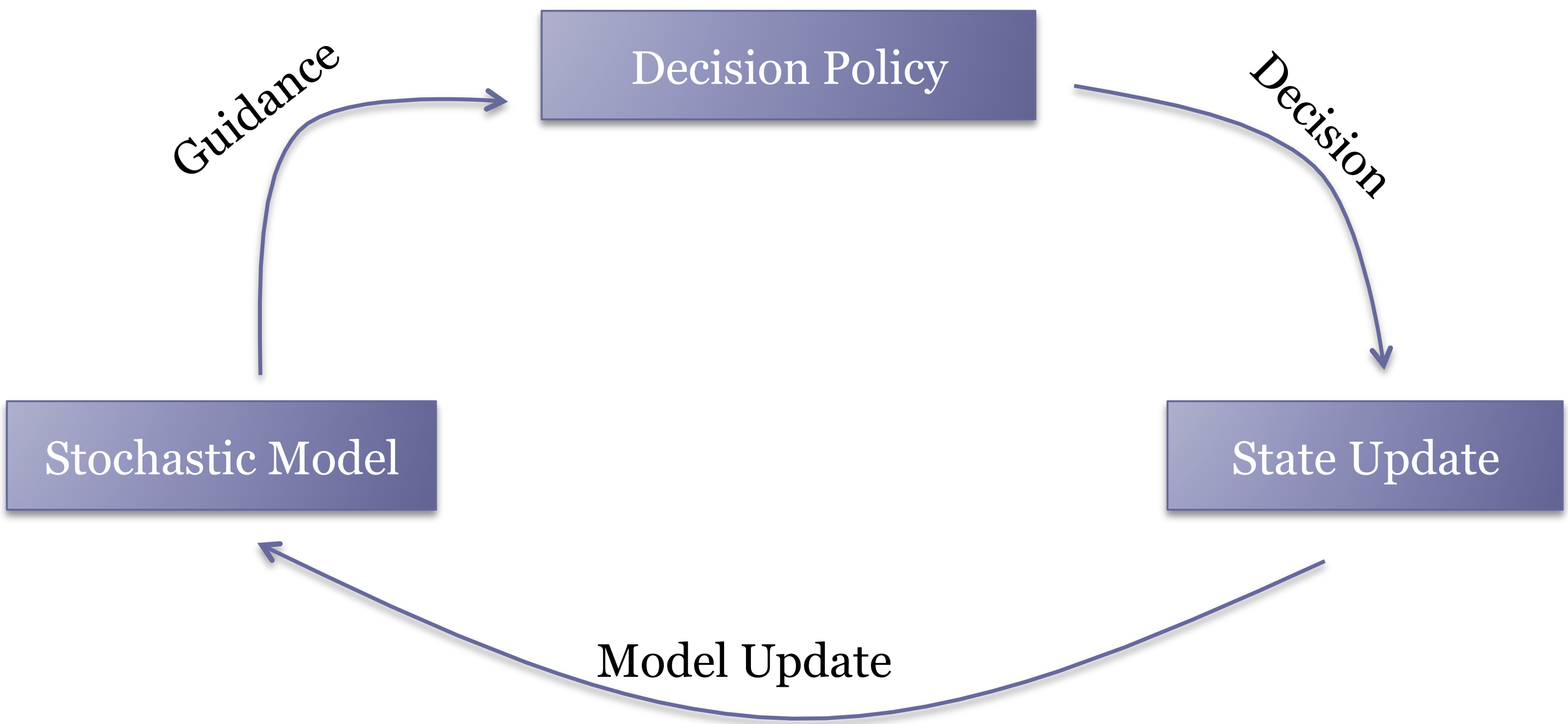
Prophet Inequalities and Stochastic Optimization

Kamesh Munagala
Duke University

Joint work with Sudipto Guha, UPenn

Bayesian Decision System



Approximating MDPs

- Computing decision policies typically requires exponential time and space
- Simpler decision policies?
  - Approximately optimal in a provable sense
  - Efficient to compute and execute
- This talk
  - Focus on a very simple decision problem
  - Known since the 1970's in statistics
  - Arises as a primitive in a wide range of decision problems

An Optimal Stopping Problem

- There is a gambler and a prophet (adversary)
- There are n boxes
  - Box j has reward drawn from distribution X_j
  - Gambler knows X_j but box is closed
  - All distributions are independent

An Optimal Stopping Problem

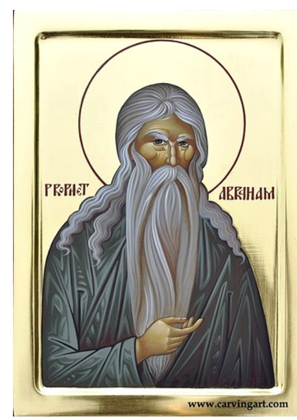
X_9

Order unknown to gambler

X_5 X_2 X_8 X_6

Curtain

- Gambler knows all the distributions
- Distributions are independent



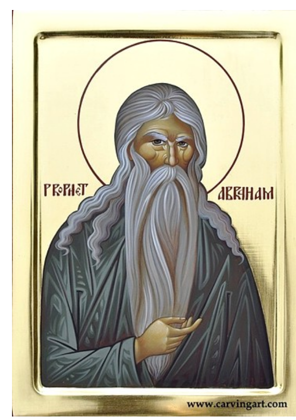
An Optimal Stopping Problem

Open box

20

X_5 X_2 X_8 X_6

Curtain



An Optimal Stopping Problem

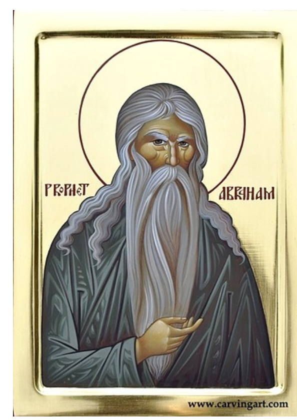
Keep it
or discard?

20

X_5 X_2 X_8 X_6

Keep it:
- Game stops and gambler's payoff = 20

Discard:
- Can't revisit this box
- Prophet shows next box



Stopping Rule for Gambler?

- Maximize expected payoff of gambler
  - Call this value ALG

- Compare against OPT = $\mathbf{E}[\max_j X_j]$
  - This is prophet's payoff assuming he knows the values inside all the boxes

- Can the gambler compete against OPT?

