

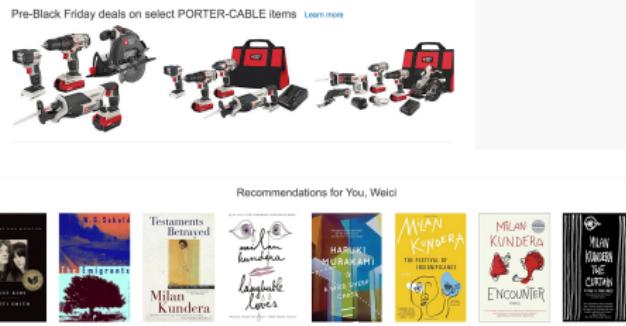
# Sequential Resource Allocation Under Uncertainty: An Index Policy Approach

Weici Hu  
Adviser: Peter Frazier

Cornell ORIE

July 13, 2017

# RMAB problems have many applications in industry



Personalized advertisement on Amazon



Project management

# Problem Setup

We consider an MDP  $(\mathbb{S}^K, \mathbb{A}^K, \mathbb{P}^{\cdot}, R)$  that consists of  $K$  identical sub-processes  $(\mathbb{S}, \mathbb{A}, P^{\cdot}, r)$ , specifically,

- Time horizon  $T < \infty$ .
- State space  $\mathbb{S}^K$  is the cross-product of  $K$   $\mathbb{S}$ .  $\mathbb{S}$  is assumed finite.
- Action space  $\mathbb{A}^K$  is the cross-product of  $K$   $\mathbb{A}$ .  $\mathbb{A} = \{0, 1\}$ .
- Reward  $R_t(\mathbf{s}, \mathbf{a}) = \sum_{x=1}^K r_t(s_x, a_x)$ ,  $1 \leq t \leq T$ , is additive of the reward of individual sub-processes.
- Transition probability  $\mathbb{P}^{\mathbf{a}}(\mathbf{s}', \mathbf{s}) = \prod_{x=1}^K P^{a_x}(s'_x, s_x)$ .
- Starts at an initial state  $\mathbf{s}_1$ .

# Problem Setup Con't

- A Markov policy  $\pi : \mathbb{S}^K \times \mathbb{A}^K \times \{1, \dots, T\} \rightarrow [0, 1]$ , with  $\pi(\mathbf{s}, \mathbf{a}, t) = P(\mathbf{a} | \mathbf{S}_t = \mathbf{s})$  (Our decision). We require  $\sum_{\mathbf{a} \in \mathbb{A}^K} \pi(\mathbf{s}, \mathbf{a}, t) = 1$ .  $\forall \mathbf{s} \in \mathbb{S}^K, \forall 1 \leq t \leq T$ .
- Objective

$$\begin{aligned} & \underset{\pi \in \Pi}{\text{maximize}} \quad \mathbb{E}^{\pi} \left[ \sum_{t=1}^T R_t(\mathbf{S}_t, \mathbf{A}_t) \right] \\ & \text{subject to} \quad P^{\pi}(|\mathbf{A}_t| = m_t) = 1, \quad \forall 1 \leq t \leq T. \end{aligned} \tag{1}$$

## Difficulty: Optimal solutions are computationally infeasible

Optimal solutions of (1) can be obtained with Bellman optimality equations.

But it requires  $O(|\mathbb{S}|^K |\mathbb{A}|^K T)$  time complexity and  $O(|\mathbb{S}|^K |\mathbb{A}|^K T)$  storage complexity.

The complexity grows exponentially with the number of sub-processes  $K$ , and becomes computationally infeasible for large  $K$ .

## Past attempts

- Gittins: Proposes Gittin's index policy that is optimal for infinite horizon MAB with single pull per period.
- Whittle: Proposes Whittle's policy that is derived from a Lagrangian relaxation of the RMAB, but only for the infinite time horizon case.  
conjectures that Whittle's index policy is asymptotically optimal when both the number of arms and the constraints go to infinity, but this is only true under restricted condition.  
(Weiss and Weber)
- Adelman: provides conditions when the gap between the Lagrangian relaxation and the optimal value is bounded by a constant, but does not offer a feasible policy.

