances, LPF found a (provably) optimal schedule on a large proportion of the runs. We considered any instance in which the best makespan found by LPF was found with frequency > 0.01 to be "easy" and discarded it from the data set. What remained was a set $\mathcal{I}$ of 169 "hard" RCPSP/max instances.

For each RCPSP/max instance $I \in \mathcal{I}$, we ran each heuristic $h \in \mathcal{H}$ 10,000 times, storing the results in a file. Using this data, we created a set $\mathcal{K}$ of 169 five-armed bandit problems (each of the five heuristics $h \in \mathcal{H}$ represents an arm). After the data were collected, makespans were converted to payoffs by multiplying each makespan by -1 and scaling them to lie in the interval $[0, 1]$.

4.4 Payoff Distributions in the RCPSP/max

To motivate the use of a distribution-free approach to the max k-armed bandit problem, we examine the payoff distributions generated by randomized priority

dispatching rules for the RCPSP/max. For a number of instances $I \in \mathcal{I}$, we plotted the payoff distribution functions for each heuristic $h \in \mathcal{H}$. For each distribution, we fitted a GEV to the empirical data using maximum likelihood estimation of the parameters μ, σ, and ξ, as recommended by Coles [6].

Our experience was that the GEV sometimes provides a good fit to the empirical cumulative distribution function but sometimes provides a very poor fit. Figure 2 shows the empirical distribution and the GEV fit to the payoff distribution of LPF on instances `PSP129` and `PSP121`. For the instance `PSP129`, the GEV accurately models the entire distribution, including the right tail. For the instance `PSP121`, however, the GEV fit severely overestimates the probability mass in the right tail. Indeed, the distribution in Figure 2 (B) is so erratic that no parametric family of distributions can be expected to be a good model of its behavior. In such cases a distribution-free approach is preferable.

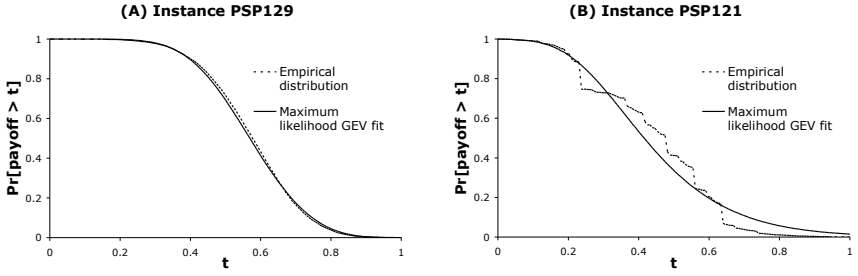

Fig. 2. Empirical cumulative distribution function of the LPF heuristic for two RCPSP/max instances. (A) depicts an instance for which the GEV provides a good fit; (B) depicts an instance for which the GEV provides a poor fit.

4.5 An Illustrative Run

Before presenting our results, we illustrate the typical behavior of Threshold Ascent by showing how it performs on the instance `PSP124`. For this and all subsequent experiments, we run Threshold Ascent with parameters $n = 10,000$, $s = 100$, and $\delta = 0.01$.

Figure 3 (A) depicts the payoff distributions for each of the five arms. As can be seen, LPF has the best performance on `PSP124`. MST has zero probability of generating a payoff > 0.8, while LST and RMS have zero probability of generating a payoff > 0.9. MTS gives competitive performance up to a payoff of $t \approx 0.9$, after which point the probability of obtaining a payoff $> t$ suddenly drops to zero.

Figure 3 (B) shows the number of pulls allocated by Threshold Ascent to each of the five arms as a function of the number of pulls performed so far. As

can be seen, Threshold Ascent is a somewhat conservative strategy, allocating a fair number of pulls to heuristics that might seem "obviously" suboptimal to a human observer. Nevertheless, Threshold Ascent spends the majority of its time sampling the single best heuristic (LPF).

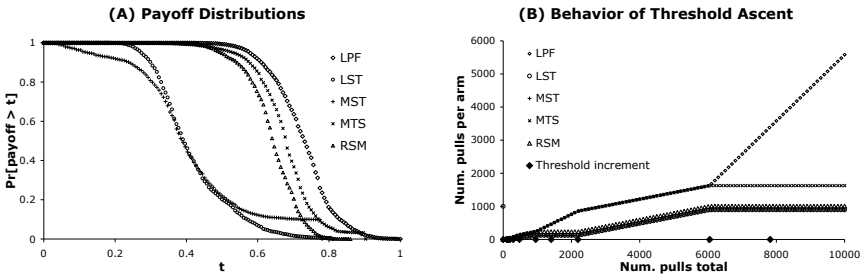

Fig. 3. Behavior of Threshold Ascent on instance `PSP124`. (A) shows the payoff distributions; (B) shows the number of pulls allocated to each arm.

4.6 Results

For each instance $K \in \mathcal{K}$, we ran three max k-armed bandit algorithms, each with a budget of $n = 10,000$ pulls: Threshold Ascent with parameters $n = 10,000$, $s = 100$, and $\delta = 0.01$, the QD-BEACON algorithm of Cicirello and Smith [5], and an algorithm that simply sampled the arms in a round-robin fashion. Cicirello and Smith describe three versions of QD-BEACON; we use the one based on the GEV distribution. For each instance $K \in \mathcal{K}$, we define the *regret* of an algorithm as the difference between the minimum makespan (which corresponds to the maximum payoff) sampled by the algorithm and the minimum makespan sampled by any of the five heuristics (on any of the $10,000$ stored runs of each of the five heuristics). For each of the three algorithms, we also recorded the number of instances for which the algorithm generated a feasible schedule. Table 1 summarizes the performance of these three algorithms, as well as the performance of each of the five heuristics in isolation.