The Prophet Inequality

[Krengel, Sucheston, and Garling ‘77]

There exists a value w such that, if the gambler stops when he observes a value at least w, then:

$$\mathbf{ALG} \geq \tfrac{1}{2}\,\mathbf{OPT} = \tfrac{1}{2}\,\mathbf{E}[\max_j X_j]$$

Gambler computes threshold w from the distributions

Talk Outline

- Three algorithms for the gambler
  - Closed form for threshold
  - Linear programming relaxation
  - Dual balancing (if time permits)
- Connection to policies for stochastic scheduling
  - “Weakly coupled” decision systems
  - Multi-armed Bandits with martingale rewards

First Proof

[Samuel-Cahn '84]

Threshold Policies

Let $X^* = \max_j X_j$

Choose threshold w as follows:

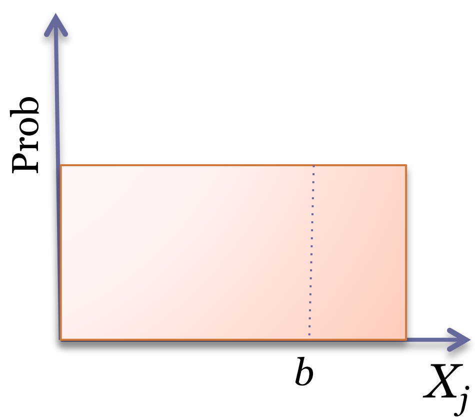
- **[Samuel-Cahn '84]** $\Pr[X^* > w] = ½$
- **[Kleinberg, Weinberg '12]** $w = ½\, \mathbf{E}[X^*]$

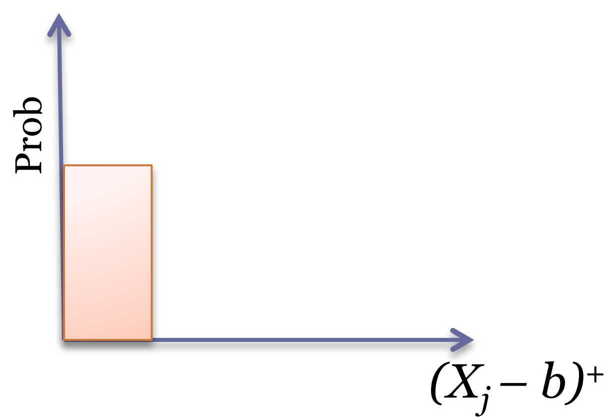
In general, many different threshold rules work

Let (unknown) order of arrival be $X_1\ X_2\ X_3 \ldots$

The Excess Random Variable

$$\text{Let} \quad (X_j - b)^+ = \max(X_j - b, 0)$$





Accounting for Reward

- Suppose threshold = w
- If $X^* \geq w$ then some box is chosen
  - Policy yields fixed payoff w

Accounting for Reward

- Suppose threshold = w
- If $X^* \geq w$ then some box is chosen
  - Policy yields fixed payoff w
- If policy encounters box j
  - It yields excess payoff $(X_j - w)^+$
  - If this payoff is positive, the policy stops.
  - If this payoff is zero, the policy continues.
- Add these two terms to compute actual payoff

In math terms…

Fixed payoff of w

$$\text{Payoff} = w \times \Pr[X^* \geq w]$$

$$+ \sum_{j=1}^{n} \Pr[j \text{ encountered}] \times \mathbf{E}\left[(X_j - w)^+\right]$$

Excess payoff conditioned on reaching j

Event of reaching j is independent of the value observed in box j

A Simple Inequality

$$\begin{aligned}
\Pr[j \text{ encountered}] &= \Pr\left[\max_{i=1}^{j-1} X_i < w\right] \\
&\geq \Pr\left[\max_{i=1}^{n} X_i < w\right] \\
&= \Pr[X^* < w]
\end{aligned}$$

Putting it all together…

$$\begin{aligned}\text{Payoff} \quad &\geq \quad w \times \Pr[X^* \geq w] \\ &+ \quad \textstyle\sum_{j=1}^{n} \Pr[X^* < w] \times \mathbf{E}\left[(X_j - w)^+\right]\end{aligned}$$

Lower bound on Pr[j encountered]

Simplifying...

$$\text{Payoff} \geq w \times \Pr[X^* \geq w]$$

$$+ \sum_{j=1}^{n} \Pr[X^* < w] \times \mathbf{E}\left[(X_j - w)^+\right]$$

Suppose we set $w = \sum_{j=1}^{n} \mathbf{E}\left[(X_j - w)^+\right]$

Then payoff ≥ w

Why is this any good?

$$\boxed{w \quad = \quad \sum_{j=1}^{n} \mathbf{E}\left[(X_j - w)^+\right]}$$

$$\begin{aligned}
2w &= w + \mathbf{E}\left[\sum_{j=1}^{n} (X_j - w)^+\right] \\
&\geq w + \mathbf{E}\left[(\max_{j=1}^{n} X_j - w)^+\right] \\
&= \mathbf{E}\left[\max_{j=1}^{n} X_j\right] = \mathbf{E}[X^*]
\end{aligned}$$

Summary

[Samuel-Cahn '84]

Choose threshold $w \quad = \quad \sum_{j=1}^{n} \mathbf{E}\left[(X_j - w)^+\right]$

Yields payoff $w \quad \geq \quad \mathbf{E}[X^*]/2 = OPT/2$

Exercise: The factor of 2 is optimal even for 2 boxes!

Second Proof

Linear Programming

[Guha, Munagala '07]

Why Linear Programming?

- Previous proof appears "magical"
  - Guess a policy and cleverly prove it works
- LPs give a "decision policy" view
  - Recipe for deriving solution
  - Naturally yields threshold policies
  - Can be generalized to complex decision problems
- Some caveats later…

Linear Programming

Consider behavior of prophet

- Chooses max. payoff box
- Choice depends on all realized payoffs

$$
\begin{aligned}
z_{jv} &= \Pr[\text{Chooses box } j \wedge X_j = v] \\
&= \Pr[X_j = X^* \wedge X_j = v]
\end{aligned}
$$

Basic Idea

- LP captures prophet behavior
  - Use z_{jv} as the variables

- These variables are insufficient to capture prophet choosing the maximum box
  - What we end up with will be a *relaxation* of max

- Steps:
  - Understand structure of relaxation
  - Convert solution to a feasible policy for gambler

