PANDORA'S PROBLEM WITH NONOBLIGATORY INSPECTION: OPTIMAL STRUCTURE AND A PTAS

Hedyeh Beyhaghi (CMU), Linda Cai (Princeton University)

Presented by Linda Cai

PANDORA BOX PROBLEM WITH NONOBLIGATORY INSPECTION: HARDNESS AND APPROXIMATION SCHEME.

Hu Fu (Shanghai University of Finance and Economics),
Jiawei Li (University of Texas at Austin),
Daogao Liu (University of Washington)

CONCURRENT PRESENTATION AT STOC 2023

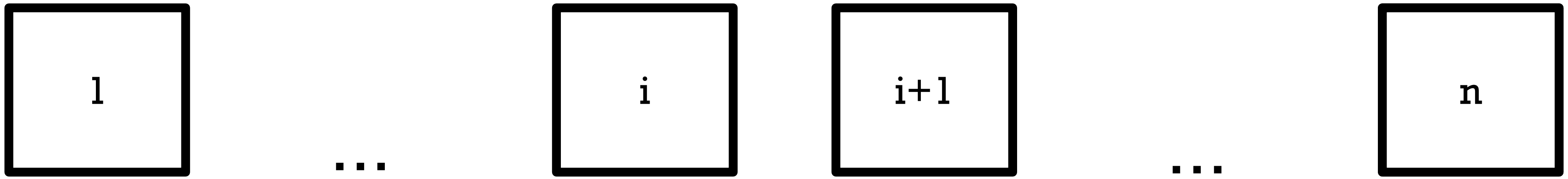
SEARCH IS COSTLY

SEARCH IS COSTLY

SEARCH IS COSTLY

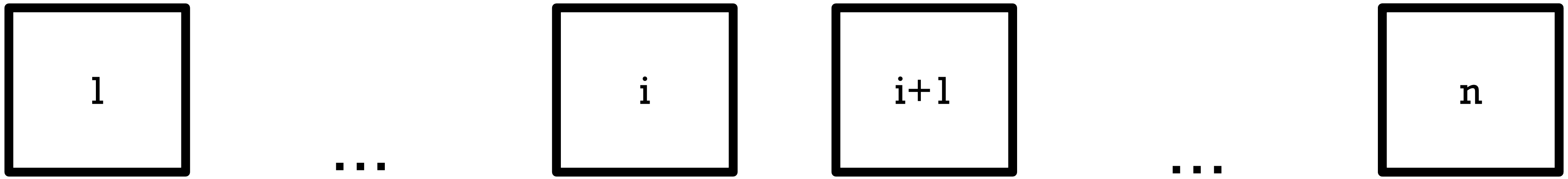
PANDORA'S BOX PROBLEM

(Introduced by Weitzman79)



PANDORA'S BOX PROBLEM

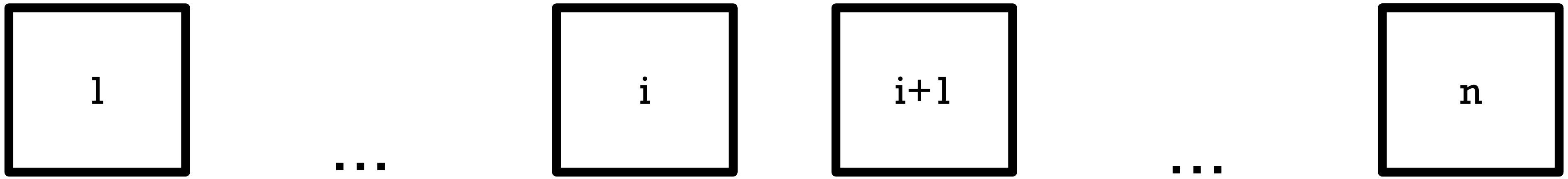
(Introduced by Weitzman79)



Value: $v_i \sim D_i$

PANDORA'S BOX PROBLEM

(Introduced by Weitzman79)

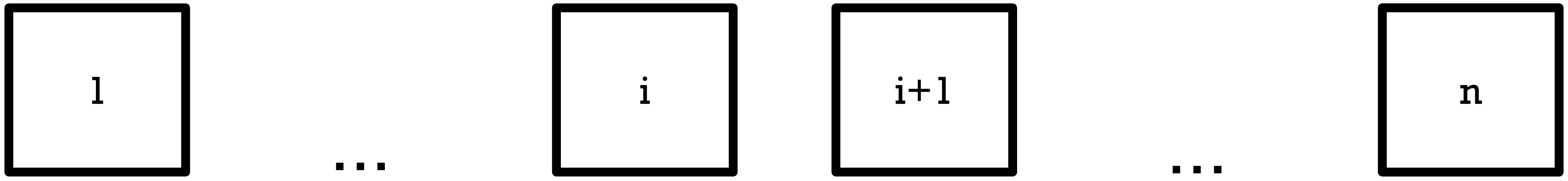


Value: $v_i \sim D_i$

Cost: c_i

PANDORA'S BOX PROBLEM

(Introduced by Weitzman79)



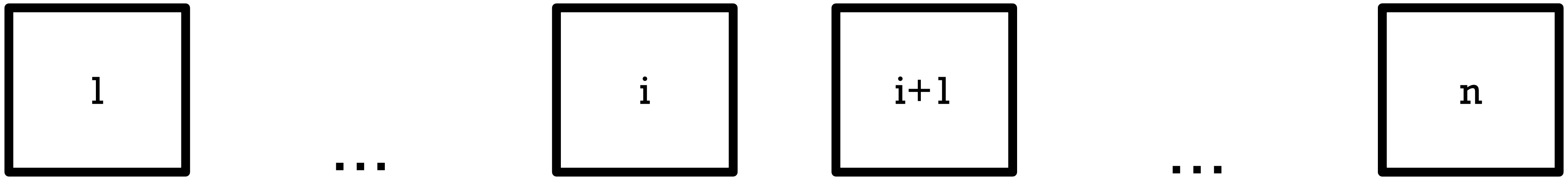
Value: $v_i \sim D_i$

Cost: c_i

The agent can inspect boxes in any order they like, and their goal is to maximize their Expected Utility = E[value of selected box – sum of inspection costs]

PANDORA'S BOX PROBLEM

(Introduced by Weitzman79)



Value: $v_i \sim D_i$

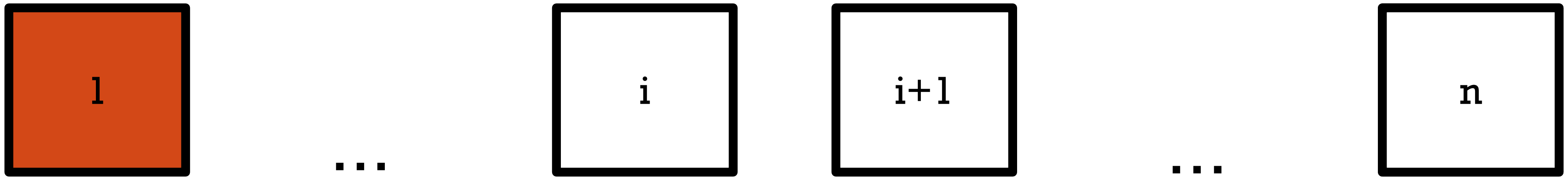
Cost: c_i

The agent can inspect boxes in any order they like, and their goal is to maximize their Expected Utility = E[value of selected box – sum of inspection costs]

The agent must inspect a box before selecting it

PANDORA'S BOX PROBLEM

(Introduced by Weitzman79)



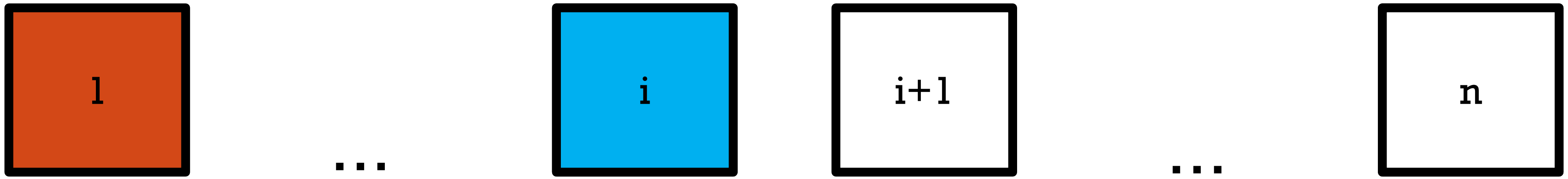
value = 2
cost = 1

The agent can inspect boxes in any order they like, and their goal is to maximize their Expected Utility = E[value of selected box – sum of inspection costs]

The agent must inspect a box before selecting it

PANDORA'S BOX PROBLEM

(Introduced by Weitzman79)



value = 2
cost = 1

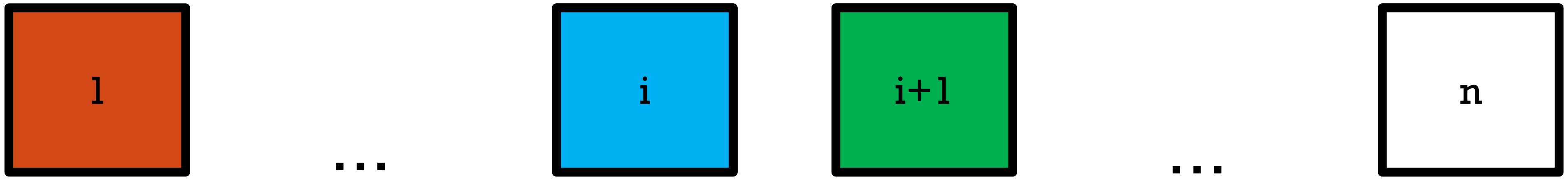
value = 8
cost = 2

The agent can inspect boxes in any order they like, and their goal is to maximize their Expected Utility = E[value of selected box – sum of inspection costs]

