Stochastic Knapsack

Viswanath Nagarajan
University of Michigan

MDS Autumn School
Sep 29 – Oct 1, 2014

Adaptive Stochastic Optimization

The model:

- Known distribution **D**
- Unknown realized input X ~ **D**

Solution (policy) is adaptive sequence of actions

- Each action reveals extra information on X

Goal: policy having best *expected* objective

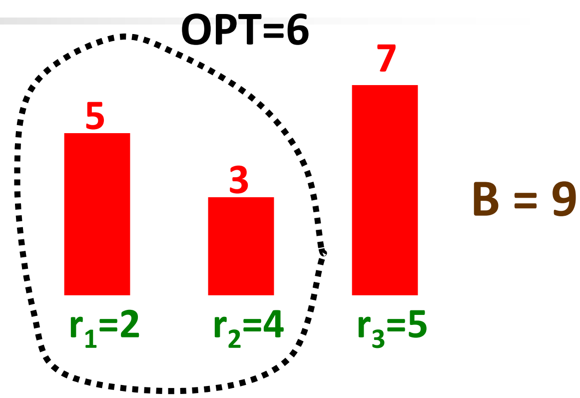
Here: maximization objective

1. Stochastic knapsack
2. Stochastic matching

Stochastic Knapsack

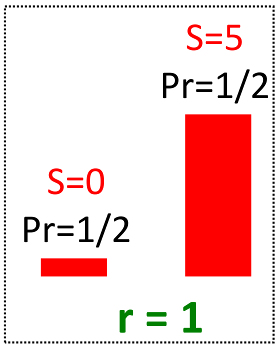
Recall: deterministic knapsack problem

- Jobs with reward and size; budget B
- Maximize *reward* s.t. total size $\leq$ **B**



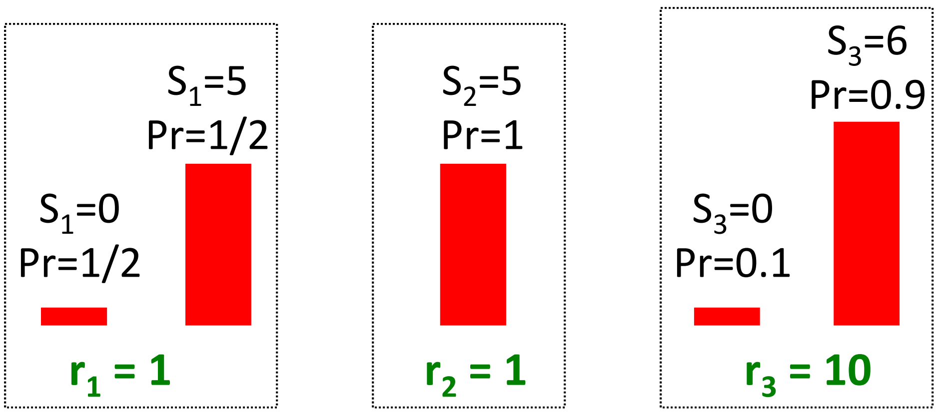
Stochastic knapsack **[Dean Goemans Vondrak '04]**

- Job sizes are *random*
  - Known, arbitrary distributions
  - Independent
  - Exact size known only *after selection*
- Deterministic rewards
- Maximize *expected reward* s.t. total size $\leq$ **B**