Constraints

$$z_{jv} \quad = \quad \Pr[X_j = X^* \wedge X_j = v]$$

$$\boxed{\Rightarrow z_{jv} \quad \leq \quad \Pr[X_j = v] = f_j(v)}$$

Relaxation

Constraints

$$z_{jv} \quad = \quad \Pr[X_j = X^* \wedge X_j = v]$$

$$\boxed{\Rightarrow z_{jv} \quad \leq \quad \Pr[X_j = v] = f_j(v)}$$

Prophet chooses exactly one box:

$$\sum_{j,v} z_{jv} \quad \leq \quad 1$$

Constraints

$$z_{jv} \quad = \quad \Pr[X_j = X^* \wedge X_j = v]$$

$$\boxed{\Rightarrow z_{jv} \quad \leq \quad \Pr[X_j = v] = f_j(v)}$$

Prophet chooses exactly one box:

$$\sum_{j,v} z_{jv} \quad \leq \quad 1$$

Payoff of prophet:

$$\sum_{j,v} v \times z_{jv}$$

LP Relaxation of Prophet's Problem

$$
\begin{array}{rcll}
\text{Maximize} & & \sum_{j,v} v \cdot z_{jv} & \\
\sum_{j,v} z_{jv} & \leq & 1 & \\
z_{jv} & \in & [0, f_j(v)] & \forall j, v
\end{array}
$$

Example

X_a → 2 with probability ½

X_a → 0 with probability ½

X_b → 1 with probability ½

X_b → 0 with probability ½

LP Relaxation

X_a → **2 with probability ½** (z_{a2}) / 0 with probability ½

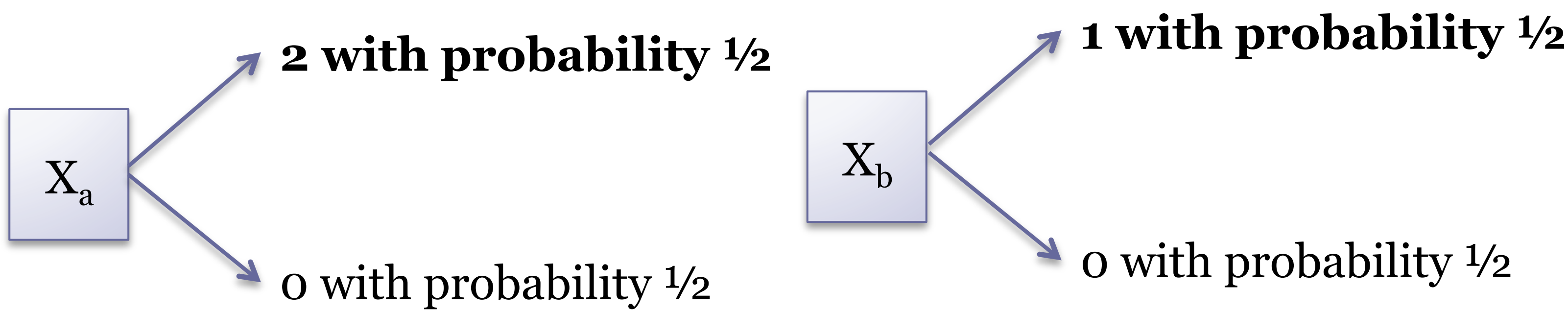
X_b → **1 with probability ½** (z_{b1}) / 0 with probability ½

$$\text{Maximize} \qquad 2 \times z_{a2} + 1 \times z_{b1}$$

$$\begin{aligned} z_{a2} + z_{b1} &\leq 1 \\ z_{a2} &\in [0, 1/2] \\ z_{b1} &\in [0, 1/2] \end{aligned}$$

Relaxation

LP Optimum



$$\text{Maximize} \qquad 2 \times z_{a2} + 1 \times z_{b1}$$

$$\begin{aligned} z_{a2} + z_{b1} &\leq 1 \\ z_{a2} &\in [0, 1/2] \\ z_{b1} &\in [0, 1/2] \end{aligned}$$

$z_{a2} = 1/2$

$z_{b1} = 1/2$

LP optimal payoff

= 1.5

Expected Value of Max?

X_a → **2 with probability ½**

X_a → 0 with probability ½

X_b → **1 with probability ½**

X_b → 0 with probability ½

$$\text{Maximize} \qquad 2 \times z_{a2} + 1 \times z_{b1}$$

$$\begin{aligned} z_{a2} + z_{b1} &\leq 1 \\ z_{a2} &\in [0, 1/2] \\ z_{b1} &\in [0, 1/2] \end{aligned}$$

$z_{a2} = 1/2$

$z_{b1} = 1/4$

Prophet's payoff

= 1.25

What do we do with LP solution?

- Will convert it into a feasible policy for gambler
- Bound the payoff of gambler in terms of LP optimum
  - LP Optimum upper bounds prophet's payoff!

Interpreting LP Variables for Box *j*

- Policy for choosing box if encountered

X_b

1 with probability ½

$z_{b1} = \frac{1}{4}$

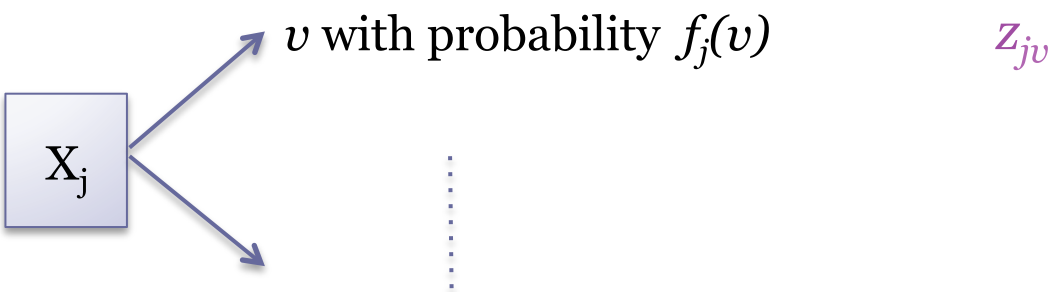
0 with probability ½

If $X_b = 1$ then

Choose b w.p. $z_{b1} / \frac{1}{2} = \frac{1}{2}$

Implies $\Pr[\,j \text{ chosen and } X_j = 1] = z_{b_1} = \frac{1}{4}$

LP Variables yield Single-box Policy P_j



If $X_j = v$ then

Choose j with probability $z_{jv} / f_j(v)$

Simpler Notation

$$
\begin{aligned}
C(P_j) = \quad & \Pr[j \text{ chosen}] & = \quad & \textstyle\sum_v \Pr\left[X_j = v \wedge j \text{ chosen }\right] \\
& & = \quad & \textstyle\sum_v z_{jv} \\
R(P_j) = \quad & \mathbf{E}[\text{Reward from } j] & = \quad & \textstyle\sum_v v \times \Pr\left[X_j = v \wedge j \text{ chosen }\right] \\
& & = \quad & \textstyle\sum_v v \times z_{jv}
\end{aligned}
$$

LP Relaxation

$$\begin{array}{rcll} \text{Maximize} & & \sum_{j,v} v \cdot z_{jv} & \\ \sum_{j,v} z_{jv} & \leq & 1 & \\ z_{jv} & \in & [0, f_j(v)] & \forall j, v \end{array}$$

$$\text{Maximize Payoff} = \sum_j R(P_j)$$

$$\mathbf{E}\left[\text{Boxes Chosen}\right] = \sum_j C(P_j) \leq 1$$

Each policy P_j is valid

LP yields collection of Single Box Policies!