# Pre-computations: 1. Optimal Lagrange Multiplier of (2)

Relax the original problem (1) to

$$\begin{aligned} & \underset{\pi \in \Pi}{\text{maximize}} \quad \mathbb{E}^{\pi} \left[ \sum_{t=1}^T R_t(\mathbf{S}_t, \mathbf{A}_t) \right] \\ & \text{subject to} \quad \mathbb{E}^{\pi}(|\mathbf{A}_t|) = m_t, \quad \forall 1 \leq t \leq T. \end{aligned} \tag{2}$$

The Lagrangian relaxation of (2)

$$P(\lambda) = \max_{\pi \in \Pi} \mathbb{E}^{\pi} \left[ \sum_{t=1}^T R_t(\mathbf{S}_t, \mathbf{A}_t) \right] - \sum_{t=1}^T \lambda_t (\mathbb{E}^{\pi}(|\mathbf{A}_t|) - m_t). \tag{3}$$

Decomposition of the Lagrangian relaxation

$$P(\lambda) = KQ(\lambda) + \sum_t \lambda_t m_t, \quad (4)$$

where

$$Q(\lambda) = \max_{\pi \in \Pi} \mathbb{E}^{\pi} \left[ \sum_{t=1}^T r_t(S_t, A_t) - \lambda_t A_t \right], \quad (5)$$

is the objective function for sub-process  $(\mathbb{S}, \mathbb{A}, P^\cdot, r)$ . Definition of policy  $\pi$  is similar to  $\pi$ , with  $\pi(s, a, t) = P(a|S_t = s)$ .

## Pre-computations: 1. Optimal Lagrange Multiplier of (2)

Optimal Lagrange Multiplier  $\lambda^*$  is a solution to the Lagrange dual

$$\min_{\lambda} P(\lambda) = K \left( \min_{\lambda} Q(\lambda) + \sum_t \lambda_t \frac{m_t}{K} \right), \quad (6)$$

which can be solved by the following linear program (LP):

$$\begin{aligned} & \min_{\{V(s,t), \lambda_t : s \in \mathbb{S}, t \in \{1, \dots, T\}\}} V(s_1, 1) + \frac{1}{K} \sum_t \lambda_t m_t \\ \text{subject to} \quad & V(s, t) - \sum_{s' \in \mathbb{S}} P^a(s, s') V(s', t+1) \geq r_t(s, a) - \lambda_t a, \\ & \forall s \in \mathbb{S}, a \in \mathbb{A}, 1 \leq t \leq T-1 \\ & V(s, T) \geq r_T(s, a) - \lambda_1 a \quad \forall s \in \mathbb{S}, a \in \mathbb{A} \end{aligned} \quad (7)$$

$$V^*(s_1, 1) = Q(\lambda^*).$$

Remarks:

- Problem (7) has  $O(|\mathbb{S}|T)$  variables and  $O(|\mathbb{S}||\mathbb{A}|)T$  constraints, which is manageable.
- $P(\lambda)$  is a point-wise maximum of a group of affine functions of  $\lambda$ , hence convex in  $\lambda$ . (6) can also be solved by sub-gradient descent.

## Pre-computations: 2. Occupation Measure $\rho^*$ of an optimal policy of $Q(\lambda^*)$

**Occupation measure** of a policy  $\pi$ : the fraction of the time a process spent in each state-action pair at a time step under  $\pi$ .