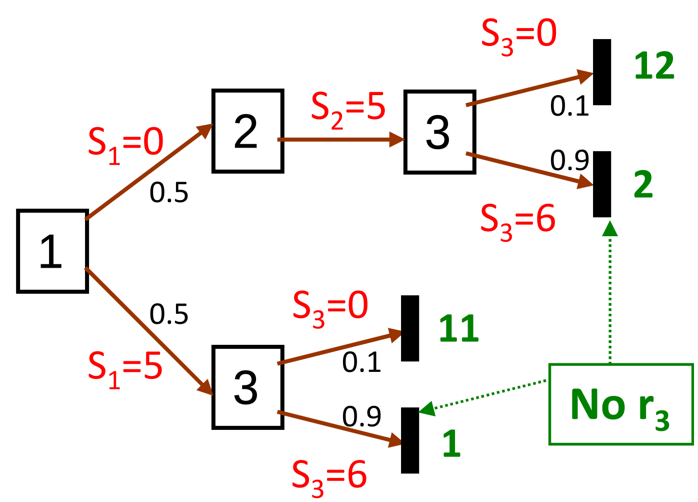


Example: Stochastic Knapsack

Feasible solution: select jobs sequentially



Budget B = 5



E [Reward] = ½ * [0.1*12 + 0.9*2] + ½ * [0.1*11 + 0.9*1] = 2.5

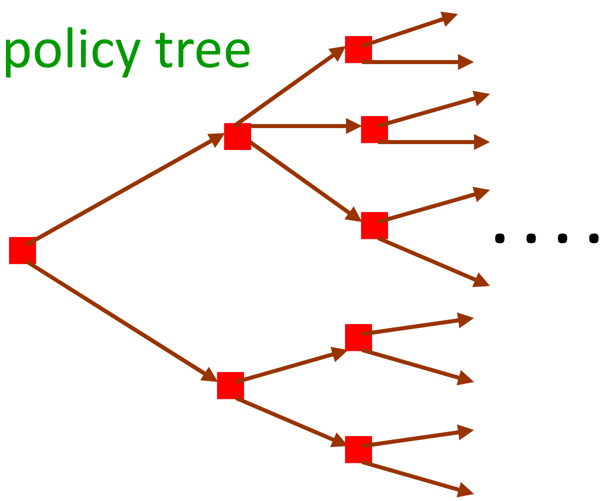
Representing Solutions

Decision tree or dynamic program

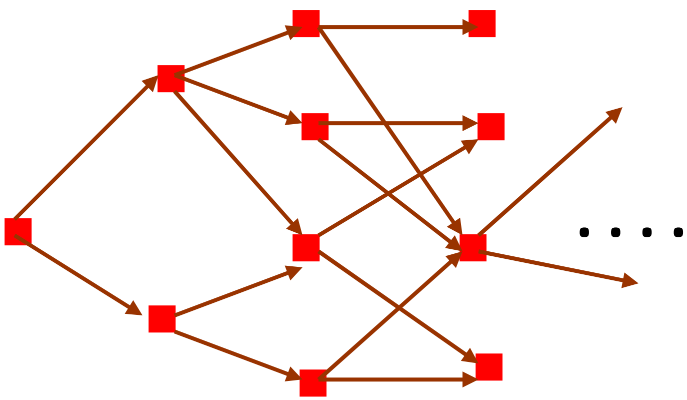
- **nodes** represent *residual budget* and *job chosen*
- **branches** represent random size outcomes
- a.k.a. *adaptive policy*

Describing policy may take exponential space

Eg. policy tree



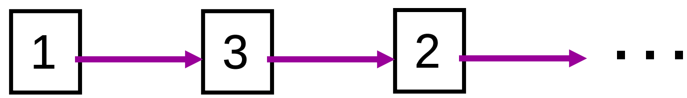
Eg. policy dynamic program



Simpler Class of Policies

Non adaptive policy

Add jobs in *a priori* fixed order, until budget exhausted

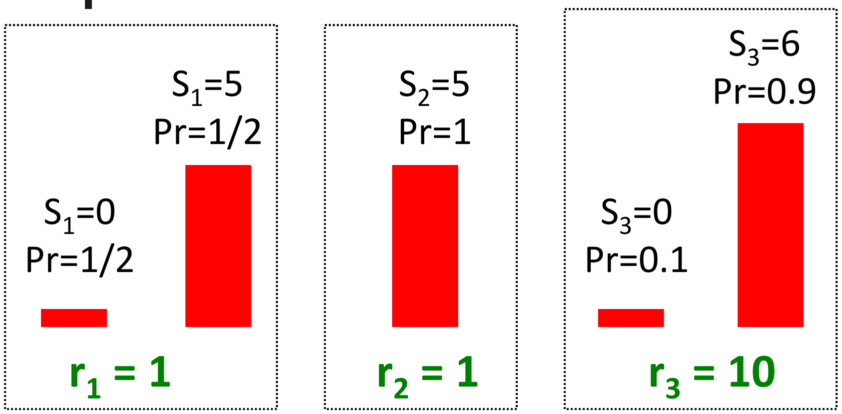


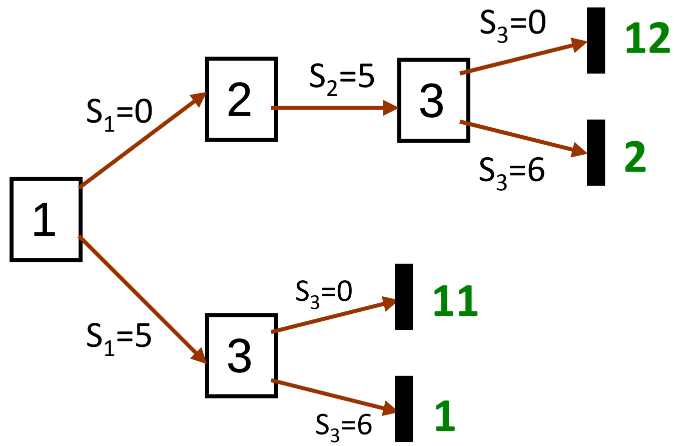
Polynomial space representation: *permutation*

Adaptivity gap captures loss in objective

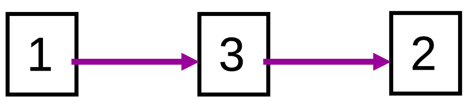
$$\max_{\text{instance } I} \frac{\text{OPT(AD(I))}}{\text{OPT(NA(I))}}$$

Adaptivity Gap Example

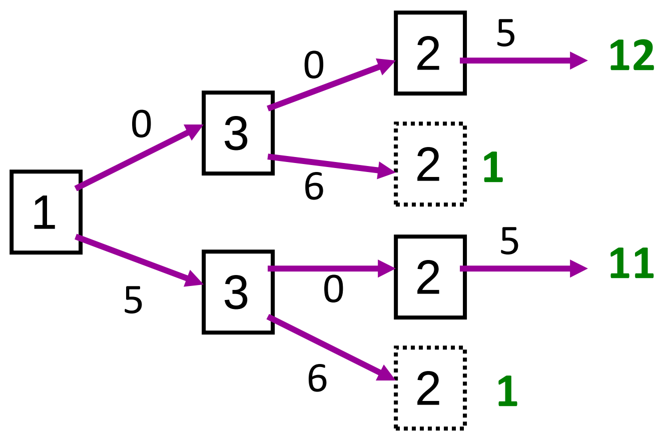




E[Adaptive] = 2.5



E [NonAdaptive] = 2.05



Adaptivity gap ≈ 1.25

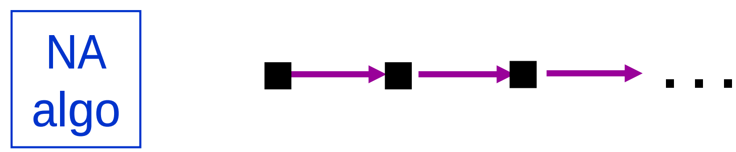
Why Non-Adaptive ?