LP Optimum

X_a

2 with probability ½

Choose a

0 with probability ½

X_b

1 with probability ½

Choose b

0 with probability ½

$$R(P_a) = \tfrac{1}{2} \times 2 = 1$$

$$R(P_b) = \tfrac{1}{2} \times 1 = \tfrac{1}{2}$$

$$C(P_a) = \tfrac{1}{2} \times 1 = \tfrac{1}{2}$$

$$C(P_b) = \tfrac{1}{2} \times 1 = \tfrac{1}{2}$$

Lagrangian

Maximize $\sum_j R(P_j)$

$\sum_j C(P_j) \leq 1$ ← Dual variable = w

P_j feasible $\forall j$

$$\text{Max. } w + \sum_j \left(R(P_j) - w \times C(P_j)\right)$$

$$P_j \text{ feasible } \forall j$$

Interpretation of Lagrangian

$$\text{Max. } w + \sum_j \left(R(P_j) - w \times C(P_j)\right)$$

$$P_j \text{ feasible } \forall j$$

- Net payoff from choosing j = Value minus w

- Can choose many boxes

- Decouples into a separate optimization per box!

Optimal Solution to Lagrangian

If $X_j \geq w$ then choose box j !

- Net payoff from choosing j = Value minus w
- Can choose many boxes
- Decouples into a separate optimization per box!

Notation in terms of *w*...

$$C(P_j) \quad = \quad C_j(w) \; = \; \Pr[X_j \geq w]$$

$$R(P_j) \quad = \quad R_j(w) \; = \; \sum_{v \geq w} v \times \Pr[X_j = v]$$

Expected payoff of policy

If $X_j \geq w$ **then** Payoff = X_j **else** 0

Strong Duality

$$\mathrm{Lag}(w) \quad = \quad \sum_j R_j(w) + w \times \left(1 - \sum_j C_j(w)\right)$$

Choose Lagrange multiplier w such that

$$\boxed{\begin{aligned} \textstyle\sum_j C_j(w) \quad &= \quad 1 \\ \Rightarrow \quad \textstyle\sum_j R_j(w) \quad &= \quad \text{LP-OPT} \end{aligned}}$$

Constructing a Feasible Policy

- ***Solve LP:*** Compute w such that

$$\sum_j \Pr[X_j \geq w] \quad = \quad \sum_j C_j(w) \quad = \quad 1$$

- ***Execute:*** If Box *j* encountered
  - Skip it with probability ½
  - With probability ½ do:
    - Open the box and observe X_j
    - If $X_j \geq w$ then choose j and STOP

Analysis

If Box *j* encountered

Expected reward = ½ × $R_j(w)$

Using union bound (or Markov's inequality)

$$
\begin{aligned}
\Pr[j \text{ encountered }] &\geq 1 - \textstyle\sum_{i=1}^{j-1} \Pr\left[X_i \geq w \wedge i \text{ opened }\right] \\
&\geq 1 - \tfrac{1}{2} \textstyle\sum_{i=1}^{n} \Pr[X_i \geq w] \\
&= 1 - \tfrac{1}{2} \textstyle\sum_{i=1}^{n} C_i(w) = \tfrac{1}{2}
\end{aligned}
$$

Analysis: ¼ Approximation

If Box j encountered

Expected reward = $\frac{1}{2} \times R_j(w)$

Box j encountered with probability at least ½

Therefore:

$$\begin{aligned} \text{Expected payoff} \quad &\geq \quad \tfrac{1}{4} \textstyle\sum_j R_j(w) \\ &= \quad \tfrac{1}{4}\text{LP-OPT} \geq \frac{OPT}{4} \end{aligned}$$

Third Proof

Dual Balancing

[Guha, Munagala '09]

Lagrangian Lag(w)

Maximize $\sum_j R(P_j)$

$\sum_j C(P_j) \leq 1$ ← Dual variable = w

P_j feasible $\forall j$

$$\text{Max. } w + \sum_j (R(P_j) - w \times C(P_j))$$

$$P_j \text{ feasible } \forall j$$

Weak Duality

$$\begin{aligned} \mathrm{Lag}(w) &= w + \textstyle\sum_j \Phi_j(w) \\ &= w + \textstyle\sum_j \mathbf{E}\left[(X_j - w)^+\right] \end{aligned}$$

Weak Duality: For all w, Lag$(w) \geq$ LP-OPT

Amortized Accounting for Single Box

$$\Phi_j(w) \quad = \quad R_j(w) - w \times C_j(w)$$

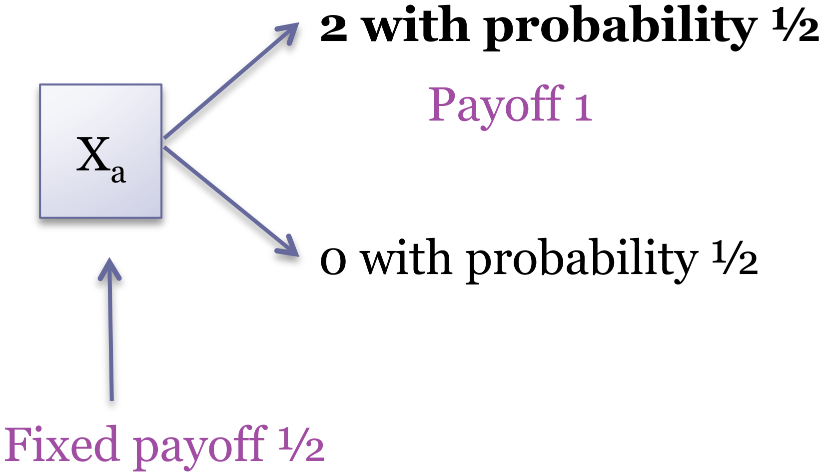
$$\Rightarrow \; R_j(w) \quad = \quad \Phi_j(w) + w \times C_j(w)$$