$$\rho(s, a, t) = \pi(s, a, t)P(S_t = s).$$

To compute  $\rho^*$ , we solve the following linear program (LP):

$$\max_{\rho} \quad \sum_{t=1}^T \sum_{a \in \mathbb{A}} \sum_{s \in \mathbb{S}} \rho(s, a, t) r_t(s, a)$$

$$\text{subject to} \quad \sum_{s \in \mathbb{S}} \rho(s, 1, t) = \frac{m_t}{K}, \forall t = 1, \dots, T$$

$$\sum_{a \in \mathbb{A}} \rho(s, a, t) - \sum_{a \in \mathbb{A}} \sum_{s' \in \mathbb{S}} \rho(s', a, t-1) P^a(s', s) = 0, \forall s \in \mathbb{S}, 2 \leq t \leq T$$

$$\sum_{a \in \mathbb{A}} \rho(s, a, 1) = \mathbb{1}(s = s_1) \quad \forall s \in \mathbb{S}$$

$$\rho(s, a, t) \geq 0, \quad \forall s \in \mathbb{S}, a \in \mathbb{A}, t = 1, \dots, T.$$

## Pre-computations: 2. Occupation Measure $\rho^*$ of an optimal policy of $Q(\lambda^*)$

### Lemma

If we construct a policy  $\pi^*$  by

$$\pi^*(s, a, t) = \begin{cases} \frac{\rho^*(s, a, t)}{\sum_{a \in \mathbb{A}} \rho^*(s, a, t)}, & \text{if } \sum_{a \in \mathbb{A}} \rho^*(s, a, t) > 0 \\ \text{arbitrary,} & \text{if } \sum_{a \in \mathbb{A}} \rho^*(s, a, t) = 0 \end{cases} \quad (9)$$

for all  $s \in \mathbb{S}, a \in \mathbb{A}, 1 \leq t \leq T$ , then  $\pi^*$  is an optimal policy for  $Q(\lambda^*)$ .

Quick justification:

- $\pi^*$  meets the definition of a policy.
- $\mathbb{E}^{\pi^*}[\sum_{t=1}^T r_t(S_t, A_t)] = Q(\lambda^*) + \sum_{t=1}^T \lambda_t^* \frac{m_t}{K}$  by strong duality.  
Hence  $\mathbb{E}^{\pi^*}[\sum_{t=1}^T r_t(S_t, A_t) - \lambda^* A_t] = Q(\lambda^*)$ .

## Pre-computations: 2. Occupation Measure $\rho^*$ of an optimal policy of $Q(\lambda^*)$

Remark:

- Any policy  $\pi^*$  constructed using  $\rho^*$  is an optimal solution to  $Q(\lambda^*)$  and satisfies

$$\mathbb{E}^{\pi^*}[A_t] = \frac{m_t}{K}. \quad (10)$$

- Solving for  $\rho^*$  requires solving an LP with  $|\mathbb{S}||\mathbb{A}|T$  variables and at most  $T|\mathbb{S}|$  constraints.

## Pre-computations: 3. Indices of states

We first describe a specific way of computing an optimal policy  $\pi^\lambda$  of  $Q(\lambda)$ :

Define value functions  $V^\lambda : \mathbb{S} \times \{1, \dots, T\} \mapsto \mathbb{R}$  of  $Q(\lambda)$  recursively by

$$V^\lambda(s, t) = \begin{cases} \max_{a \in \mathbb{A}} \{r_T(s, a) - a\lambda_T\} & \text{if } t = T, \\ \max_{a \in \mathbb{A}} \{r_t(s, a) - a\lambda_t + \sum_{s' \in \mathbb{S}} P^a(s, s') V^\lambda(s', t+1)\} & \text{o.w.} \end{cases} \quad (11)$$

Construct  $\pi^\lambda$  by

$$\pi^\lambda(s, 1, t) = \begin{cases} 1 & \text{if one-step lookahead value for } a = 1 \text{ is} \\ & \text{greater than or equal to } a = 0 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

## Pre-computations: 3. Indices of states

Use  $\mathbf{v}[c, t]$  to denote  $\mathbf{v} + (c - v_t) * \mathbf{e}_t$ .