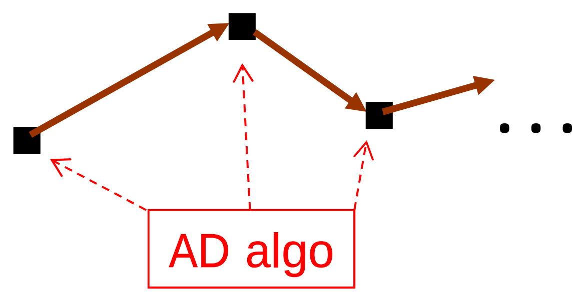
Problem consists of "offline" and "online" phases

- Offline = before any job is run
- Online = when jobs are running

Non-adaptive:

- All the work in offline phase
- Online phase easy/fast to implement





Approximation Ratio

Stochastic Knapsack instance I

OPT(I) = max **E** [objective under π] : π is policy

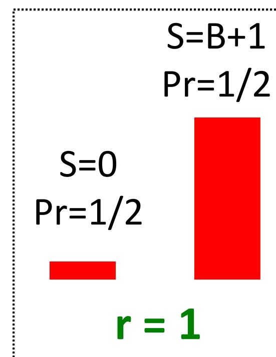
$$\text{Approximation ratio} = \max_{\text{instance I}} \frac{\text{OPT(I)}}{\text{ALG(I)}}$$

Contrast with *online* “competitive ratio” relative to:

EOPT(I) = E$_{X \leftarrow D}$ [optimal value on input X]

Eg. n *identical* jobs

EOPT = n/2 but **OPT** = 2



Here: information gradual in both ALG & OPT

Outline

1. Stochastic knapsack (SK) basics ✓
2. Non-adaptive algorithm for SK
3. Correlated stochastic knapsack (CSK)
   - Non-adaptive algorithm
4. Adaptive algorithm for CSK
5. Extensions

Some Definitions

- Jobs $[n] := \{1,2,\cdots,n\}$
- r_i = reward of job i
- S_i = size of job i (random variable)
- Budget **B** = 1 (by scaling)
- Capped mean size $e_i = E[\min\{S_i, 1\}]$
- *Effective* reward $w_i = r_i \cdot \Pr[S_i \leq 1]$

LP Relaxation

$$\max \quad \sum_{i=1}^{n} w_i \cdot x_i$$

$$\text{s.t.} \quad \sum_{i=1}^{n} e_i \cdot x_i \le 2$$

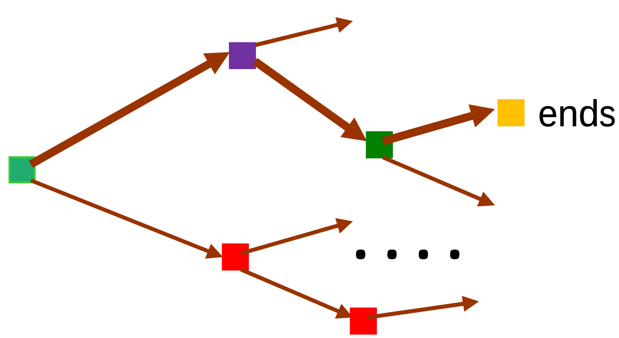
$$0 \le x_i \le 1, \quad \forall\, i=1, 2, \ldots, n$$

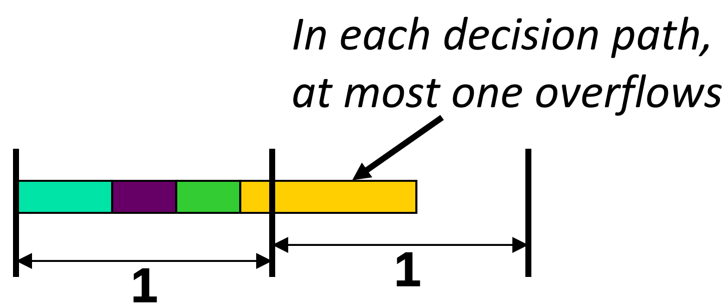
Theorem: $LP^* \ge OPT$

T_i := indicator that job i chosen in OPT (may not fit)

Consider LP solution $x_i = \Pr[T_i=1]$

Claim: $\sum_{i=1}^{n} \min\{S_i, 1\} \cdot T_i \le 2 \quad \Rightarrow \quad x \in LP$ (S_i & T_i independent)





LP relaxation: formal proof

- A_t := set of first t jobs chosen in OPT $(t=0,1,\cdots,n)$

  $A_0 = \emptyset$, A_n = all jobs chosen in OPT

  Note $\mathbf{T_i}=1$ iff $i \in A_n$

Thus $\mathbf{x} \in$ LP

- $Y_t := \sum_{i\in A_t} (\min\{S_i , 1\} - e_i)$
- Conditioned on Y_t and next job j :

$$E[Y_{t+1} \mid Y_t , j] = Y_t + E[\min\{S_j , 1\}] - e_j = Y_t$$

- $Y_0 \cdots Y_n$ martingale, $E[Y_n] = E[Y_0] = 0$
- i.e. $E[\sum_{i\in A_n} e_i] = E[\sum_{i\in A_n} \min\{S_j , 1\}] \leq 2$