Fixed payoff for opening box

Payoff w if box is chosen

Expected payoff of policy is preserved in new accounting

Example: w = 1



$$\Phi_a(1) = \mathbf{E}\left[(X_a - 1)^+\right] = \tfrac{1}{2}$$

$$R_a(1) = 2 \times \tfrac{1}{2} = 1$$

$$\boxed{\Phi_a(w) + \frac{1}{2} \times w = \frac{1}{2} + \frac{1}{2} = 1}$$

Balancing Algorithm

$$\begin{aligned} \text{Lag}(w) &= w + \textstyle\sum_j \Phi_j(w) \\ &= w + \textstyle\sum_j \mathbf{E}\left[(X_j - w)^+\right] \end{aligned}$$

Weak Duality: For all w, Lag(w) ≥ LP-OPT

$$\begin{aligned} \text{Suppose we set} \quad w &= \textstyle\sum_j \Phi_j(w) \\ \text{Then} \quad w &\geq \text{LP-OPT}/2 \\ \text{and} \quad \textstyle\sum_j \Phi_j(w) &\geq \text{LP-OPT}/2 \end{aligned}$$

Algorithm

[Guha, Munagala '09]

- Choose w to balance it with total "excess payoff"
- Choose first box with payoff at least w
  - Same as Threshold algorithm of [Samuel-Cahn '84]
- Analysis:
  - Account for payoff using amortized scheme

Analysis: Case 1

- Algorithm chooses some box
- In amortized accounting:
  - Payoff when box is chosen = w
- Amortized payoff = $w \geq$ LP-OPT / 2

Analysis: Case 2

- All boxes opened

- In amortized accounting:
  - Each box j yields fixed payoff $\Phi_j(w)$

- Since all boxes are opened:
  - Total amortized payoff = $\Sigma_j \Phi_j(w) \geq$ LP-OPT / 2

Either Case 1 or Case 2 happens!

Implies Expected Payoff $\geq$ LP-OPT / 2

Takeaways…

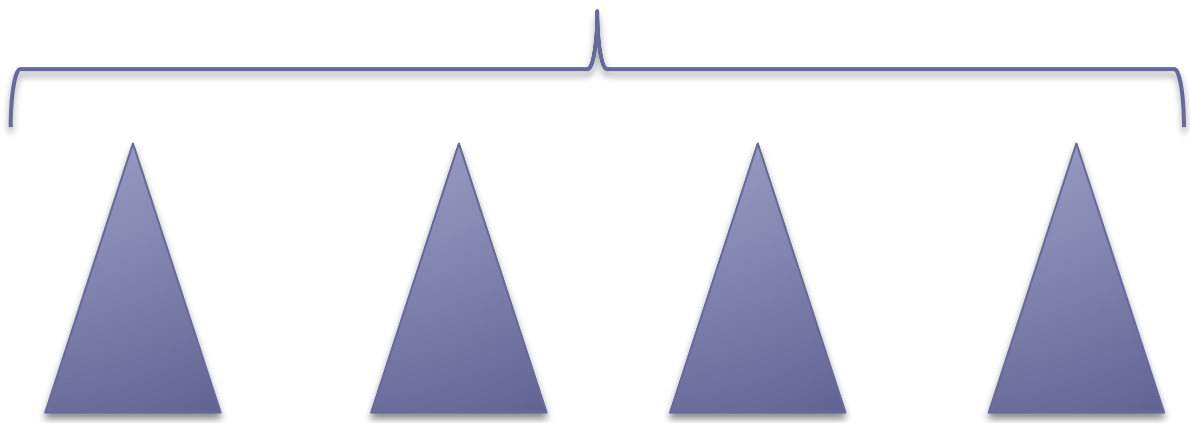
- LP-based proof is oblivious to closed forms
  - Did not even use probabilities in dual-based proof!
- Automatically yields policies with right “form”
- Needs independence of random variables
  - “Weak coupling”

General Framework

Weakly Coupled Decision Systems

Independent decision spaces

Few constraints coupling decisions across spaces



[Singh & Cohn '97; Meuleau *et al*. '98]

Prophet Inequality Setting

- Each box defines its own decision space
  - Payoffs of boxes are independent
- Coupling constraint:
  - At most one box can be finally chosen

Multi-armed Bandits

- Each bandit arm defines its own decision space
  - Arms are independent
- Coupling constraint:
  - Can play at most one arm per step
- Weaker coupling constraint:
  - Can play at most T arms in horizon of T steps
- Threshold policy ≈ Index policy

Bayesian Auctions

- Decision space of each agent
  - What value to bid for items
  - Agent's valuations are independent of other agents

- Coupling constraints
  - Auctioneer matches items to agents

- Constraints per bidder:
  - Incentive compatibility
  - Budget constraints

- Threshold policy = Posted prices for items

Prophet-style Ideas

- Stochastic Scheduling and Multi-armed Bandits
  - **Kleinberg, Rabani, Tardos ‘97**
  - **Dean, Goemans, Vondrak ‘04**
  - **Guha, Munagala ’07, ‘09, ’10, ‘13**
  - **Goel, Khanna, Null ‘09**
  - **Farias, Madan ’11**

- Bayesian Auctions
  - **Bhattacharya, Conitzer, Munagala, Xia ‘10**
  - **Bhattacharya, Goel, Gollapudi, Munagala ‘10**
  - **Chawla, Hartline, Malec, Sivan ‘10**
  - **Chakraborty, Even-Dar, Guha, Mansour, Muthukrishnan ’10**
  - **Alaei ’11**

- Stochastic matchings
  - **Chen, Immorlica, Karlin, Mahdian, Rudra ‘09**
  - **Bansal, Gupta, Li, Mestre, Nagarajan, Rudra ‘10**

Generalized Prophet Inequalities

- *k*-choice prophets
  - **Hajiaghayi, Kleinberg, Sandholm '07**
- Prophets with matroid constraints
  - **Kleinberg, Weinberg '12**
  - Adaptive choice of thresholds
  - Extension to polymatroids in **Duetting, Kleinberg '14**
- Prophets with samples from distributions
  - **Duetting, Kleinberg, Weinberg '14**

Martingale Bandits

[Guha, Munagala '07, '13]
[Farias, Madan '11]