The agent must inspect a box before selecting it

PANDORA'S BOX PROBLEM

(Introduced by Weitzman79)



value = 2
cost = 1

value = 8
cost = 2

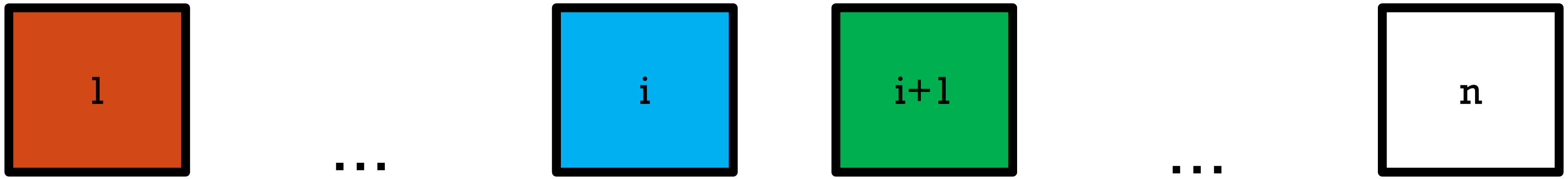
value = 7
cost = 2

The agent can inspect boxes in any order they like, and their goal is to maximize their Expected Utility = E[value of selected box – sum of inspection costs]

The agent must inspect a box before selecting it

PANDORA'S BOX PROBLEM

(Introduced by Weitzman79)



value = 2
cost = 1

value = 8
cost = 2

value = 7
cost = 2

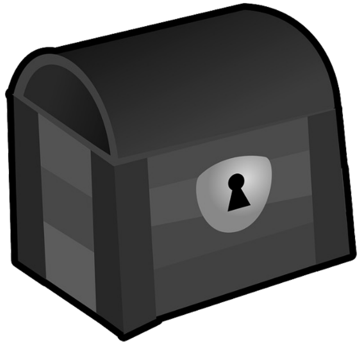
Utility = max(value) – sum(cost) = max(2, 8, 7) – (1 + 2 + 2) = 3

The agent can inspect boxes in any order they like, and their goal is to maximize their Expected Utility = E[value of selected box – sum of inspection costs]

The agent must inspect a box before selecting it

PANDORA'S BOX: OPTIMAL POLICY [WEITZ79]

PANDORA'S BOX: OPTIMAL POLICY [WEITZ79]



Value: $v_i \sim D_i$

Cost: c_i

Reservation value: σ_i such that $\mathrm{E}[(v_i - \sigma_i)^+] = c_i$

PANDORA'S BOX: OPTIMAL POLICY [WEITZ79]

Value: $v_i \sim D_i$

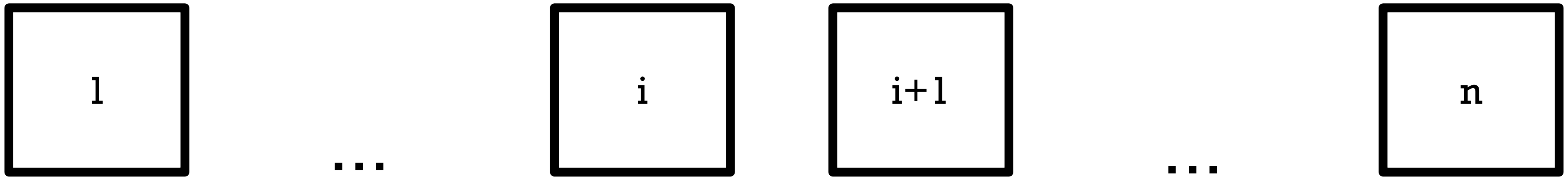
Cost: c_i

Reservation value: σ_i such that $\mathrm{E}[(v_i - \sigma_i)^+] = c_i$

Maximum value seen so far: a →

a < reservation value ⇔ opening the box has positive marginal utility gain

PANDORA'S BOX: OPTIMAL POLICY [WEITZ79]



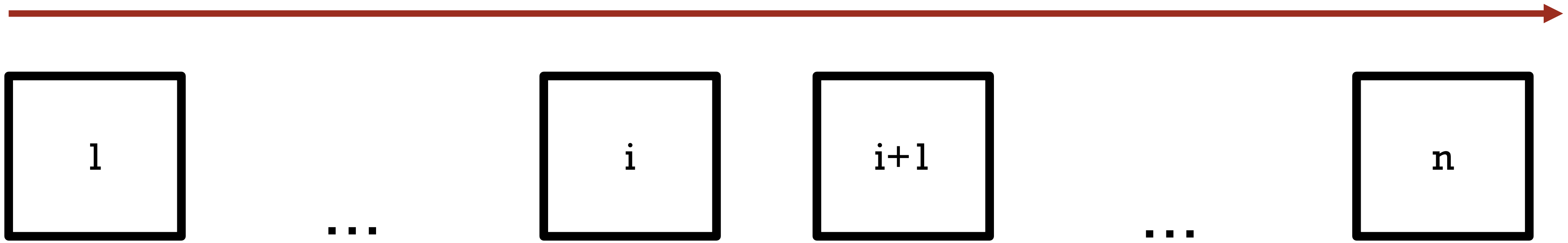
Value: $v_i \sim D_i$

Cost: c_i

Reservation value: σ_i such that $\mathrm{E}[(v_i - \sigma_i)^+] = c_i$

PANDORA'S BOX: OPTIMAL POLICY [WEITZ79]

Reorder in decreasing value of **reservation value** $\sigma: \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$



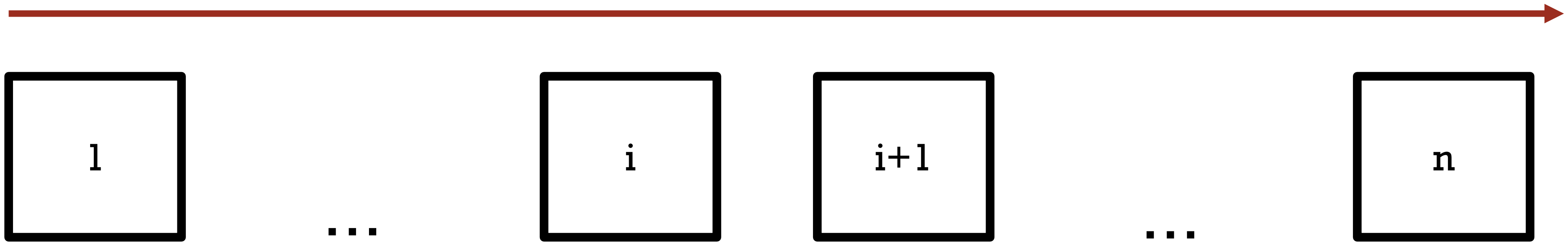
Value: $v_i \sim D_i$

Cost: c_i

Reservation value: σ_i such that $\mathrm{E}[(v_i - \sigma_i)^+] = c_i$

PANDORA'S BOX: OPTIMAL POLICY [WEITZ79]

Reorder in decreasing value of **reservation value** $\sigma : \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$



Value: $v_i \sim D_i$

Cost: c_i

Reservation value: σ_i such that $\mathrm{E}[(v_i - \sigma_i)^+] = c_i$

Agent opens the boxes in sequential order until position k where $\max_{i<k} v_i \geq \sigma_k$,
in which case the agents stops and returns the maximum value they have seen so far.

PANDORA'S BOX: IS INSPECTION NECESSARY?

International student: campus visits are too costly and time consuming

GIVING THE AGENT THE FREEDOM OF CHOICE

GIVING THE AGENT THE FREEDOM OF CHOICE

- **[Guha, Munagala and Sarkar 2008]**
  Information Acquisition and Exploitation in Multichannel Wireless Networks.

GIVING THE AGENT THE FREEDOM OF CHOICE

- **[Guha, Munagala and Sarkar 2008]**
  Information Acquisition and Exploitation in Multichannel Wireless Networks.
- **[Chang and Liu 2009]**
  Optimal channel probing and transmission scheduling for opportunistic spectrum access.

GIVING THE AGENT THE FREEDOM OF CHOICE

- **[Guha, Munagala and Sarkar 2008]**
  Information Acquisition and Exploitation in Multichannel Wireless Networks.
- **[Chang and Liu 2009]**
  Optimal channel probing and transmission scheduling for opportunistic spectrum access.
- **[Attias, Krauthgamer, Levi and Shaposhnik 2017]**
  Stochastic selection problems with testing.

GIVING THE AGENT THE FREEDOM OF CHOICE

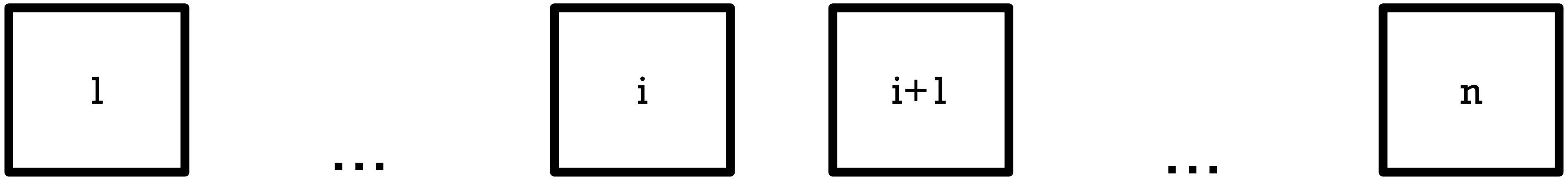
- **[Guha, Munagala and Sarkar 2008]**
  Information Acquisition and Exploitation in Multichannel Wireless Networks.
- **[Chang and Liu 2009]**
  Optimal channel probing and transmission scheduling for opportunistic spectrum access.
- **[Attias, Krauthgamer, Levi and Shaposhnik 2017]**
  Stochastic selection problems with testing.
- **[Doval 2018] (direct extension of Weitzman's model)**
  Whether or not to open Pandora's box.