The index of a state  $s \in \mathbb{S}$  at time  $t$  is defined by

$$\beta_t(s) = \sup\{\beta : \pi^{\lambda^*[s,t]}(s, 1, t) = 1\} \quad (13)$$

We compute  $\beta = \{\beta_t(s) : s \in \mathbb{S}, 1 \leq t \leq T\}$

Remark: Computing  $\pi^\lambda$  takes  $O(|\mathbb{S}||\mathbb{A}|T)$  time and space.  
Computational complexity for  $\beta$  is  $O(|\mathbb{S}|^2|\mathbb{A}|T^2)$

# Algorithm of Index policy $\hat{\pi}$

Pre-compute:  $\lambda^*$ ;  $\beta$ ;  $\rho$ . (Refer to earlier discussions for computation details)

**for**  $t = 1, \dots, T$  **do**

    Let  $\beta_{t,[i]}$  be the  $i^{th}$  largest element in the list  $\beta_t(\mathbf{S}_{t,1}), \dots, \beta_t(\mathbf{S}_{t,K})$ , so  $\beta_{t,[1]} \geq \dots \geq \beta_{t,[K]}$ .

    Let  $\bar{\beta}_t = \beta_{t,[m_t]}$

    Let  $I_t = \{s : \beta_t(s) = \bar{\beta}_t \text{ and } s = \mathbf{S}_{t,x} \text{ for some } x\}$

    Let  $N_t(s) = |\{x : \mathbf{S}_{t,x} = s\}|$ , for all  $s$ .

    For  $s \in I_t$ , let

$$q(s) = \begin{cases} \frac{\rho^*(s, 1, t)}{\sum_{s' \in I_t} \rho^*(s', 1, t)}, & \text{if } \sum_{s' \in I_t} \rho^*(s', 1, t) > 0 \\ \frac{N_t(s)}{\sum_{s' \in I_t} N_t(s')}, & \text{otherwise} \end{cases}$$

    Let  $b = \text{Rounding}(m_t - \sum_{s' : \beta_t(s') > \bar{\beta}_t} N_t(s'), (q(s) : s \in I_t), (N_t(s) : s \in I_t))$

**for** all  $s$  **do**

        If  $\beta_t(s) > \bar{\beta}_t$ , set all  $N_t(s)$  sub-processes in  $s$  active.

        If  $\beta_t(s) = \bar{\beta}_t$ , set  $b(s)$  sub-processes in  $s$  active.

        If  $\beta_t(s) < \bar{\beta}_t$ , set 0 sub-processes in  $s$  active.

**end for**

**end for**

Remark:

- *Rounding* takes care of all the corner cases.
- Computational complexity of  $\hat{\pi}$  (excluding pre-computations) is  $O(TK\log(K))$ .

# Asymptotic optimality of the index policy $\hat{\pi}$

Notations:

- For  $\alpha \in \mathbb{R}^T$ , and  $c \in \mathbb{R}$ , define  $\lfloor \alpha c \rfloor = (\lfloor \alpha_1 c \rfloor, \dots, \lfloor \alpha_T c \rfloor)$ .
- Let  $Z(\pi, \mathbf{m}, K)$  denote the expected reward of the original MDP (1), with constraint  $\mathbf{m} = (m_1, \dots, m_T)$  and  $K$  sub-processes.
- Let  $\Pi_{\mathbf{m}, K}$  denote the set of all feasible Markov policies of the original MDP (1), with constraint  $\mathbf{m} = (m_1, \dots, m_T)$  and  $K$  sub-processes.

## Theorem (1)

For any  $\alpha \in [0, 1]^T$ ,

$$\lim_{K \rightarrow \infty} \frac{1}{K} \left( \max_{\pi \in \Pi_{\lfloor \alpha K \rfloor, K}} Z(\pi, \lfloor \alpha K \rfloor, K) - Z(\hat{\pi}, \lfloor \alpha K \rfloor, K) \right) = 0. \quad (14)$$

# Asymptotic optimality of the index policy $\hat{\pi}$

Notations:

- Define  $N_t(s)$  as the number of sub-processes in state  $s$  at time  $t$ , and  $M_t(s)$  the number of sub-processes taking active actions in state  $s$  at time  $t$ , both under  $\hat{\pi}$ .
- Let  $\pi^*$  be a policy constructed using  $\rho^*$ .
- Let  $P_t(s)$  denote the probability of a sub-process being in state  $s$  at time  $t$  under  $\pi^*$ .