(Finite Horizon) Multi-armed Bandits

- n arms of unknown effectiveness
  - Model "effectiveness" as probability $p_i \in [0,1]$
  - All p_i are independent and unknown *a priori*

(Finite Horizon) Multi-armed Bandits

- n arms of unknown effectiveness
  - Model "effectiveness" as probability $p_i \in [0,1]$
  - All p_i are independent and unknown *a priori*
- At any step:
  - Play an arm i and observe its reward

(Finite Horizon) Multi-armed Bandits

- n arms of unknown effectiveness
  - Model "effectiveness" as probability $p_i \in [0,1]$
  - All p_i are independent and unknown *a priori*
- At any step:
  - Play an arm i and observe its reward (0 or 1)
  - Repeat for at most T steps

(Finite Horizon) Multi-armed Bandits

- n arms of unknown effectiveness
  - Model "effectiveness" as probability $p_i \in [0,1]$
  - All p_i are independent and unknown *a priori*
- At any step:
  - Play an arm i and observe its reward (0 or 1)
  - Repeat for at most T steps
- Maximize expected total reward

What does it model?

- Exploration-exploitation trade-off
  - Value to playing arm with high expected reward
  - Value to refining knowledge of p_i
  - These two trade off with each other
- Very classical model; dates back many decades

  [Thompson '33, Wald '47, Arrow et al. '49, Robbins '50, …, Gittins & Jones '72, …]

Reward Distribution for arm *i*

- Pr[Reward = 1] = p_i
- Assume p_i drawn from a "prior distribution" Q_i
  - Prior refined using Bayes' rule into posterior

Conjugate Prior: Beta Density

- $Q_i = Beta(a,b)$
- $\Pr[p_i = x] \propto x^{a-1}\,(1-x)^{b-1}$

Conjugate Prior: Beta Density

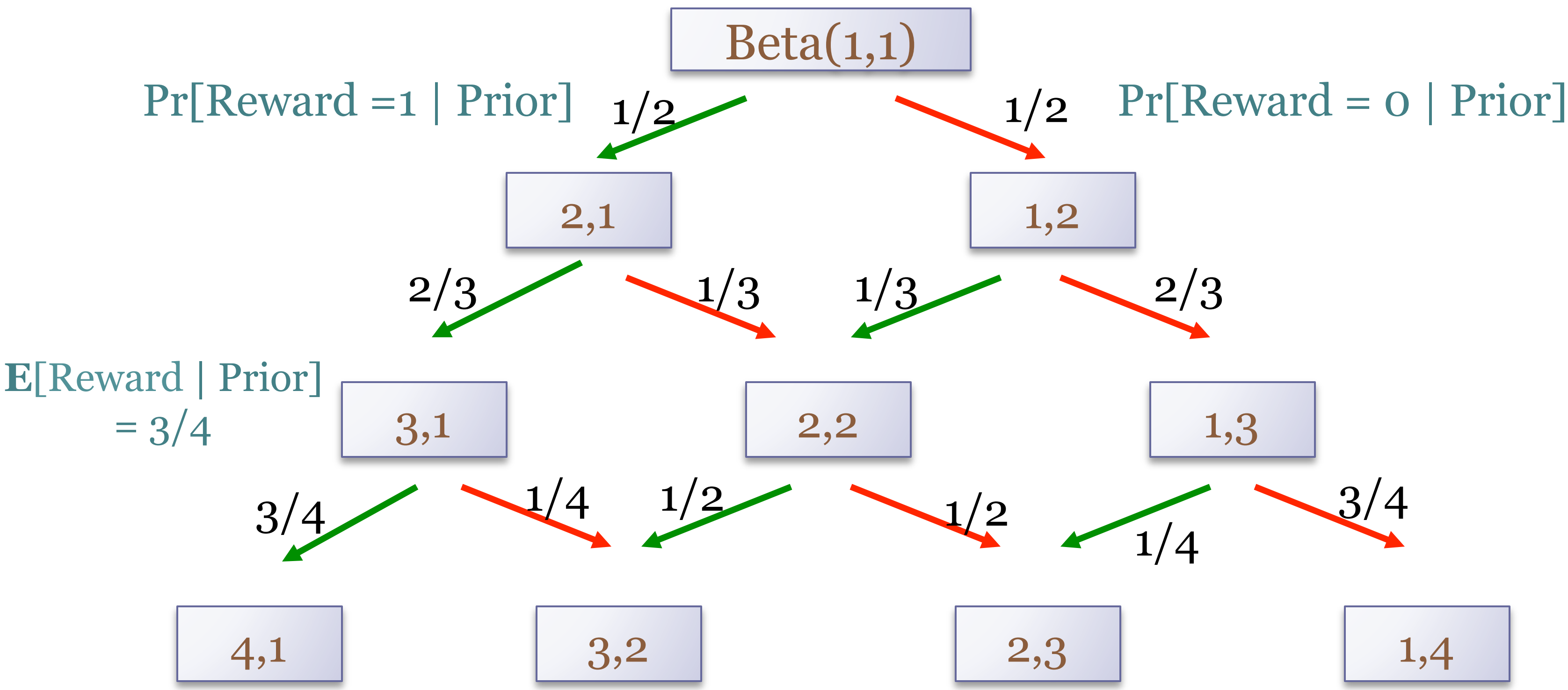
- $Q_i = Beta(a,b)$

- $\Pr[p_i = x] \propto x^{a-1}\,(1-x)^{b-1}$

- Intuition:
  - Suppose have previously observed $(a-1)$ **1's** and $(b-1)$ **0's**
  - $Beta(a,b)$ is posterior distribution given observations
  - Updated according to Bayes' rule starting with:
    - $Beta(1,1) = Uniform[0,1]$

- Expected Reward = $\mathbf{E}[p_i] = a/(a+b)$

Prior Update for Arm i



Convenient Abstraction

- Posterior distribution of arm captured by:
  - Observed rewards from arm so far
  - Called the “state” of the arm

Convenient Abstraction

- Posterior distribution of arm captured by:
    - Observed rewards from arm so far
    - Called the “state” of the arm
    - Expected reward evolves as a *martingale*