GIVING THE AGENT THE FREEDOM OF CHOICE

- **[Guha, Munagala and Sarkar 2008]**
  Information Acquisition and Exploitation in Multichannel Wireless Networks.
- **[Chang and Liu 2009]**
  Optimal channel probing and transmission scheduling for opportunistic spectrum access.
- **[Attias, Krauthgamer, Levi and Shaposhnik 2017]**
  Stochastic selection problems with testing.
- **[Doval 2018] (direct extension of Weitzman's model)**
  Whether or not to open Pandora's box.

PANDORA'S BOX PROBLEM WITH NON-OBLIGATORY INSPECTION

PANDORA'S BOX WITH NON-OBLIGATORY INSPECTION (PNOI*)



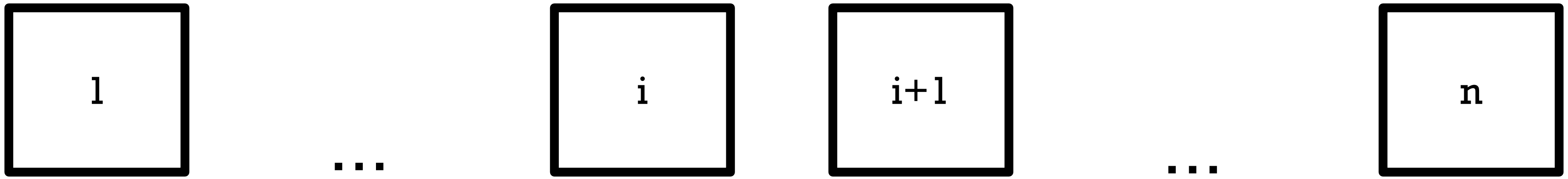
Value: $v_i \sim D_i$

Cost: c_i

The agent can inspect boxes in any order they like, and their goal is to maximize their Expected Utility = E[Value of selected box – sum of inspection costs]

* The acronym "PNOI" is first used in an earlier version of Fu Li and Liu 2023

PANDORA'S BOX WITH NON-OBLIGATORY INSPECTION (PNOI*)



Value: $v_i \sim D_i$

Cost: c_i

The agent can inspect boxes in any order they like, and their goal is to maximize their Expected Utility = E[Value of selected box – sum of inspection costs]

The agent can either inspect a box, or claim the box closed without inspection

* The acronym "PNOI" is first used in an earlier version of Fu Li and Liu 2023

PNOI: WHAT IS DIFFERENT

- Weitzman's policy is no longer optimal

WEITZMAN'S POLICY IS NOT OPTIMAL

Box 1

$$v_1 = \begin{cases} \frac{1}{\epsilon} \text{ w.p. } \epsilon \\ 0 \text{ w.p. } 1-\epsilon \end{cases}$$

$$c_1 = 1/2$$

Box 2

$$v_2 = \begin{cases} 1 \text{ w.p. } ½ \\ 0 \text{ w.p. } ½ \end{cases}$$

$$c_2 = \epsilon$$

WEITZMAN'S POLICY IS NOT OPTIMAL

Box 1

$$v_1 = \begin{cases} \frac{1}{\epsilon} \text{ w.p. } \epsilon \\ 0 \text{ w.p. } 1-\epsilon \end{cases}$$

$c_1 = 1/2$ $\sigma_1 = \frac{1}{2\epsilon}$

Box 2

$$v_2 = \begin{cases} 1 \text{ w.p. } ½ \\ 0 \text{ w.p. } ½ \end{cases}$$

$c_2 = \epsilon$ $\sigma_2 = 1 - 2\epsilon$

WEITZMAN'S POLICY IS NOT OPTIMAL

Box 1

$$v_1 = \begin{cases} \frac{1}{\epsilon} & \text{w.p. } \epsilon \\ 0 & \text{w.p. } 1-\epsilon \end{cases}$$

$c_1 = 1/2$ $\sigma_1 = \frac{1}{2\epsilon}$

Box 2

$$v_2 = \begin{cases} 1 & \text{w.p. } ½ \\ 0 & \text{w.p. } ½ \end{cases}$$

$c_2 = \epsilon$ $\sigma_2 = 1 - 2\epsilon$

Weitzman's policy: open box 1 first, then open box 2 only when the value of box 1 is 0

WEITZMAN'S POLICY IS NOT OPTIMAL

Box 1

$$v_1 = \begin{cases} \frac{1}{\epsilon} & \text{w.p. } \epsilon \\ 0 & \text{w.p. } 1-\epsilon \end{cases}$$

Box 2

$$v_2 = \begin{cases} 1 & \text{w.p. } ½ \\ 0 & \text{w.p. } ½ \end{cases}$$

$c_1 = 1/2$ $\sigma_1 = \frac{1}{2\epsilon}$ $c_2 = \epsilon$ $\sigma_2 = 1 - 2\epsilon$

Weitzman's policy: open box 1 first, then open box 2 only when the value of box 1 is 0

Optimal policy in non-obligatory inspection: open box 2 first,

$v_2 = 0 \rightarrow$ claim box 1 closed

$v_2 = 1 \rightarrow$ open box 1

WEITZMAN'S POLICY IS NOT OPTIMAL

Box 1

$$v_1 = \begin{cases} \frac{1}{\epsilon} & \text{w.p. } \epsilon \\ 0 & \text{w.p. } 1-\epsilon \end{cases}$$

Box 2

$$v_2 = \begin{cases} 1 & \text{w.p. } ½ \\ 0 & \text{w.p. } ½ \end{cases}$$

$c_1 = 1/2$ $\quad \sigma_1 = \frac{1}{2\epsilon}$ $\quad c_2 = \epsilon$ $\quad \sigma_2 = 1 - 2\epsilon$

Weitzman's policy: open box 1 first, then open box 2 only when the value of box 1 is 0

Agent Utility $= 1 - \frac{3\epsilon}{2} + \epsilon^2$

Optimal policy in non-obligatory inspection: open box 2 first,

$v_2 = 0 \rightarrow$ claim box 1 closed

$v_2 = 1 \rightarrow$ open box 1

Agent Utility $= \frac{5}{4} - \frac{3\,\epsilon}{2}$

PNOI: WHAT IS DIFFERENT

- Weitzman's policy is no longer optimal
- Adaptivity is required in the optimal policy

ADAPTIVITY IS REQUIRED (DOVAL 2018)

Box 1

Box 2

Box 3

$$v_1 = \begin{cases} \frac{1}{\epsilon} & \text{w.p. } \epsilon \\ 0 & \text{w.p. } 1-\epsilon \end{cases}$$

$c_1 = 1/2$

$$v_2 = \begin{cases} 1 & \text{w.p. } ½ \\ 0 & \text{w.p. } ½ \end{cases}$$

$c_2 = \epsilon$

$$v_3 = \begin{cases} \frac{1}{\epsilon^2} & \text{w.p. } \epsilon^2 \\ 1 & \text{w.p. } \epsilon \\ 0 & \text{w.r.p} \end{cases}$$

$c_3 = 1/2$

Optimal policy in non-obligatory inspection: open box 3 first,

$v_3 = 1/\epsilon^2 \rightarrow$ stop

$v_3 = 1 \rightarrow$ open box 1 first

$v_3 = 0 \rightarrow$ open box 2 first

PNOI: WHAT IS DIFFERENT

- Weitzman's policy is no longer optimal
- Adaptivity is required in the optimal policy
- [Fu Li and Liu 2022] NP-Hardness

Theorem [Fu Li and Liu Arxiv Preprint 2022*]: Finding the optimal policy for the pandora box with non- obligatory inspection problem is NP-hard.

*An updated version of Fu Li and Liu is accepted to STOC 2023 together with our paper.

PNOI: CAN WE SAY ANYTHING ABOUT THE OPTIMAL POLICY?

PNOI: CAN WE SAY ANYTHING ABOUT THE OPTIMAL POLICY?

Related NP-Hard problem with a structurally interesting optimal policy:

Theorem [Agrawal, Sethuraman and Zhang 2020]:

1) Finding the optimal policy for the free order prophet inequality problem is NP-hard.
2) The optimal policy is **non-adaptive**.

PNOI: CAN WE SAY ANYTHING ABOUT THE OPTIMAL POLICY?

Related NP-Hard problem with a structurally interesting optimal policy:

Theorem [Agrawal, Sethuraman and Zhang 2020]:

1) Finding the optimal policy for the free order prophet inequality problem is NP-hard.
2) The optimal policy is **non-adaptive**.

We have just shown that for our problem adaptivity is required…

PNOI: CAN WE SAY ANYTHING ABOUT THE OPTIMAL POLICY?

Related NP-Hard problem with a structurally interesting optimal policy:

Theorem [Agrawal, Sethuraman and Zhang 2020]:

1) Finding the optimal policy for the free order prophet inequality problem is NP-hard.
2) The optimal policy is **non-adaptive**.

We have just shown that for our problem adaptivity is required…

Main Result 1*: the optimal policy for PNOI consists of **two phases**, where in each phase, the order of visiting boxes is pre-determined and nonadaptive.