$\uparrow$ $\sum_{i=1}^n e_i \cdot \Pr[\mathbf{T_i}=1]$

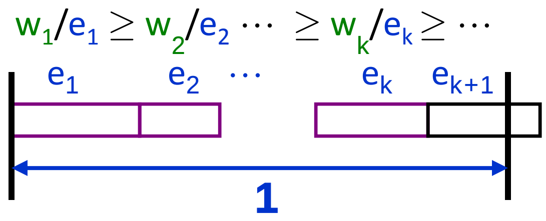
$\uparrow$ By **Claim**

Scaled LP

$\max \sum_{i=1}^{n} w_i \cdot x_i$
s.t. $\sum_{i=1}^{n} e_i \cdot x_i \leq \mathbf{2}$
$0 \leq x \leq 1.$

$\max \sum_{i=1}^{n} w_i \cdot x_i$
s.t. $\sum_{i=1}^{n} e_i \cdot x_i \leq \mathbf{1}$
$0 \leq x \leq 1.$

- LP(1) more convenient to work with
- Note $LP^*(1) \geq LP^*(2)/2 \geq OPT/2$
  $x \in LP(2) \Rightarrow x/2 \in LP(1)$
- $LP^*(1)$ has $x_1 = x_2 = \cdots = x_k = 1$, $x_{k+1}=\theta$ where:
  $e_1 + e_2 + \cdots + e_k + \theta \cdot e_{k+1} = 1$
- So $LP^*(1) \leq w_1 + w_2 + \cdots + w_{k+1}$



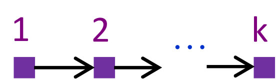
Algorithm

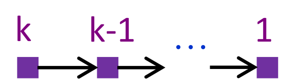
$G := \{1,2,\cdots,k\}$, $OPT/2 \leq LP^*(1) \leq w(G) + w_{k+1} \leq w(G) + w_{max}$

Algorithm: Run one of the following w.p. ½ each:

1. Place best *single* job

   Expected reward $ALG_1 \geq w_{max}$

2. Place jobs of G as: 

or 

   Expected reward ALG_2

Lemma: $ALG_2 \geq w(G)/2 - w_{max}/2$

$\Rightarrow ALG = ½\, ALG_1 + ½\, ALG_2 \geq w(G)/4 + w_{max}/4 \geq OPT/8$

Theorem: 8-approximation for stochastic knapsack.

Also adaptivity gap ≤ 8

Analysis of ALG_2

ALG_2 : $1 \to 2 \to \cdots \to k$ or $k \to k\text{-}1 \to \cdots \to 1$ $\qquad G = \{1,2,\cdots,k\}$

Lemma: $ALG_2 \geq w(G)/2 - w_{max}/2$

- $S'_i := \min\{ S_i , 1 \}$, so $E[S'_i] = e_i$
- Job v yields reward $\leftrightarrow$ v fits in knapsack $\leftrightarrow$ $S'_v + \sum_{i \prec v} S'_i \leq 1$
- $E[\sum_{i \prec v} S'_i] = ½ \cdot e(G \setminus v)$
- $\Pr[\text{ v doesn't fit }] \leq E[S'_v + \sum_{i \prec v} S'_i] \leq e(G)/2 + e_v/2 \leq ½ + e_v/2$

Markov's ineq.

$e(G) = \sum_{v \in G} e_v \leq 1$

- $ALG_2 = \sum_{v \in G} r_v \cdot \Pr[v \text{ fits}] \geq \sum_{v \in G} w_v \cdot (1/2 - e_v/2)$

$\geq w(G)/2 - w_{max}/2$

Better Bounds

Stochastic Knapsack

- 4-approx and adaptivity gap **[Dean Goemans Vondrak ’08]**
  - Stronger LP relaxation for AD-OPT
  - Better analysis of NA algorithm
- 3-approx adaptive algorithm **[Dean Goemans Vondrak ’08]**
- (2+ϵ)-approx adaptive algorithm **[Bhalgat Goel Khanna ’11]**

Outline

1. Stochastic knapsack (SK) basics ✓
2. Non-adaptive algorithm for SK ✓
3. **Correlated stochastic knapsack (CSK)**
   Non-adaptive algorithm
4. Adaptive algorithm for CSK
5. Extensions

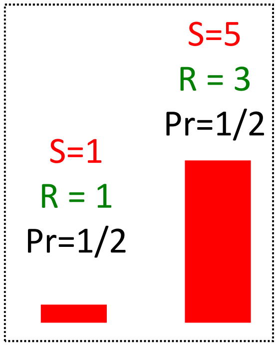
Correlated Stochastic Knapsack

[Gupta Krishnaswamy Molinaro Ravi ’11]

- Job rewards R_i also random, correlated to size S_i

  Joint distribution Pr [R_i=a , S_i=b]

- *Different jobs independent*
- Max expected reward s.t. budget **B**



- Assume all sizes are integral
- Assume $S_i \in \{0, 1, \cdots B\} = [B]$

  Wlog. by zeroing reward for larger sizes

- Distribution i = (p_{it}, r_{it}) for $t \in [B]$

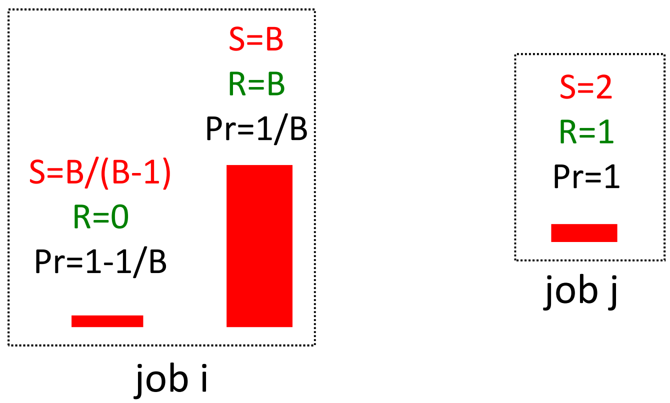
  $p_{it} = \Pr[S_i=t]$, $r_{it} = E[R_i \mid S_i=t]$