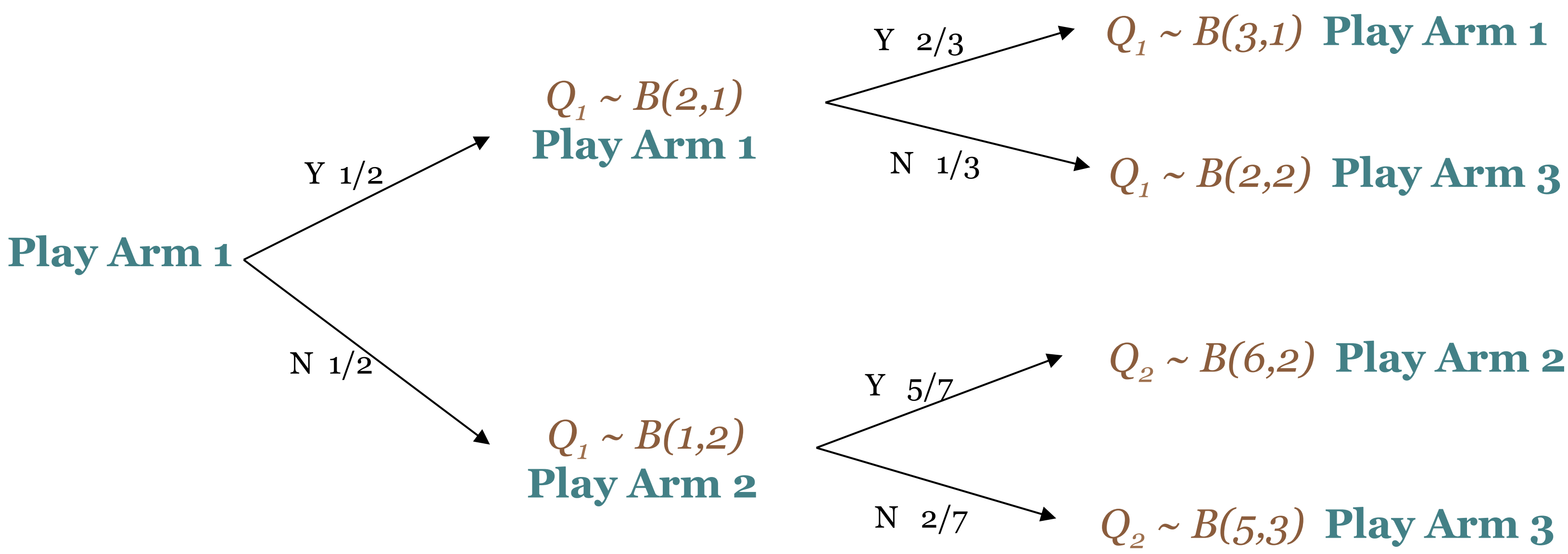
- State space of single arm typically small:
    - $O(T^2)$ if rewards are 0/1

Decision Policy for Playing Arms

- Specifies which arm to play next
- Function of current states of all the arms
- Can have exponential size description

Example: $T = 3$

$Q_1 = Beta(1,1)$ $Q_2 = Beta(5,2)$ $Q_3 = Beta(21,11)$



Goal

- Find decision policy with maximum value:
  - Value = **E [** Sum of rewards every step **]**

- Find the policy maximizing expected reward when p_i drawn from prior distribution Q_i
  - OPT = Expected value of optimal ***decision policy***

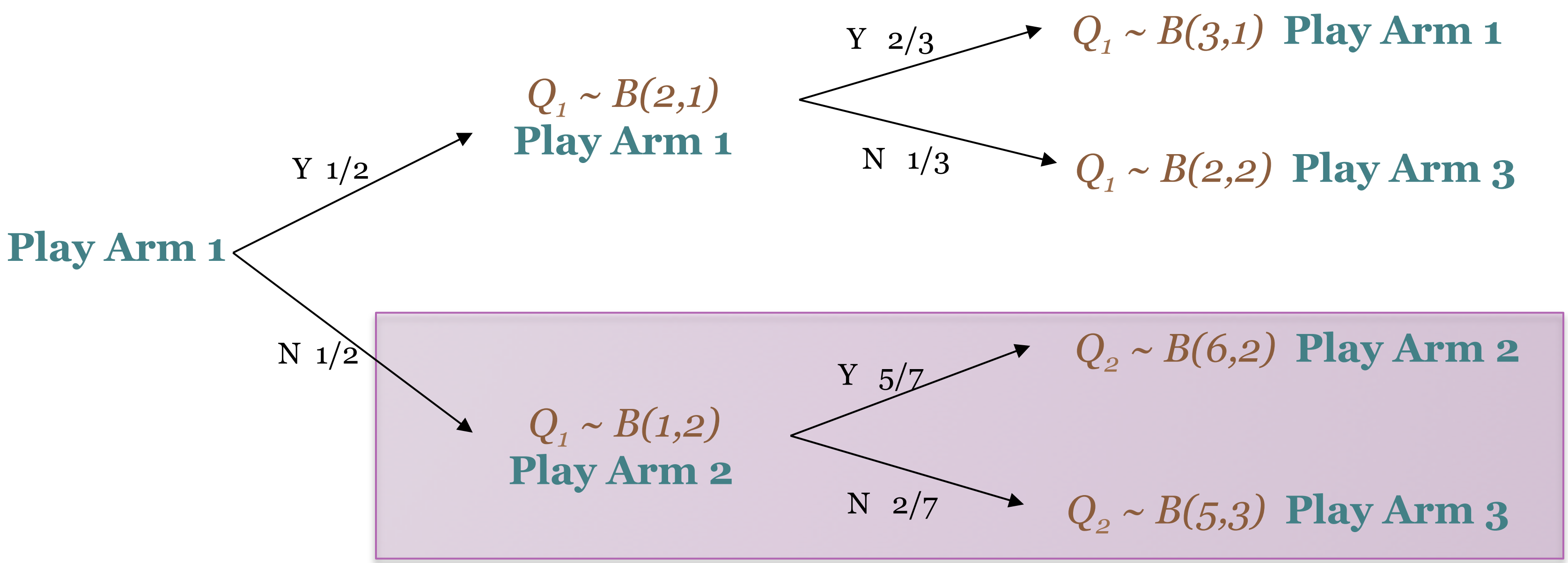
Solution Recipe using Prophets

Step 1: Projection

- Consider any decision policy **P**
- Consider its behavior restricted to arm i
- What state space does this define?
- What are the actions of this policy?

Example: Project onto Arm 2

$Q_1 \sim Beta(1,1) \quad Q_2 \sim Beta(5,2) \quad Q_3 \sim Beta(21,11)$



Behavior Restricted to Arm 2

$Q_2 \sim Beta(5,2)$

w.p. 1/2 **Play Arm 2**

Y $5/7$ → $Q_2 \sim B(6,2)$ **Play Arm 2**

N $2/7$ → $Q_2 \sim B(5,3)$ **STOP**

With remaining probability, do nothing

Plays are contiguous and ignore global clock!