*Also proven in an updated version of Fu Li and Liu (accepted to STOC 2023 jointly with our paper).

PNOI: STRUCTURE OF THE OPTIMAL POLICY

PNOI: STRUCTURE OF THE OPTIMAL POLICY

Definition: A backup box in a policy is a box that the policy sometimes claim closed without inspection.

PNOI: STRUCTURE OF THE OPTIMAL POLICY

Definition: A backup box in a policy is a box that the policy sometimes claim closed without inspection.

Box 1

$$v_1 = \begin{cases} \frac{1}{\epsilon} & \text{w.p. } \epsilon \\ 0 & \text{w.p. } 1-\epsilon \end{cases}$$

$c_1 = 1/2$

Box 2

$$v_2 = \begin{cases} 1 & \text{w.p. } ½ \\ 0 & \text{w.p. } ½ \end{cases}$$

$c_2 = \epsilon$

Optimal policy in non-obligatory inspection: open box 2 first,

$v_2 = 0 \rightarrow$ claim box 1 closed

$v_2 = 1 \rightarrow$ open box 1

PNOI: STRUCTURE OF THE OPTIMAL POLICY

Definition: A backup box in a policy is a box that the policy sometimes claim closed without inspection.

Box 1

$$v_1 = \begin{cases} \frac{1}{\epsilon} & \text{w.p. } \epsilon \\ 0 & \text{w.p. } 1-\epsilon \end{cases}$$

backup box

$c_1 = 1/2$

Box 2

$$v_2 = \begin{cases} 1 & \text{w.p. } ½ \\ 0 & \text{w.p. } ½ \end{cases}$$

not a backup box

$c_2 = \epsilon$

Optimal policy in non-obligatory inspection: open box 2 first,

$v_2 = 0 \rightarrow$ claim box 1 closed

$v_2 = 1 \rightarrow$ open box 1

PNOI: STRUCTURE OF THE OPTIMAL POLICY

Structural Theorem [Guha, Munagala and Sarkar 2008]

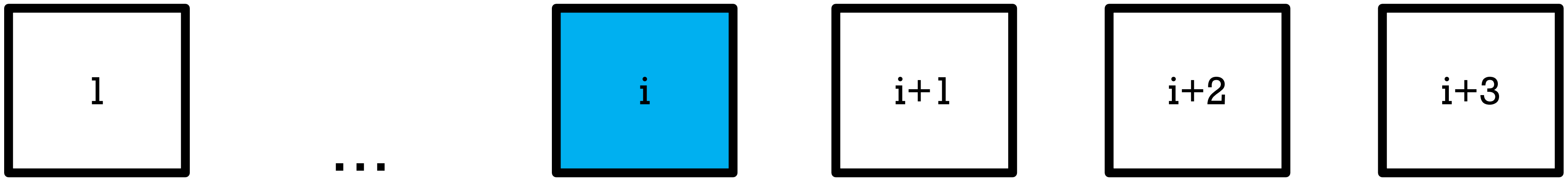
There exists an optimal policy for PNOI that has *at most one* back up box.

PNOI: STRUCTURE OF THE OPTIMAL POLICY

Structural Theorem [Guha, Munagala and Sarkar 2008]

There exists an optimal policy for PNOI that has *at most one* back up box.

Step one: select a backup box (if any)

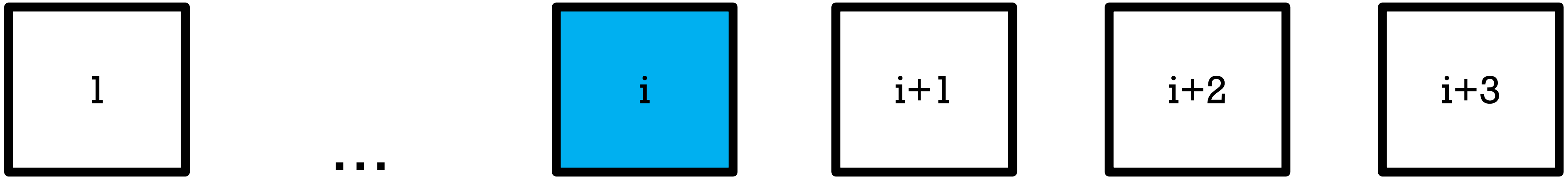


PNOI: STRUCTURE OF THE OPTIMAL POLICY

Structural Theorem [Guha, Munagala and Sarkar 2008]

There exists an optimal policy for PNOI that has *at most one* back up box.

Step one: select a backup box (if any)



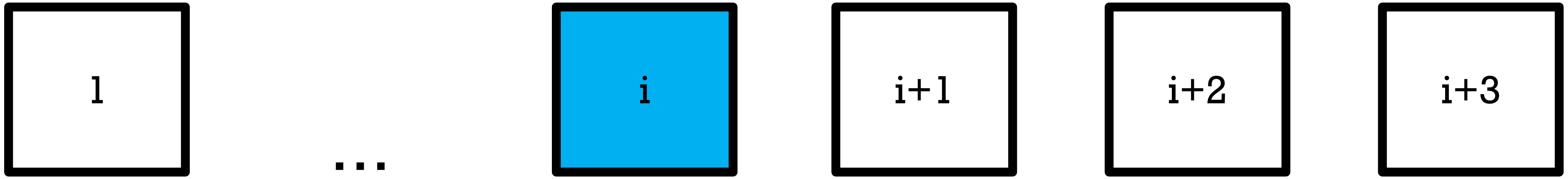
Step two: find optimal policy given that box i is the unique backup box

PNOI: STRUCTURE OF THE OPTIMAL POLICY

Structural Theorem [Guha, Munagala and Sarkar 2008]

There exists an optimal policy for PNOI that has *at most one* back up box.

Step one: select a backup box (if any)



Step two: find optimal policy given that box i is the unique backup box

Note: even after fixing the backup box, an adaptive policy could still have *exponential* number of branches, we are still not sure that PNOI is in NP

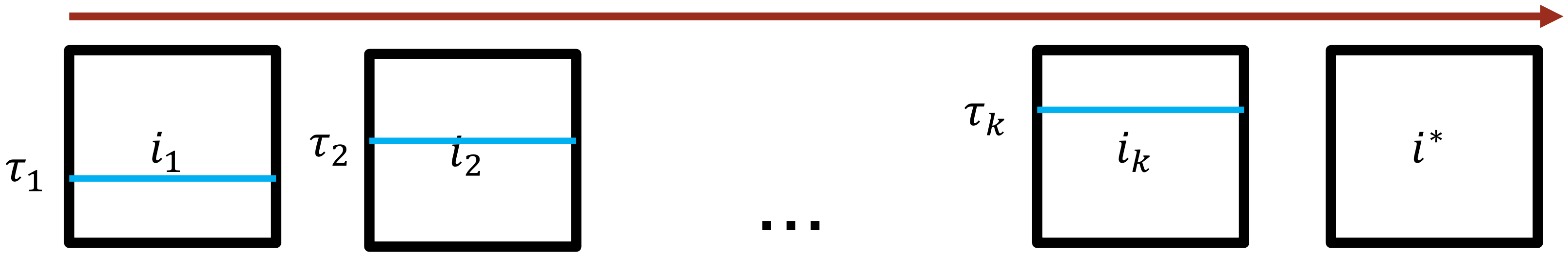
PNOI: STRUCTURE OF THE OPTIMAL POLICY

Main Result 1: there exists an optimal policy in the form of the following two-phase policy.

Policy selection:

Step one: select a back up box i^* (or choose no backup box)

Step two: fix an initial order of the boxes $(i_1, \cdots, i_k, i^*)$ and associated thresholds $(\tau_1, \cdots, \tau_k)$



PNOI: STRUCTURE OF THE OPTIMAL POLICY

Main Result 1: there exists an optimal policy in the form of the following two-phase policy.

Two-phase Policy:

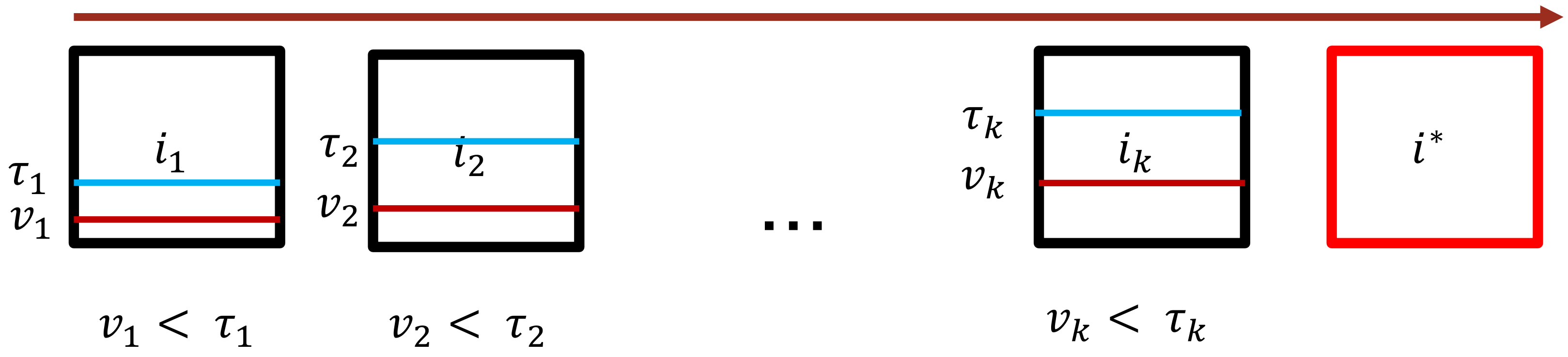
Phase one: while all seen values are below the threshold, keep opening boxes in initial order

PNOI: STRUCTURE OF THE OPTIMAL POLICY

Main Result 1: there exists an optimal policy in the form of the following two-phase policy.