Distribution Information Used

- In uncorrelated case, we only used:

  Capped mean size e_i = E[min(S_i , B)]

  Effective reward w_i = E[$R_i \cdot \mathbf{1}_{S_i \leq B}$]



$e_i = e_j = 2$

$w_i = w_j = 1$

- SOL_1 = i, i,··· has reward 1
- SOL_2 = j, j,··· has reward B/2

$\Rightarrow$ Previous LP/algorithm insufficient

Need to use more info from distribution in CSK

LP Relaxation for CSK (1)

Use capped mean and reward at all sizes $t \in [B]$:

$e_{it} := E[\min(S_i , t)]$ $\qquad$ $w_{it} := E[R_i \cdot \mathbf{1}_{S_i \leq B-t}]$

Time indexed LP

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} \sum_{t=0}^{B} w_{it} \cdot x_{it} \\
\text{s.t.} \quad & \sum_{i=1}^{n} e_{it} \cdot \sum_{s=0}^{t} x_{is} \leq 2t, \quad \forall\, t \in [B] \\
& \sum_{t=0}^{B} x_{it} \leq 1, \quad \forall\, i \in [n] \\
& x \geq 0.
\end{aligned}
$$

Theorem: $LP^* \geq OPT$.

LP Relaxation for CSK (2)

$$\max \quad \sum_{i=1}^{n} \sum_{t=0}^{B} w_{it} \cdot x_{it}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} e_{it} \cdot \sum_{s=0}^{t} x_{is} \le 2t, \quad \forall\, t \in [B]$$

$$\sum_{t=0}^{B} x_{it} \le 1, \quad \forall\, i \in [n]$$

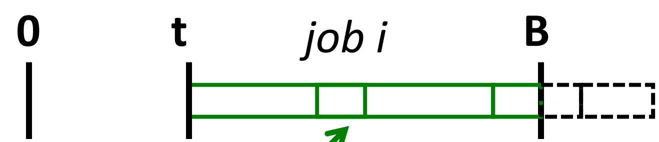
$$x \ge 0.$$

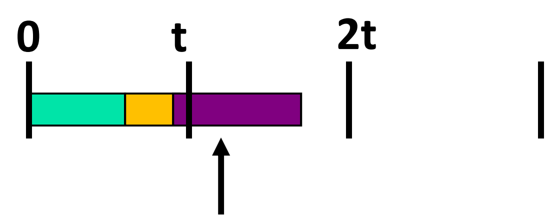
T_{it} := indicator OPT starts job i @ time t

Consider LP solution $x_{it} = \Pr[T_{it}=1]$

Claim 1: $\sum_{t=0}^{B} w_{it} \cdot x_{it} = OPT$

E[reward from i | $T_{it}=1$] = $E[R_i \cdot \mathbf{1}_{S_i \le B-t}] = w_{it}$





Claim 2: $\sum_{i=1}^{n} \min\{S_i, t\} \cdot \sum_{s=0}^{t} T_{is} \le 2t$

In any decision path, *at most one* job running @ time t

Non-Adaptive Algorithm

1) Solve time-indexed LP relaxation poly(n,B) time

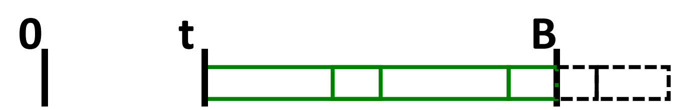
2) For each i, set $D_i = \begin{cases} t & \text{w.p. } x_{it}/4 \text{ for } t \in [\mathbf{B}] \\ \infty & \text{otherwise} \end{cases}$ (valid since $\sum_t x_{it} \leq 1$)

3) Place jobs $i_1, i_2, \cdots i_n$ by non-decreasing D_is

Theorem: 8-approx. for correlated stochastic knapsack

Lemma: $\Pr[\text{i starts by t} \mid D_i = t] \geq ½$ for all $i \in [n]$, $t < \infty$

$$\text{ALG} \geq \sum_i \sum_t \Pr[\text{i starts by t} \wedge D_i = t] \cdot w_{it}$$

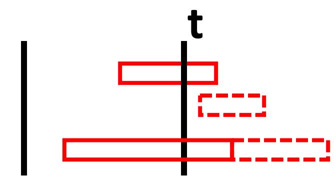


$$= \sum_i \sum_t \Pr[D_i = t] \cdot \Pr[\text{i starts by t} \mid D_i = t] \cdot w_{it} \geq \sum_i \sum_t \frac{x_{it}}{8} \cdot w_{it}$$

Analysis

Lemma: Pr[i doesn't start by t | $D_i=t$] $\leq$ ½ for all $i \in [n]$, $t < \infty$

Fix job i, time $t < \infty$

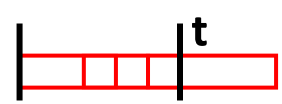


$\mathbf{Z_j} := \mathbf{1}_{D_j \leq t} \cdot \min(S_j, t)$

$E[\mathbf{Z_j}] = \sum_{s=0}^{t} \Pr[D_j=s] \cdot E[\min(S_j, t) \mid D_j=s] = \sum_{s=0}^{t} e_{jt} \cdot x_{js}/4$

indep.