Of the eight max k-armed bandit strategies we evaluated (Threshold Ascent, QD-BEACON, round-robin sampling, and the five pure strategies), Threshold Ascent has the least regret and achieves zero regret on the largest number of instances. Additionally, Threshold Ascent generated a feasible schedule for the 166 (out of 169) instances for which any of the five heuristics was able to generate a feasible schedule (for three instances, none of the five randomized priority rules generated a feasible schedule after 10,000 runs).

Table 1. Performance of eight heuristics on 169 RCPSP/max instances.

Heuristic	Σ **Regret**	$\mathbb{P}[\text{Regret} = 0]$	**Num. Feasible**
Threshold Ascent	188	0.722	166
Round-robin sampling	345	0.556	166
LPF	355	0.675	164
MTS	402	0.657	166
QD-BEACON	609	0.538	165
RSM	2130	0.166	155
LST	3199	0.095	164
MST	4509	0.107	164

4.7 Discussion

Two of the findings summarized in Table 1 may seem counterintuitive: the fact that round-robin performs better than any single heuristic, and the fact that QD-BEACON performs worse than round-robin. We now examine each of these findings in more detail.

Why Round-Robin Sampling Performs Well In the classical k-armed bandit problem, round-robin sampling can never outperform the best pure strategy (where a pure strategy is one that samples the same arm the entire time), either on a single instance or across multiple instances. In the max k-armed bandit problem, however, the situation is different, as the following example illustrates.

Example 2. Suppose we have 2 heuristics, and we run them each for n trials on a set of I instances. On half the instances, heuristic A returns payoff 0 with probability 0.9 and returns payoff 1 with probability 0.1, while heuristic B returns payoff 0 with probability 1. On the other half of the instances, the roles of heuristics A and B are reversed.

If n is large, round-robin sampling will yield total regret ≈ 0, while either of the two heuristics will have regret $\approx \frac{1}{2}I$. By allocating pulls equally to each arm, round-robin sampling is guaranteed to sample the best heuristic at least $\frac{n}{k}$ times, and if n is large this number of samples may be enough to exploit the tail behavior of the best heuristic.

Understanding QD-BEACON QD-BEACON is designed to converge to a single arm at a doubly-exponential rate. That is, the number of pulls allocated to the (presumed) optimal arm increases doubly-exponentially relative to the number of pulls allocated to presumed suboptimal arms. In our experience, QD-BEACON usually converges to a single arm after at most 10-20 pulls from each arm. This rapid convergence can lead to large regret if the presumed best arm is actually suboptimal.

5 Conclusions

We presented an algorithm, Chernoff Interval Estimation, for solving the classical k-armed bandit problem, and proved that it has good performance guarantees when the mean payoff returned by each arm is small relative to the maximum possible payoff. Building on Chernoff Interval Estimation we presented an algorithm, Threshold Ascent, that solves the max k-armed bandit problem without making strong assumptions about the payoff distributions. We demonstrated the effectiveness of Threshold Ascent on the problem of selecting among randomized priority dispatching rules for the RCPSP/max.

Acknowledgment. This work was sponsored in part by the National Science Foundation under contract #9900298 and the CMU Robotics Institute.

References

1. Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002a.
2. Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002b.
3. Donald. A. Berry and Bert Fristedt. *Bandit Problems: Sequential Allocation of Experiments*. Chapman and Hall, London, 1986.
4. Vincent A. Cicirello and Stephen F. Smith. Heuristic selection for stochastic search optimization: Modeling solution quality by extreme value theory. In *Proceedings of the 10th International Conference on Principles and Practice of Constraint Programming*, pages 197–211, 2004.
5. Vincent A. Cicirello and Stephen F. Smith. The max k-armed bandit: A new model of exploration applied to search heuristic selection. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, pages 1355–1361, 2005.
6. Stuart Coles. *An Introduction to Statistical Modeling of Extreme Values*. Springer-Verlag, London, 2001.
7. Leslie P. Kaelbling. *Learning in Embedded Systems*. The MIT Press, Cambridge, MA, 1993.
8. Tze Leung Lai. Adaptive treatment allocation and the multi-armed bandit problem. *The Annals of Statistics*, 15(3):1091–1114, 1987.
9. Rolf H. Möhring, Andreas S. Schulz, Frederik Stork, and Marc Uetz. Solving project scheduling problems by minimum cut computations. *Management Science*, 49(3):330–350, 2003.
10. Klaus Neumann, Christoph Schwindt, and Jürgen Zimmerman. *Project Scheduling with Time Windows and Scarce Resources*. Springer-Verlag, 2002.
11. Herbert Robbins. Some aspects of sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535, 1952.
12. C. Schwindt. Generation of resource–constrained project scheduling problems with minimal and maximal time lags. Technical Report WIOR-489, Universität Karlsruhe, 1996.
13. Matthew J. Streeter and Stephen F. Smith. An asymptotically optimal algorithm for the max k-armed bandit problem. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 135–142, 2006.