Two-phase Policy:

Phase one: while all seen values are below the threshold, keep opening boxes in initial order



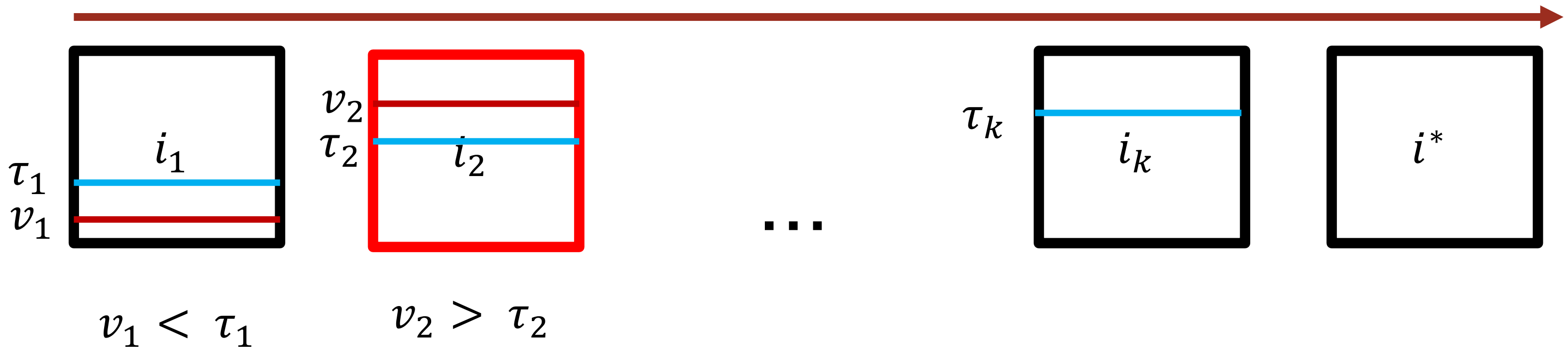
If we reach the end of the order, claim the backup box closed without inspection.

PNOI: STRUCTURE OF THE OPTIMAL POLICY

Main Result 1: there exists an optimal policy in the form of the following two-phase policy.

Two-phase Policy:

Phase one: while all seen values are below the threshold, keep opening boxes in initial order



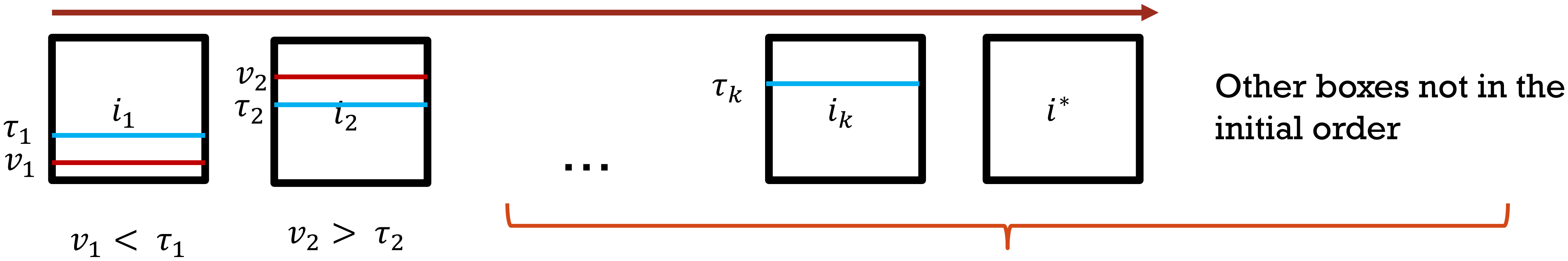
If we see a value above threshold, the policy enters phase two

PNOI: STRUCTURE OF THE OPTIMAL POLICY

Main Result 1: there exists an optimal policy in the form of the following two-phase policy.

Two-phase Policy:

Phase one: while all seen values are below the threshold, keep opening boxes in initial order



Reorder in decreasing value of σ: $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$

Phase two : Once $v_i > \tau_i$, run Weitzman's policy on remaining boxes with outside option v_i

PNOI: STRUCTURE OF THE OPTIMAL POLICY

Algorithm 1 Two-Phase Policy(InitialOrder=$i_1, \cdots, i_k, i^*$, Thresholds=$\tau_1, \cdots, \tau_k$)

for $j = 1, \cdots, k$ **do**
    Let $\mathcal{U}_j = \mathcal{M} \setminus \{i_1, \cdots, i_j\}$.
    Open box i_j, observe value v_{i_j} from the box.
    if $v_{i_j} > \tau_j$ **then**
        Run Weitzman's policy on remaining boxes from state $(\mathcal{U}_j, v_{i_j})$.
        return
    end if
end for
Claim box i^* closed.

PNOI: STRUCTURE OF THE OPTIMAL POLICY

Algorithm 1 Two-Phase Policy(InitialOrder=$i_1, \cdots, i_k, i^*$, Thresholds=$\tau_1, \cdots, \tau_k$)

1: **for** $j = 1, \cdots, k$ **do**
2: Let $\mathcal{U}_j = \mathcal{M} \setminus \{i_1, \cdots, i_j\}$.
3: Open box i_j, observe value v_{i_j} from the box.
4: **if** $v_{i_j} > \tau_j$ **then**
5: Run Weitzman's policy on remaining boxes from state $(\mathcal{U}_j, v_{i_j})$.
6: **return**
7: **end if**
8: **end for**
9: Claim box i^* closed.

Corollary: Pandora's box problem with non-obligatory inspection is in NP.

PROOF SKETCH OF OPTIMALITY

PROOF SKETCH OF OPTIMALITY

- Consider an optimal policy that uses at most one backup box.

PROOF SKETCH OF OPTIMALITY

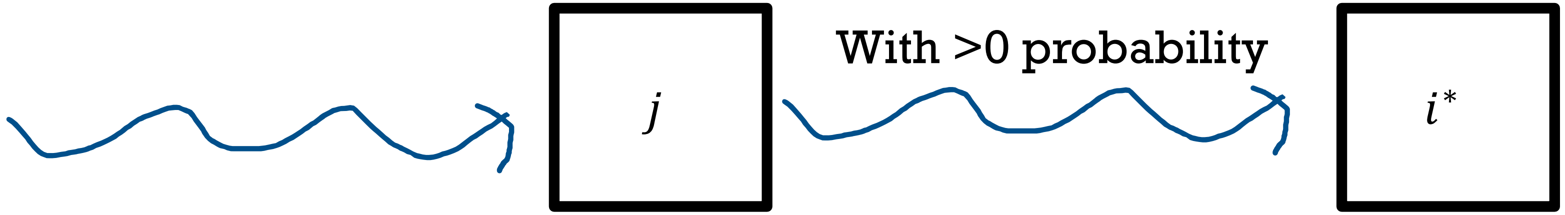
- Consider an optimal policy that uses at most one backup box.
- If the policy does not use backup box, then Weitzman policy is the optimal policy

PROOF SKETCH OF OPTIMALITY

- Consider an optimal policy that uses at most one backup box.
- If the policy does not use backup box, then Weitzman policy is the optimal policy
- If the policy uses backup box, let this box be i^*

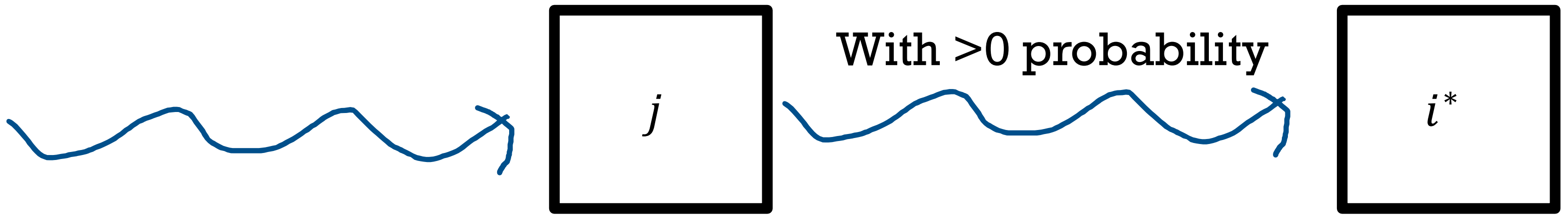
PROOF SKETCH OF OPTIMALITY

- Consider an optimal policy that uses at most one backup box.
- If the policy does not use backup box, then Weitzman policy is the optimal policy
- If the policy uses backup box, let this box be i^*



PROOF SKETCH OF OPTIMALITY

- Consider an optimal policy that uses at most one backup box.
- If the policy does not use backup box, then Weitzman policy is the optimal policy
- If the policy uses backup box, let this box be i^*

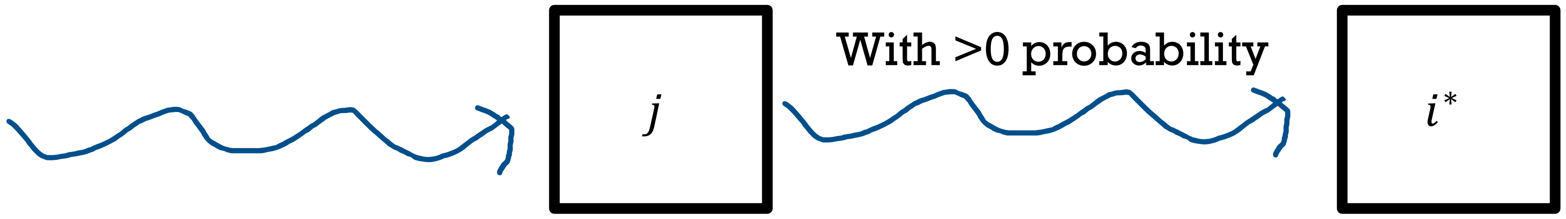


If after opening box j, we still may claim backup box closed with some probability, then:

- Either we see a value above v_j in the future
- Or we claim the box i^* closed

PROOF SKETCH OF OPTIMALITY

- Consider an optimal policy that uses at most one backup box.
- If the policy does not use backup box, then Weitzman policy is the optimal policy
- If the policy uses backup box, let this box be i^*



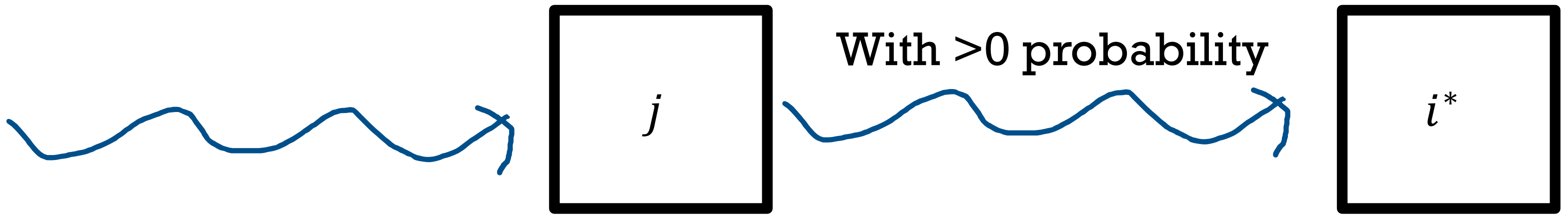
If after opening box j, we still may claim backup box closed with some probability, then:

- Either we see a value above v_j in the future
- Or we claim the box i^*closed

The value of v_j is *irrelevant* to the final value we select, can pretend $v_j = 0$

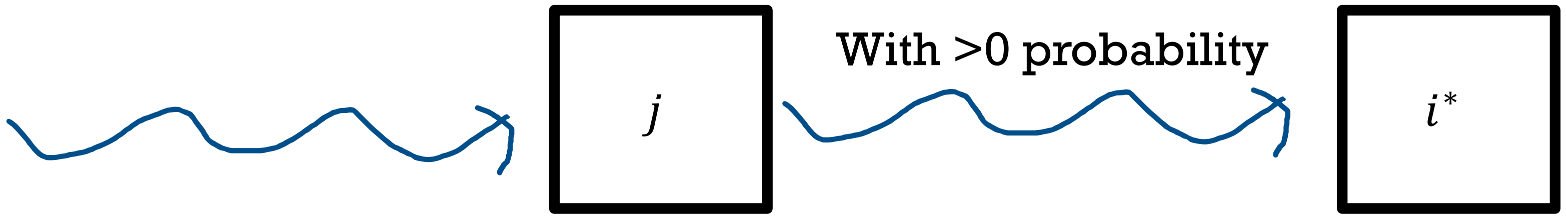
PROOF SKETCH OF OPTIMALITY

- Consider an optimal policy that uses at most one backup box.
- If the policy does not use backup box, then Weitzman policy is the optimal policy
- If the policy uses backup box, let this box be i^*



PROOF SKETCH OF OPTIMALITY

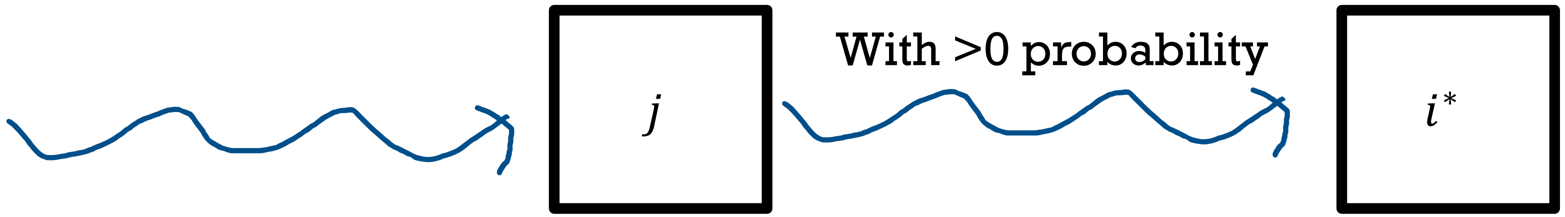
- Consider an optimal policy that uses at most one backup box.
- If the policy does not use backup box, then Weitzman policy is the optimal policy
- If the policy uses backup box, let this box be i^*



Let τ_j be the maximum value of box j where we still sometimes claim backup box closed

PROOF SKETCH OF OPTIMALITY

- Consider an optimal policy that uses at most one backup box.
- If the policy does not use backup box, then Weitzman policy is the optimal policy
- If the policy uses backup box, let this box be i^*



Let τ_j be the maximum value of box j where we still sometimes claim backup box closed

There is an optimal policy where:

- For $v_j \leq \tau_j$, we always take the same future actions
- For $v_j > \tau_j$, backup box is NEVER claimed closed, use Weitzman policy for future boxes

PNOI: POLYNOMIAL TIME APPROXIMATION SCHEME

Main result 2*: There exists a PTAS for the Pandora's box with nonobligatory inspection problem.

Our approach:

- Stochastic dynamic program formulated in [Fu Li and Xu 2018] has a PTAS
- We restrict the search space to finding approximately optimal two-phase policy
- Then we reduce our problem to stochastic dynamic program in [Fu Li and Xu 2018]

*Also proven in an updated version of Fu Li and Liu (accepted to STOC 2023 jointly with our paper).

PNOI: POLYNOMIAL TIME APPROXIMATION SCHEME

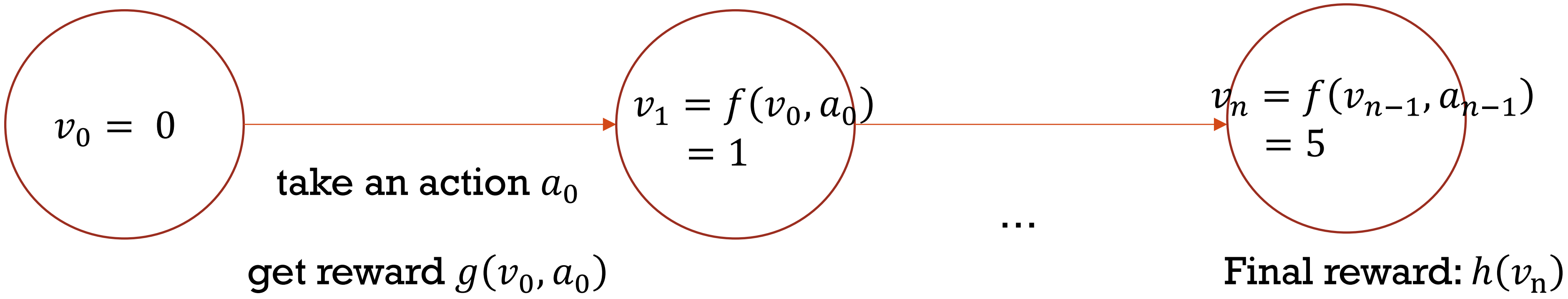
Main result 2*: There exists a PTAS for the Pandora's box with nonobligatory inspection problem.

Fu Li and Liu 2023:

- Stochastic dynamic program formulated in [Fu Li and Xu 2018] has a PTAS
- ~~We restrict the search space to finding approximately optimal two-phase policy~~
- Then we reduce our problem to stochastic dynamic program in [Fu Li and Xu 2018]

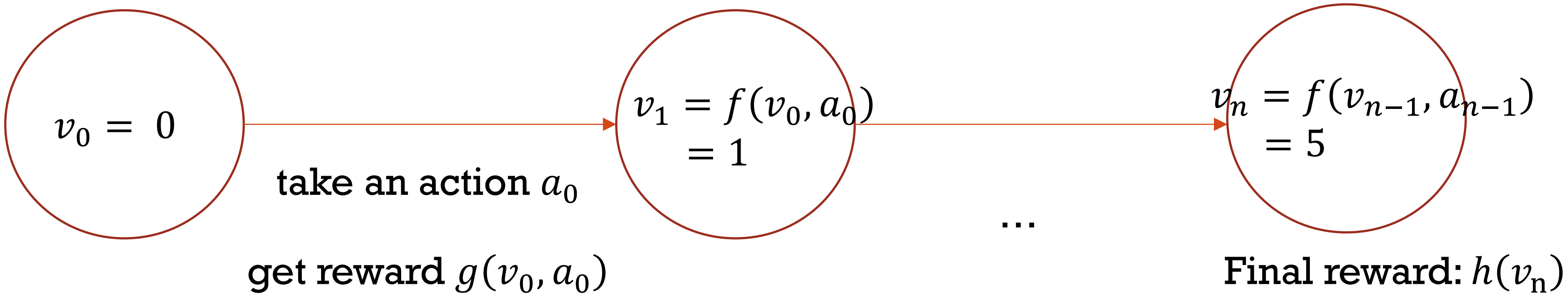
*Also proven in an updated version of Fu Li and Liu (accepted to STOC 2023 jointly with our paper).