"i doesn't start by t | $D_i=t$" $\Rightarrow \sum_{\mathbf{j \prec i}} S_j > t \Rightarrow \sum_{j \neq i} \mathbf{Z_j} \geq t$



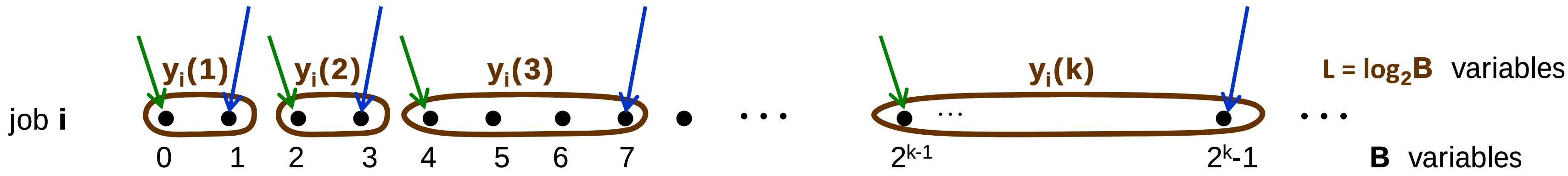
So Pr[ ] $\leq E[\sum_{j \neq i} \mathbf{Z_j}] / t = \sum_{i=1}^{n} \sum_{s=0}^{t} e_{it} \cdot x_{is} / 4t \leq$ ½

Polynomial Time Algorithm for CSK

- Time indexed LP has *pseudo-polynomial* size
  - Input size = O(n · log B · support)
  - LP size = n·B

- Succinct LP with size O(n · log B · support)
  - Group x_{it} variables
  - Lose constant factor more

Succinct LP for CSK



LP_t

$$\max \sum_{i=1}^{n} \sum_{t=0}^{B} w_{it} \cdot x_{it}$$

$$\sum_{i=1}^{n} e_{it} \cdot \sum_{s=0}^{t} x_{is} \leq 2t, \forall t \in [B]$$

$$\sum_{t=0}^{B} x_{it} \leq 1, \quad \forall i \in [n]$$

$$\mathbf{x} \geq 0.$$

LP_p

$$\max \sum_{i=1}^{n} \sum_{k=1}^{L} w_i(k) \cdot y_i(k)$$

$$\sum_{i=1}^{n} e_i(k) \cdot \sum_{h=0}^{k} y_i(h) \leq 2(2^k-1), \forall k \in [L]$$

$$\sum_{k=1}^{L} y_i(k) \leq 1, \quad \forall i \in [n]$$

$$\mathbf{y} \geq 0.$$

$w_i(k) = w_{i,2^{k-1}}$ and $e_i(k) = e_{i,2^{k-1}}$

Theorem: $LP_p \geq LP_t \geq OPT$

Modified Rounding

1) Solve LP_p (poly-time)

2) For each i, set $D_i = \begin{cases} 0 & \text{w.p. } y_i(1)/8 \\ 2^{k-1} & \text{w.p. } y_i(k)/8 \text{ for } k=2,\cdots,L \\ \infty & \text{otherwise} \end{cases}$

3) Place jobs $i_1, i_2, \cdots i_n$ by non-decreasing D_is

Theorem: 16-approx. for correlated stochastic knapsack

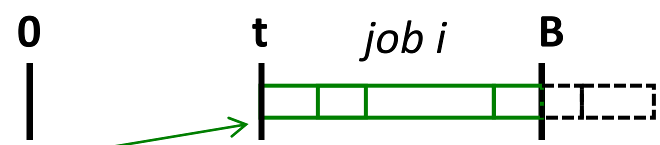
Lemma 3': $\Pr[i \text{ starts by } t \mid D_i=t] \geq ½$ for all $i \in [n]$, $t<\infty$

Outline

1. Stochastic knapsack (SK) basics ✓
2. Non-adaptive algorithm for SK ✓
3. Correlated stochastic knapsack (CSK) ✓
   - Non-adaptive algorithm
4. **Adaptive algorithm for CSK**
5. Extensions

New LP Relaxation for CSK

[Ma '14]

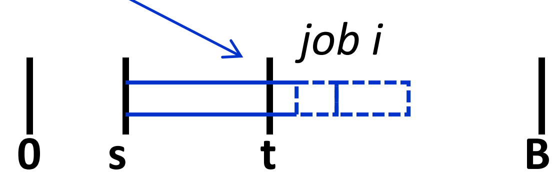


$$\max \ \sum_{i=1}^{n} \sum_{t=0}^{B} w_{it} \cdot x_{it}$$

$$\text{s.t.} \ \sum_{i=1}^{n} \sum_{s=0}^{t} x_{is} \cdot \Pr[S_i > t\text{-}s] \le 1, \ \forall\, t \in [B]$$

$$\sum_{t=0}^{B} x_{it} \le 1, \quad \forall\, i \in [n]$$

$$x \ge 0.$$



Theorem: LP* $\ge$ OPT.