## Theorem (2)

For every  $s \in \mathbb{S}$  and  $1 \leq t \leq T$ ,

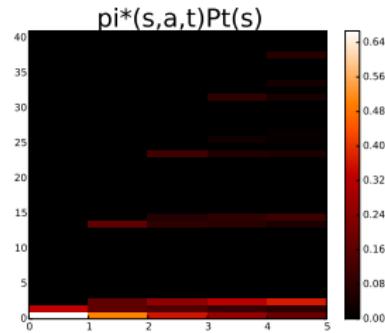
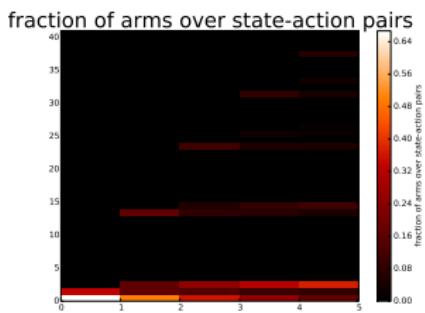
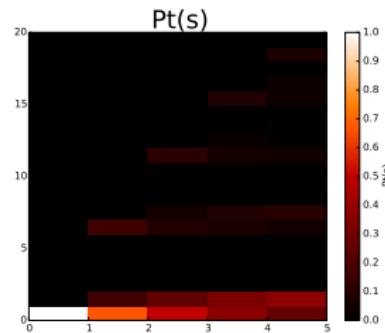
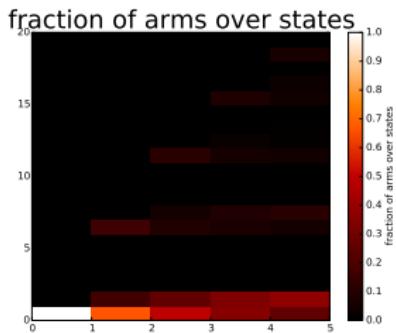
$$\lim_{K \rightarrow \infty} \frac{N_t(s)}{K} = P_t(s), \quad P^{\hat{\pi}} - \text{a.s.}, \quad (15)$$

and

$$\lim_{K \rightarrow \infty} \frac{M_t(s)}{K} = P_t(s) * \pi^*(s, 1, t), \quad P^{\hat{\pi}} - \text{a.s.}, \quad (16)$$

# Theorem 2 is proven by induction over time

An numerical illustration of Theorem 2 using an instance of MAB with  $K = 10000$



## Theorem 2 is proven by induction over time

Proof Intuition of Theorem 2 (Warning: lots of hand-waves):

Assume Theorem 2 holds for time  $t - 1$ .

- Proof of the first result holds at  $t$ :

Same distribution of arms over state, same fraction of arms get to set active in each state  $\Rightarrow$  same distribution of arms over state in the next time period.

- Proof of the second result holds at  $t$ :

- 1.  $\{s \in \mathbb{S} : \beta_t(s) > \lambda_t^*\} \approx \{s \in \mathbb{S} : \pi^* \text{ chooses } a = 1 \text{ in } s\}$ .  
2.  $\pi^*$  chooses  $a = 1$   $\alpha_t$  of the time.
- $1 + 2 \Rightarrow \sum_{s \in \{s \in \mathbb{S} : \beta_t(s) > \lambda_t^*\}} P_t(s) \approx \alpha_t$   
 $\Rightarrow \sum_{s \in \{s \in \mathbb{S} : \beta_t(s) > \lambda_t^*\}} \frac{N_t(s)}{K} \approx \alpha_t$ .
- The index policy  $\hat{\pi}$  sets  $\alpha_t K$  arms active, hence