Projection onto Arm i

- Yields a randomized policy for arm i

- At each state of the arm, policy probabilistically:
  - PLAYS the arm
  - STOPS and quits playing the arm

Notation

- $T_i = \mathbf{E}$[Number of plays made for arm i]
- $R_i = \mathbf{E}$[Reward from events when i chosen]

Step 2: Weak Coupling

- In any decision policy:
  - Number of plays is at most T
  - True on all decision paths

Step 2: Weak Coupling

- In any decision policy:
  - Number of plays is at most T
  - True on all decision paths

- Taking expectations over decision paths
  - $\Sigma_i T_i \leq T$
  - Reward of decision policy = $\Sigma_i R_i$

Relaxed Decision Problem

- Find a decision policy P_i for each arm i such that
  - $\Sigma_i \, T_i \, (P_i) \, / \, T \leq 1$
  - Maximize: $\Sigma_i \, R_i \, (P_i)$
- Let optimal value be *OPT*
  - $OPT \geq$ Value of optimal decision policy

Lagrangean with Penalty λ

- Find a decision policy P_i for each arm i such that
  - Maximize: $\lambda + \Sigma_i R_i (P_i) - \lambda \Sigma_i T_i (P_i) / T$
- No constraints connecting arms
  - Find optimal policy separately for each arm i

Lagrangean for Arm i

Maximize: $R_i(P_i) - \lambda\ T_i(P_i) / T$

- Actions for arm i:
  - PLAY: Pay penalty = λ/T & obtain reward
  - STOP and exit
- Optimum computed by dynamic programming:
  - Time per arm = Size of state space = $O(T^2)$
  - Similar to Gittins index computation
- Finally, binary search over λ

Step 3: Prophet-style Execution

- Execute single-arm policies sequentially
  - Do not revisit arms
- Stop when some constraint is violated
  - T steps elapse, or
  - Run out of arms

Analysis for Martingale Bandits

Idea: Truncation [Farias, Madan ‘11; Guha, Munagala ’13]

- Single arm policy defines a stopping time

- If policy is stopped after time αT
  - $\mathbf{E}[\text{Reward}] \geq \alpha R(P_i)$

- Requires “martingale property” of state space

- Holds only for the projection onto one arm!
  - Does not hold for optimal multi-arm policy

Proof of Truncation Theorem

$$R(P_i) = \int_p \underbrace{\Pr[Q_i = p]}_{\text{Probability Prior} = p} \times \underbrace{p \times \mathrm{E}[\text{Stopping Time} \mid Q_i = p]}_{\text{Reward given Prior} = p} dp$$

Truncation reduces this term by at most factor α

Analysis of Martingale MAB

- Recall: Collection of single arm policies s.t.
  - $\Sigma_i R(P_i) \geq OPT$
  - $\Sigma_i T(P_i) = T$
- Execute arms in decreasing $R(P_i)/T(P_i)$
  - Denote arms 1,2,3,… in this order
- If P_i quits, move to next arm

Arm-by-arm Accounting

- Let T_j = Time for which policy P_j executes
  - Random variable

- Time left for P_i to execute $= T - \sum_{j<i} T_j$

Arm-by-arm Accounting

- Let T_j = Time for which policy P_j executes
  - Random variable
- Time left for P_i to execute $= T - \sum_{j<i} T_j$
- Expected contribution of P_i conditioned on $j < i$

$$= \left(1 - \frac{1}{T} \sum_{j<i} T_j\right) R(P_i)$$

Uses the Truncation Theorem!

Taking Expectations…

- Expected contribution to reward from P_i

$$= E\left[\left(1 - \frac{1}{T}\sum_{j<i} T_j\right) R(P_i)\right]$$

$$\geq \left(1 - \frac{1}{T}\sum_{j<i} T(P_j)\right) R(P_i)$$

T_j independent of P_i

2-approximation

$$ALG \geq \sum_i \left(1 - \frac{1}{T}\sum_{j<i} T(P_j)\right) R(P_i)$$

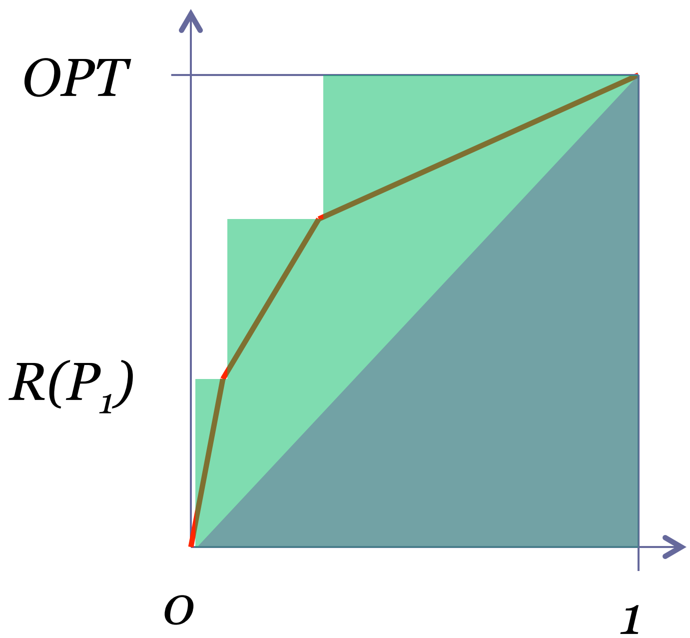
Constraints:

$$OPT = \sum_i R(P_i) \quad \& \quad T = \sum_i T(P_i)$$

$$\frac{R(P_1)}{T(P_1)} \geq \frac{R(P_2)}{T(P_2)} \geq \frac{R(P_3)}{T(P_3)} \geq \ldots$$

Implies:

$$ALG \geq \frac{OPT}{2}$$



Stochastic knapsack analysis
Dean, Goemans, Vondrak '04

Final Result

- 2-approximate irrevocable policy!
- Same idea works for several other problems
  - Concave rewards on arms
  - Delayed feedback about rewards
  - Metric switching costs between arms
- Dual balancing works for variants of bandits
  - Restless bandits
  - Budgeted learning

Open Questions

- How far can we push LP based techniques?
  - Can we encode adaptive policies more generally?
  - For instance, MAB with matroid constraints?
  - Some success for non-martingale bandits
- What if we don’t have full independence?
  - Some success in auction design
  - Techniques based on convex optimization
    - Seems unrelated to prophets

Thanks!