STOCHASTIC DYNAMIC PROGRAM



Goal: maximize expected total reward $\sum_{i=0}^{n-1} g(v_i, a_i) + h(v_n)$

STOCHASTIC DYNAMIC PROGRAM

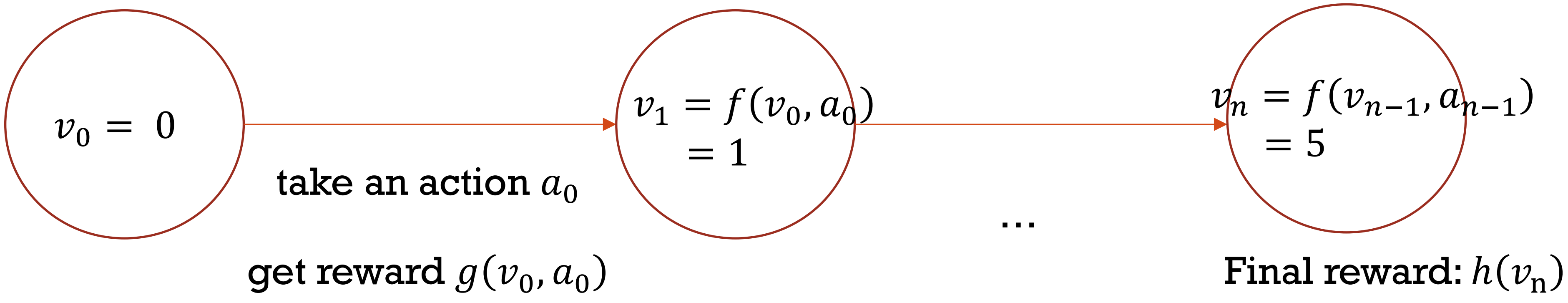


Goal: maximize expected total reward $\sum_{i=0}^{n-1} g(v_i, a_i) + h(v_n)$

[Fu Li and Xu 2018] There is a PTAS for any stochastic dynamic program such that

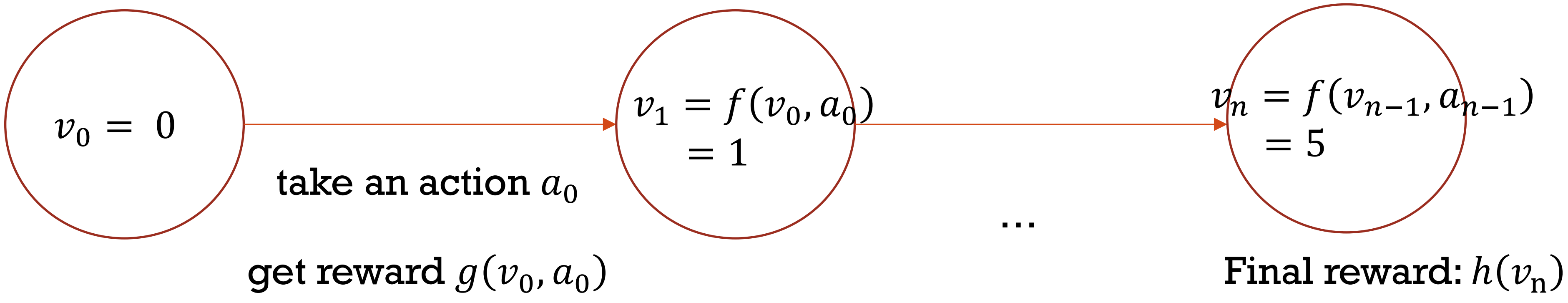
- v_i increase as time step i increases
- Value and action space are of reasonable size (related to ϵ)
- Immediate reward $g(v_i, a_i)$ has *expectation* ≥ 0
- Final reward $h(v_n) \geq 0$

REDUCING TO STOCHASTIC DYNAMIC PROGRAM



Goal: maximize expected total reward $\sum_{i=0}^{n-1} g(v_i, a_i) + h(v_n)$

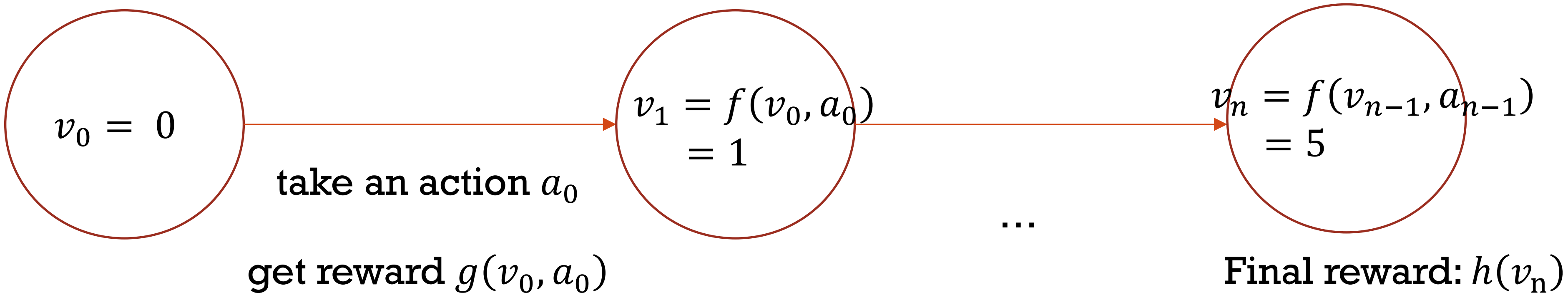
REDUCING TO STOCHASTIC DYNAMIC PROGRAM



Goal: maximize expected total reward $\sum_{i=0}^{n-1} g(v_i, a_i) + h(v_n)$

Optimal policy can be described as $(i_1, \cdots, i_k, i^*, \tau_1, \cdots, \tau_k)$

REDUCING TO STOCHASTIC DYNAMIC PROGRAM

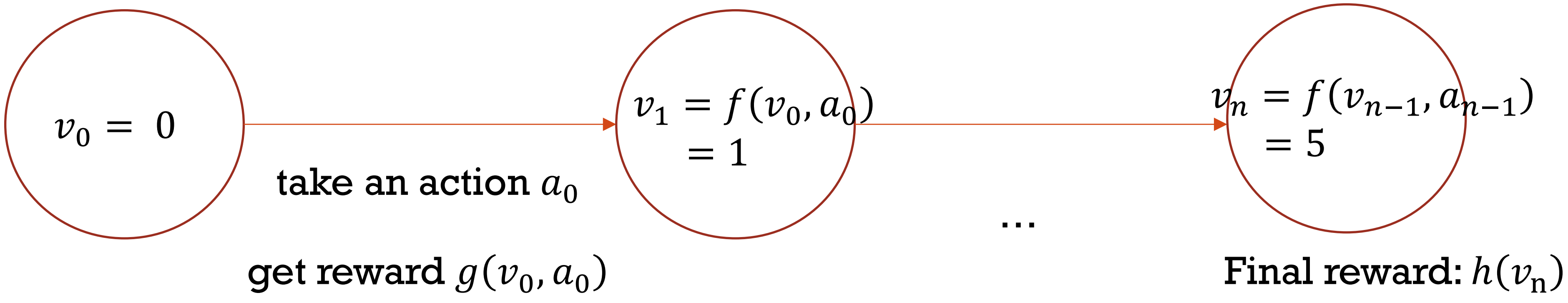


Goal: maximize expected total reward $\sum_{i=0}^{n-1} g(v_i, a_i) + h(v_n)$

Optimal policy can be described as $(i_1, \cdots, i_k, i^*, \tau_1, \cdots, \tau_k)$

- **Actions:** (phase one) open a box i with threshold τ_i
- (phase two) open a box i with threshold 0

REDUCING TO STOCHASTIC DYNAMIC PROGRAM

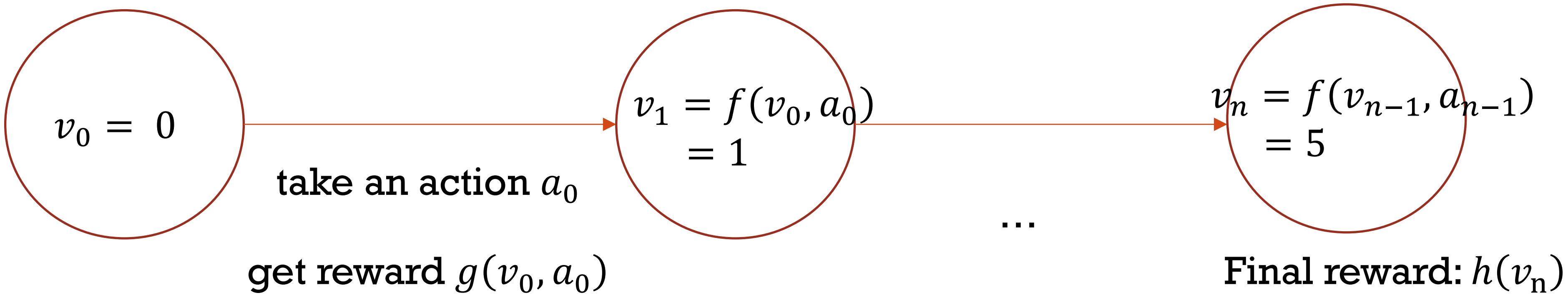


Goal: maximize expected total reward $\sum_{i=0}^{n-1} g(v_i, a_i) + h(v_n)$

Optimal policy can be described as $(i_1, \cdots, i_k, i^*, \tau_1, \cdots, \tau_k)$

- **Actions:** (phase one) open a box i with threshold τ_i
- (phase two) open a box i with threshold 0
- **Value:** the best value seen so far (0 if value is below the threshold)