# Proof of Theorem 1

That  $\hat{\pi}$  being feasible implies

$$Z(\hat{\pi}, \lfloor \alpha K \rfloor, K) \leq \max_{\pi \in \Pi_{\lfloor \alpha K \rfloor, K}} Z(\pi, \lfloor \alpha K \rfloor, K)$$

. Thus,

$$\lim_{K \rightarrow \infty} \frac{1}{K} Z(\hat{\pi}, \lfloor \alpha K \rfloor, K) \leq \lim_{K \rightarrow \infty} \frac{1}{K} \sup_{\pi \in \Pi_{\lfloor \alpha K \rfloor, K}} Z(\pi, \lfloor \alpha K \rfloor, K).$$

# Proof of Theorem 1 Cont'

On the other hand,

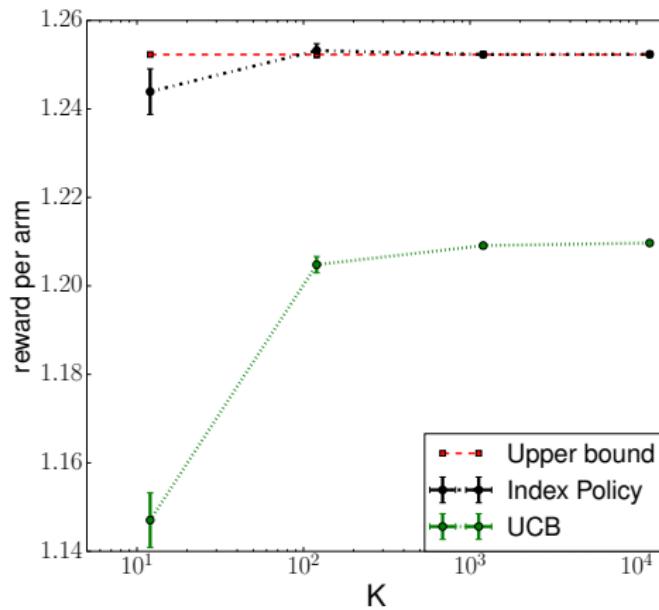
$$\begin{aligned}\lim_{K \rightarrow \infty} \frac{1}{K} Z(\hat{\pi}, \lfloor \alpha K \rfloor, K) &= \lim_{K \rightarrow \infty} \frac{1}{K} \mathbb{E}^{\hat{\pi}} \left[ \sum_{t=1}^T \sum_{s \in \mathbb{S}} r_t(s, 1) M_t(s) + r_t(s, 0) (N_t(s) - M_t(s)) \right] \\ &= \sum_{t=1}^T \sum_{s \in \mathbb{S}} [r_t(s, 1) \rho^*(s, 1, t) + r_t(s, 0) \rho^*(s, 0, t)] \\ &\quad - \mathbb{E}^{\pi^*} \left[ \sum_t \lambda_t^* (A_t - \alpha_t) \right] \\ &= Q(\boldsymbol{\lambda}^*) + \alpha \sum \lambda_t^* \\ &= \lim_{K \rightarrow \infty} \frac{1}{K} (KQ(\boldsymbol{\lambda}^*) + \lfloor \alpha K \rfloor \sum \lambda_t^*) \\ &= \lim_{K \rightarrow \infty} \frac{1}{K} P(\boldsymbol{\lambda}^*, \lfloor \alpha K \rfloor, K) \\ &\geq \lim_{K \rightarrow \infty} \frac{1}{K} \sup_{\boldsymbol{\pi} \in \Pi_{\lfloor \alpha K \rfloor, K}} Z(\boldsymbol{\pi}, \lfloor \alpha K \rfloor, K).\end{aligned}$$

# Numerical Experiments: Bernoulli MAB

$K$  arms, each returns a random reward of 0 or 1 when pulled.

$T = 6$ ,  $\alpha_t = \frac{1}{3}$ , for all  $t$ .