- Assume size $S_i \in \{1, 2, \cdots B\}$ (for simplicity)
- LP size pseudo-polynomial O(nB)

Adaptive Algorithm

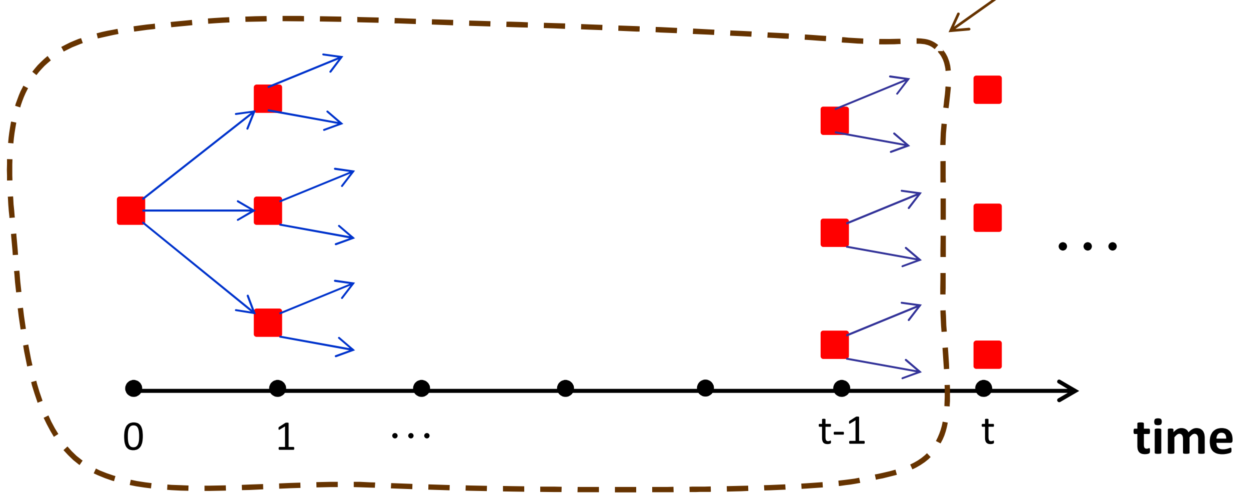
For t = 0, 1, ⋯ B do:

1. If some job is running, skip. Else:
2. Compute **F(i,t)** = Pr[job i can be started at t], $\forall$ i $\in$ [n]

   job i not already started $\leq$ t-1

   no running job at time t

   depends on ALG $\leq$ t-1
3. Start each remaining job i w.p. $\frac{x_{it}}{2 \cdot F(i,t)}$



Analysis Outline

*Assume **F(i,t)** computed exactly at each step*

Inductive Lemma: At each time t,

a) Pr[job i starts @ t] = $x_{it} / 2$

b) $F(i,t) \geq \frac{1}{2} \sum_{j=1}^{n} x_{jt}$

for all jobs i $\in$ [n]

(a) $\Rightarrow$ $E[ALG] = \sum_{i=1}^{n} \sum_{t=0}^{B} w_{it} \cdot \frac{x_{it}}{2} = \frac{LP^*}{2}$

(b) $\Rightarrow$ ALG well-def as $\sum_{j=1}^{n} \frac{x_{jt}}{2 \cdot F(j,t)} \leq \sum_{j=1}^{n} \frac{x_{jt}}{2 \cdot \min_i F(i,t)} \leq 1$

Theorem: 2-approximation algorithm for CSK

Pseudo-poly time **Hypothetical!**

Analysis (induction)

Inductive Lemma: At each time t,

a) $\Pr[\text{job } i \text{ starts @ } t] = x_{it} / 2$

b) $F(i,t) \geq ½ \sum_{j=1}^{n} x_{jt}$

for all jobs $i \in [n]$

$1-F(i,t) = \Pr[\text{job } i \text{ can not start @ } t]$

$\leq \Pr[\text{job } i \text{ started} \leq t-1] + \Pr[\text{some job running @ } t]$

Lem$_{t-1}$ $\leq ½ \sum_{s=0}^{t-1} x_{is} + ½ \sum_{s=0}^{t-1} \sum_{j=1}^{n} x_{js} \cdot \Pr[S_j > t-s]$

$= ½ \sum_{s=0}^{t-1} x_{is} + ½ \left[\sum_{s=0}^{t} \sum_{j=1}^{n} x_{js} \cdot \Pr[S_j > t-s] - \sum_{j=1}^{n} x_{jt} \cdot \Pr[S_j > 0]\right]$

LP constr $\leq ½ + ½ \left[1 - \sum_{j=1}^{n} x_{jt} \cdot \Pr[S_j > 0]\right] = 1 - ½ \sum_{j=1}^{n} x_{jt}$ $\Rightarrow$ *(b)*

$\Rightarrow \Pr[\text{job } i \text{ starts @ } t] = F(i,t) \cdot \dfrac{x_{it}}{2 \cdot F(i,t)}$ $\Rightarrow$ *(a)*

Actual Algorithm

- At each time t, estimate F(i,t) by sampling
  - poly(n,B) indep. samples
- Standard analysis
- Prune LP solution by ignoring times t : $\sum_{j=1}^{n} x_{jt} < 1/B^2$
  - *Lose (1-o(1)) factor*

Inductive Lemma' : At each time t,

a) $x_{it} / 2(1+\epsilon)^2 \leq$ Pr[job i starts @ t] $\leq x_{it} / 2$

b) $F(i,t) \geq ½ \sum_{j=1}^{n} x_{jt}$

Theorem: (2+ϵ)-approximation algorithm for CSK

Pseudo-poly time

Outline

1. Stochastic knapsack (SK) basics ✓
2. Non-adaptive algorithm for SK ✓

   8-approx
3. Correlated stochastic knapsack (CSK)