REDUCING TO STOCHASTIC DYNAMIC PROGRAM

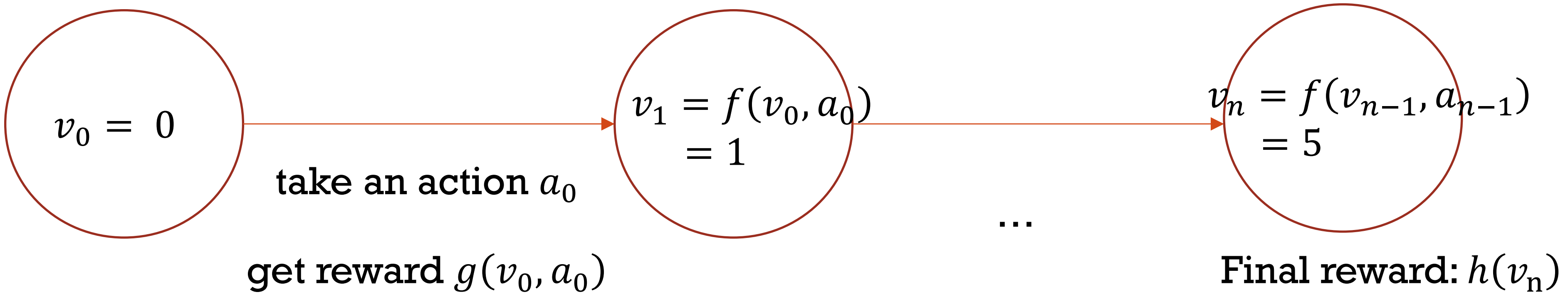


Goal: maximize expected total reward $\sum_{i=0}^{n-1} g(v_i, a_i) + h(v_n)$

Optimal policy can be described as $(i_1, \cdots, i_k, i^*, \tau_1, \cdots, \tau_k)$

- **Actions:** (phase one) open a box i with threshold τ_i
- (phase two) open a box i with threshold 0
- **Value:** the best value seen so far (0 if value is below the threshold)
- **Immediate Reward:** $v_i - v_{\{i-1\}} - c_i$

REDUCING TO STOCHASTIC DYNAMIC PROGRAM



Goal: maximize expected total reward $\sum_{i=0}^{n-1} g(v_i, a_i) + h(v_n)$

Optimal policy can be described as $(i_1, \cdots, i_k, i^*, \tau_1, \cdots, \tau_k)$

- **Actions:** (phase one) open a box i with threshold τ_i
- (phase two) open a box i with threshold 0
- **Value:** the best value seen so far (0 if value is below the threshold)
- **Immediate Reward:** $v_i - v_{\{i-1\}} - c_i$
- **Final reward:** (if we never reached phase two) $\mathrm{E}[v_{i^*}]$

REDUCING TO STOCHASTIC DYNAMIC PROGRAM

Optimal policy can be described as $(i_1, \cdots, i_k, i^*, \tau_1, \cdots, \tau_k)$

- **Actions:** (phase one) open a box i with threshold τ_i
- (phase two) open a box i with threshold 0
- **Value:** the best value seen so far (0 if value is below the threshold)
- **Immediate Reward:** $v_i - v_{\{i-1\}} - c_i$
- **Final reward:** (if we never reached phase two) $\mathrm{E}[v_{i^*}]$

[Fu Li and Xu 2018] There is a PTAS for any stochastic dynamic program such that

- v_i increase as time step i increases
- Value and action space are of reasonable size (related to ϵ)
- Immediate reward $g(v_i, a_i)$ has *expectation* ≥ 0
- Final reward $h(v_n) \geq 0$

REDUCING TO STOCHASTIC DYNAMIC PROGRAM

Optimal policy can be described as $(i_1, \cdots, i_k, i^*, \tau_1, \cdots, \tau_k)$

- **Actions:** (phase one) open a box i with threshold τ_i
- (phase two) open a box i with threshold 0
- **Value:** the best value seen so far (0 if value is below the threshold)
- **Immediate Reward:** $v_i - v_{\{i-1\}} - c_i$
- **Final reward:** (if we never reached phase two) $\mathrm{E}[v_{i^*}]$

[Fu Li and Xu 2018] There is a PTAS for any stochastic dynamic program such that

- v_i increase as time step i increases ✓
- Value and action space are of reasonable size (related to ϵ)
- Immediate reward $g(v_i, a_i)$ has *expectation* ≥ 0
- Final reward $h(v_n) \geq 0$

REDUCING TO STOCHASTIC DYNAMIC PROGRAM

Optimal policy can be described as $(i_1, \cdots, i_k, i^*, \tau_1, \cdots, \tau_k)$

- **Actions:** (phase one) open a box i with threshold τ_i
- (phase two) open a box i with threshold 0
- **Value:** the best value seen so far (0 if value is below the threshold)
- **Immediate Reward:** $v_i \; - \; v_{\{i-1\}} - \; c_i$
- **Final reward:** (if we never reached phase two) $\mathrm{E}[v_{i^*}]$

[Fu Li and Xu 2018] There is a PTAS for any stochastic dynamic program such that

- v_i increase as time step i increases ✓
- Value and action space are of reasonable size (related to ϵ) ✗
- Immediate reward $g(v_i, a_i)$ has *expectation* $\geq \; 0$
- Final reward $h(v_n) \geq \; 0$

REDUCING TO STOCHASTIC DYNAMIC PROGRAM

Optimal policy can be described as $(i_1, \cdots, i_k, i^*, \tau_1, \cdots, \tau_k)$

- **Actions:** (phase one) open a box i with threshold τ_i
- (phase two) open a box i with threshold 0
- **Value:** the best value seen so far (0 if value is below the threshold)
- **Immediate Reward:** $v_i - v_{\{i-1\}} - c_i$
- **Final reward:** (if we never reached phase two) $\mathrm{E}[v_{i^*}]$

[Fu Li and Xu 2018] There is a PTAS for any stochastic dynamic program such that

- v_i increase as time step i increases ✓
- Value and action space are of reasonable size (related to ϵ) ✗
- Immediate reward $g(v_i, a_i)$ has *expectation* ≥ 0 ?
- Final reward $h(v_n) \geq 0$

REDUCING TO STOCHASTIC DYNAMIC PROGRAM

Optimal policy can be described as $(i_1, \cdots, i_k, i^*, \tau_1, \cdots, \tau_k)$

- **Actions:** (phase one) open a box i with threshold τ_i
- (phase two) open a box i with threshold 0
- **Value:** the best value seen so far (0 if value is below the threshold)
- **Immediate Reward:** $v_i - v_{\{i-1\}} - c_i$
- **Final reward:** (if we never reached phase two) $\mathrm{E}[v_{i^*}]$

[Fu Li and Xu 2018] There is a PTAS for any stochastic dynamic program such that

- v_i increase as time step i increases ✓
- Value and action space are of reasonable size (related to ϵ) ✗
- Immediate reward $g(v_i, a_i)$ has *expectation* ≥ 0 ?
- Final reward $h(v_n) \geq 0$ ✓

REDUCING TO STOCHASTIC DYNAMIC PROGRAM

REDUCING TO STOCHASTIC DYNAMIC PROGRAM

- **Challenge 1:** negative terms in reward function reflecting costs
- **Solution:** Reduce finding the optimal two-phase policy to an equivalent problem without cost

REDUCING TO STOCHASTIC DYNAMIC PROGRAM

- **Challenge 1:** negative terms in reward function reflecting costs
- **Solution:** Reduce finding the optimal two-phase policy to an equivalent problem without cost

- **Challenge 2:** value space is too large to discretize in reasonable increment
- **Solution:**
- 1) For any fixed initial ordering of boxes, we can discretize the values to a set of size $poly\left(\frac{1}{\epsilon}\right)$
- 2) Only $n^{\left\{poly\left(\frac{1}{\epsilon}\right)\right\}}$ possible "small" sets of discretization, can try all of them

CONCLUSION

CONCLUSION

We show a simple two-phased structure of the optimal policy and provide a PTAS for the Pandora's box with nonobligatory inspection problem.

CONCLUSION

We show a simple two-phased structure of the optimal policy and provide a PTAS for the Pandora's box with nonobligatory inspection problem.

Future and concurrent directions:

- Does there exists a FPTAS for the Pandora's box with nonobligatory inspection problem?

CONCLUSION

We show a simple two-phased structure of the optimal policy and provide a PTAS for the Pandora's box with nonobligatory inspection problem.

Future and concurrent directions:

- Does there exists a FPTAS for the Pandora's box with nonobligatory inspection problem?
- What happens when the cost is not additive, or if we allow selection of multiple boxes subject to feasibility constraints?

CONCLUSION

We show a simple two-phased structure of the optimal policy and provide a PTAS for the Pandora's box with nonobligatory inspection problem.

Future and concurrent directions:

- Does there exists a FPTAS for the Pandora's box with nonobligatory inspection problem?
- What happens when the cost is not additive, or if we allow selection of multiple boxes subject to feasibility constraints?
- Could we model the fact that we could often inspect an option in different ways (e.g. online research, in person campus visit)?

CONCLUSION

We show a simple two-phased structure of the optimal policy and provide a PTAS for the Pandora's box with nonobligatory inspection problem.

Future and concurrent directions:

- Does there exists a FPTAS for the Pandora's box with nonobligatory inspection problem?
- What happens when the cost is not additive, or if we allow selection of multiple boxes subject to feasibility constraints?
- Could we model the fact that we could often inspect an option in different ways (e.g. online research, in person campus visit)?
- What would be the effect of risk aversion on the Pandora's box problem?

THANK YOU FOR LISTENING!