The problem aims to maximize the total expected reward.

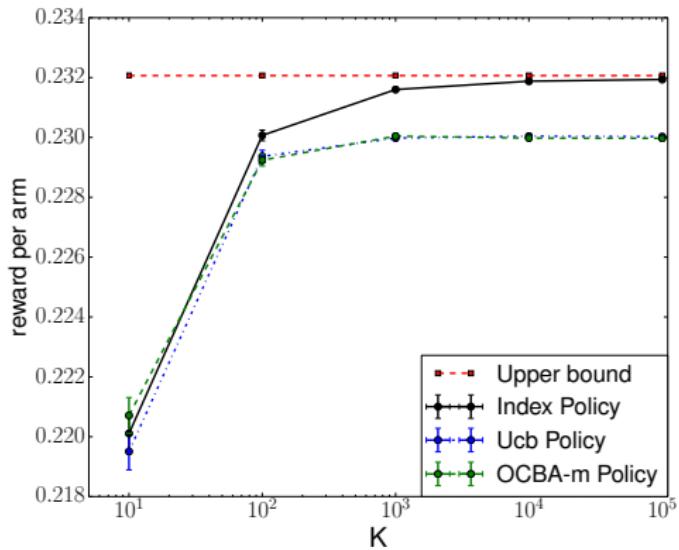


# Numerical Experiments: Subset Selection Problem

$K$  designs,  $m$  parallel computing resources.  $T = 5$

$\alpha_t = 0.5$  for  $1 \leq t \leq 4$ .  $\alpha_5 = 0.3$

The problem aims to select the best  $\bar{m}$  designs out of  $K$ .



# Application 1: Multiple Selection with Standard

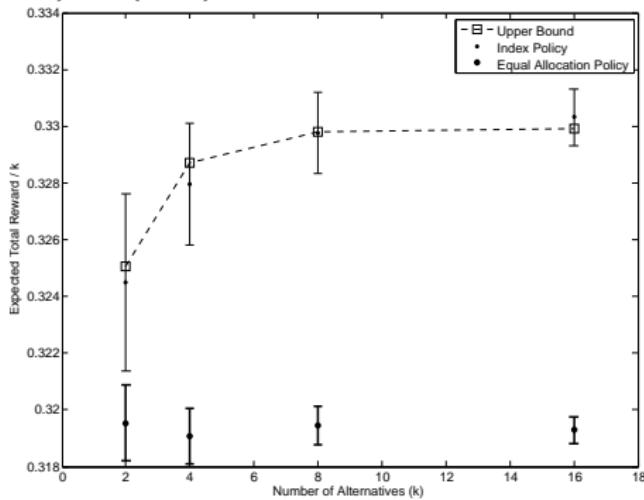
- $k$  alternatives
- $m$  parallel simulating resources per time step
- $T$  time horizon
- $\theta_x$  is the underlying true performance of alternative  $x$ ,  
 $x \in \{1, \dots, k\}$
- $d_x$  the known threshold for alternative  $x$
- $z_{n,x}$  is the number of simulation resources to use on alternative  $x$  at time step  $n$ . This is our allocation decision.  $\sum_x z_{n,x} \leq m$
- After time step  $T$ , for each alternative  $x$ , we decide whether  $\theta_x > d_x$  based on the past results of simulation
- We obtain a reward  $R = \sum_x R_x$ .  $R_x$  can be a 0-1 reward or linear reward.

Goal: Find an allocation of simulation resources to best support the decision at time  $T$

# Application 1: Multiple Selection with Standard

For each  $k = 2, 4, 8, 16$

- $m = k$
- $d_x = 0.2, \forall x \in \{1, \dots, k\}$
- $\mathbf{S}_0 = (\alpha_0, \beta_0) = (1, 1)^k$



Conjecture: The index policy has asymptotic optimality as  $m$  and  $K$  goes to infinity at the same rate.