   Non-adaptive algorithm ✓

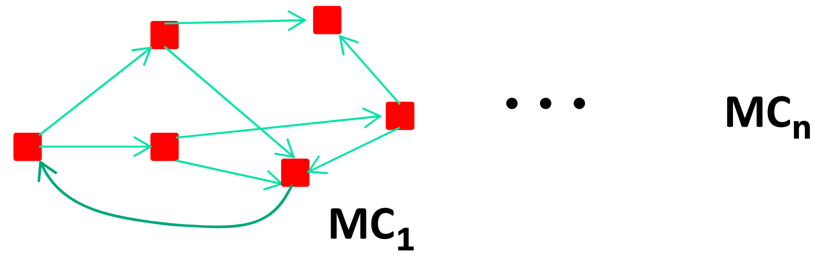
   16-approx poly-time
4. Adaptive algorithm for CSK ✓

   (2+ϵ)-approx pseudo-poly-time
5. **Extensions**

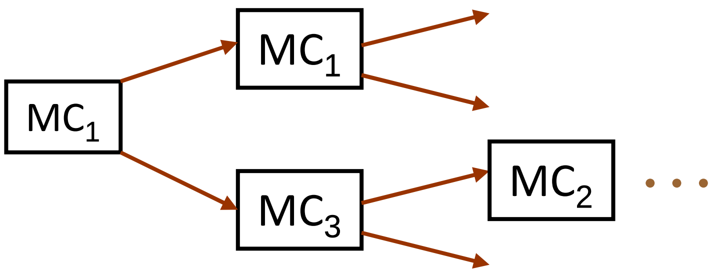
Budgeted Multi-Armed Bandits

- Collection of Markov chains

  Each play yields reward



- Limited plays (cardinality/cost constraint)



- 2-approx "non-adaptive" algorithm for *Martingale rewards*

  [Guha Munagala '07]

- O(1)-approx "adaptive" algorithm in general

  [Gupta Krishnaswamy Molinaro Ravi '12]

Stochastic Orienteering

Jobs at vertices of metric

- random **size** S_i
- deterministic **distances**

Bound B on total allowed time

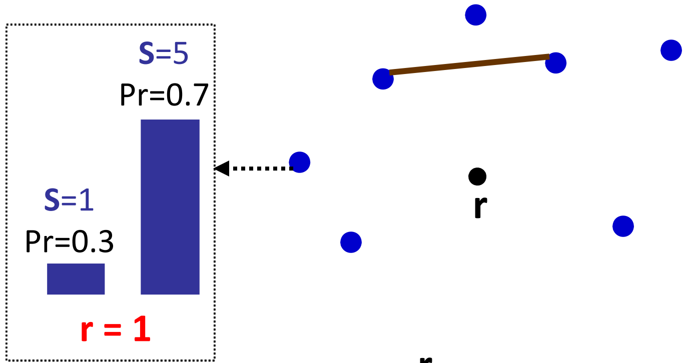
- **travel-time plus size**

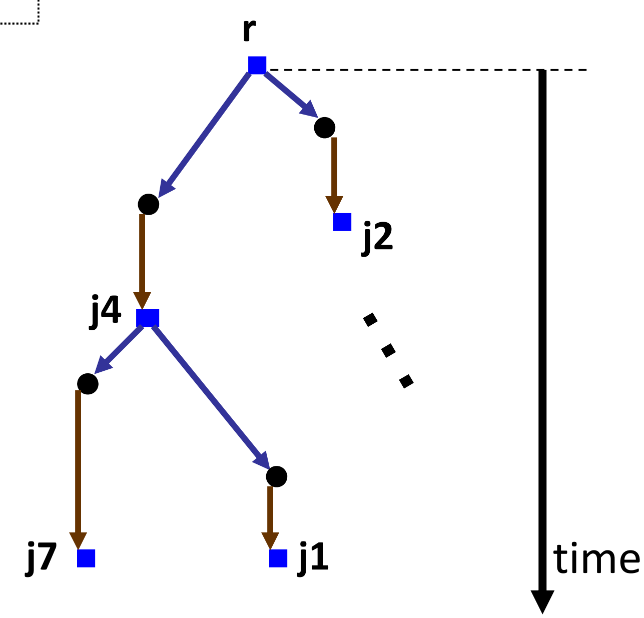
Find policy for visiting and running jobs

- maximize expected reward
- starts at root r, non-preemptive

O(loglog B)-approximation **[Gupta Krishnaswamy N. Ravi '12]**

$\Omega(\sqrt{\text{loglogB}})$ adaptivity gap **[Bansal N. '14]**





Open Questions

- Stochastic knapsack
  - Is there a PTAS? Or hardness of approximation?
  - Precise adaptivity gap?
  - Poly-time 2-approx for CSK?
- Stochastic orienteering
  - Better adaptive algorithm?
  - Tight adaptivity gap?

References

[1] B. Dean, M. Goemans, J. Vondrak, Approximating the stochastic knapsack problem: the benefit of adaptivity, Math. of OR, 2008.

[2] A. Gupta, R. Krishnaswamy, M. Molinaro, R. Ravi, Approximation algorithms for correlated knapsacks and non-martingale bandits, FOCS, 2011.

[3] W. Ma, Improvements and generalizations of stochastic knapsack and multi-armed bandit approximation algorithms, SODA, 2014.

[4] S. Guha, K. Munagala, Approximation algorithms for budgeted learning, STOC, 2007

[5] A. Gupta, R. Krishnaswamy, V. Nagarajan, R. Ravi, Approximation algorithms for stochastic orienteering, SODA, 2012.

[6] N. Bansal, V. Nagarajan, On the adaptivity gap of stochastic orienteering, IPCO, 2014.

[7] A. Bhalgat, A. Goel, S. Khanna, Improved approximation results for stochastic knapsack problems, SODA, 2011.

[8] J. Li, W. Yuan, Stochastic combinatorial optimization via Poisson approximation, STOC, 2013.