## Application 2: Classification with Crowdsourcing

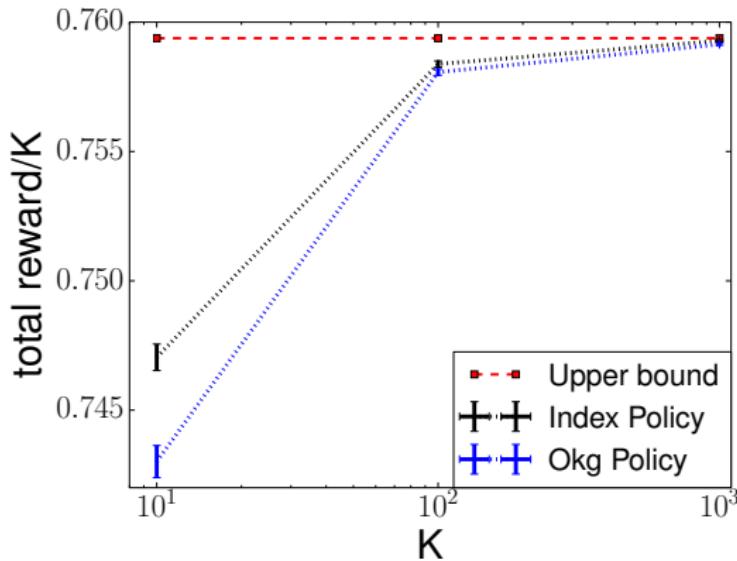
- $k$  labeling tasks
- $U$  total workers
- M/M/k queue: workers come in with rate  $r$ , and complete their job with rate  $\mu$ .
- $\theta_x$  is the underlying likelihood for a task to have a positive label.
- $d_x$  the known threshold for alternative  $x$
- After worker budget  $U$  has been exhausted, for each alternative  $x$ , we decide whether the true label is positive or negative based on a 1-0 reward.

Goal: Find an allocation of workers to best support the final decision on true labels

## Application 2: Classification with Crowdsourcing

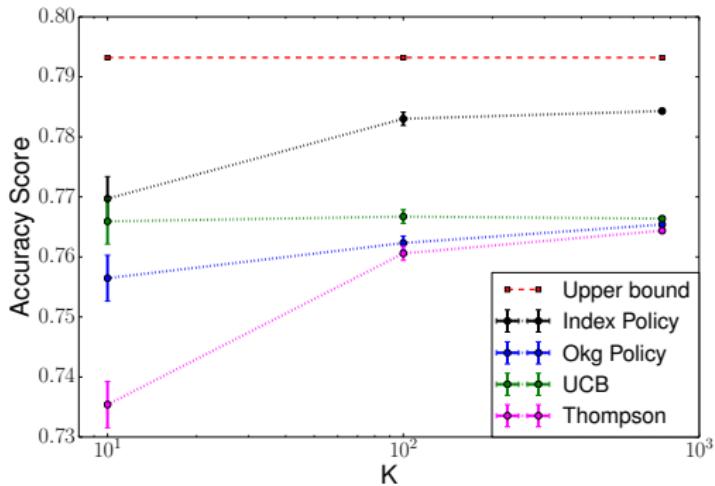
Queue:  $r = 0.1$ ,  $\mu = 0.4$ ,  $K = 10, 100, 1000$ ,  $U = 1.2K$ .

We use a non-informative prior with  $\alpha = 1$  and  $\beta = 1$ , and a threshold  $d_x = 0.5$  for all the tasks.



## Application 2: Classification with Crowdsourcing

We used dataset PASCAL RTE-1 [Snow2008], which consists of 800 tasks, each comes with 10 labels obtained from crowdworkers and a gold standard label



Conjecture: The index policy has asymptotic optimality as  $U$  and  $K$  goes to infinity at the same rate.

Index policies under different setting show asymptotically optimal behaviors while significantly reduce the computational complexity.  
Possible extension of the framework:

- To include infinite state space
- To allow a total budget over all time period
- To allow multiple actions
- To allow a more general weakly coupled dynamic program setting