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

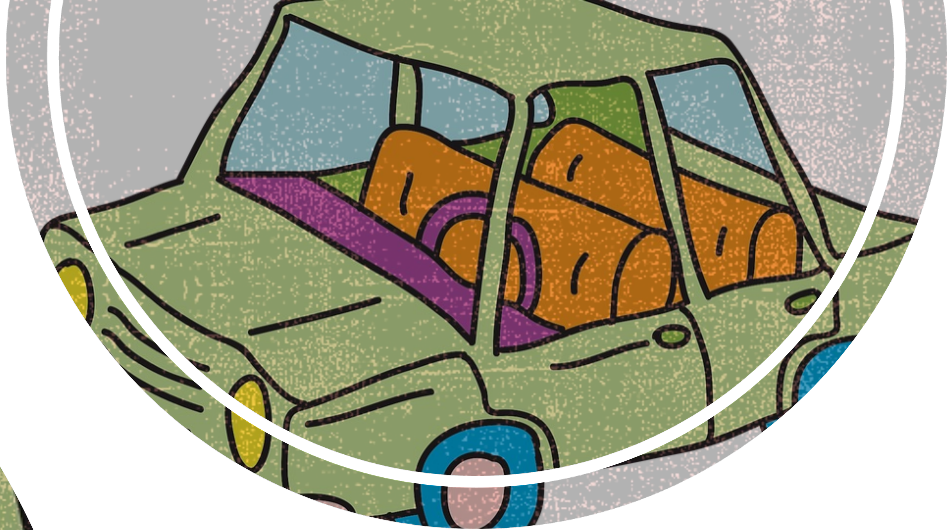
Linda Cai

Joint work with Hedyeh Beyhaghi

# OUR QUEST OF SEARCH

# OUR QUEST OF SEARCH

# OUR QUEST OF SEARCH

# WHY WOULD ANYONE BUY AN IDENTICAL PRODUCT AT A HIGHER PRICE?

# WEITZMAN'S ANSWER: SEARCH FRICTION

# WEITZMAN'S ANSWER: SEARCH FRICTION

Buying from a car dealer: the cost of inspection is high

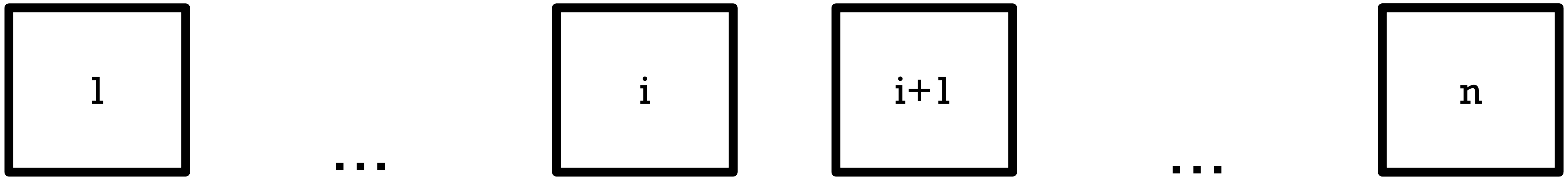
# WEITZMAN'S ANSWER: SEARCH FRICTION

Buying from a car dealer: the cost of inspection is high

Buying from a trusted friend: the cost of inspection is low

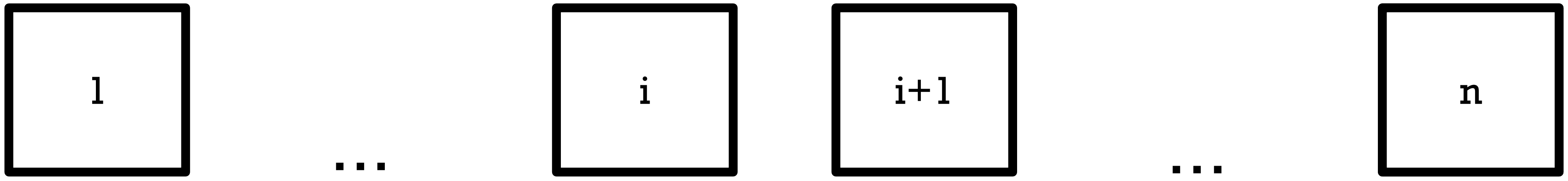
# PANDORA'S BOX PROBLEM

(Introduced by Weitzman79)



# PANDORA'S BOX PROBLEM

(Introduced by Weitzman79)




Value: $v_i \sim D_i$

# PANDORA'S BOX PROBLEM

(Introduced by Weitzman79)

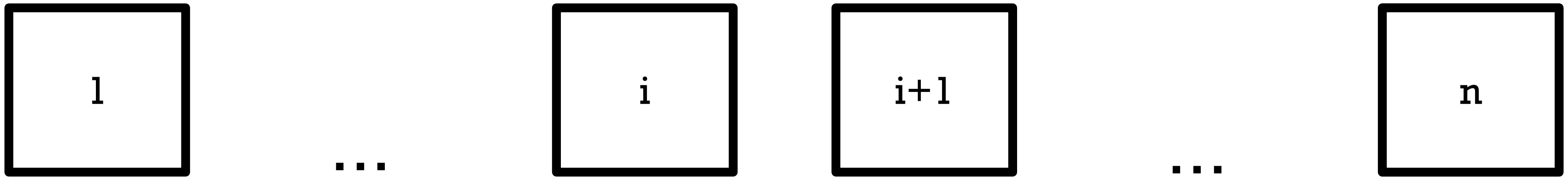
1 … i i+1 … n

Value: $v_i \sim D_i$

Cost: $c_i$

# PANDORA'S BOX PROBLEM

(Introduced by Weitzman79)



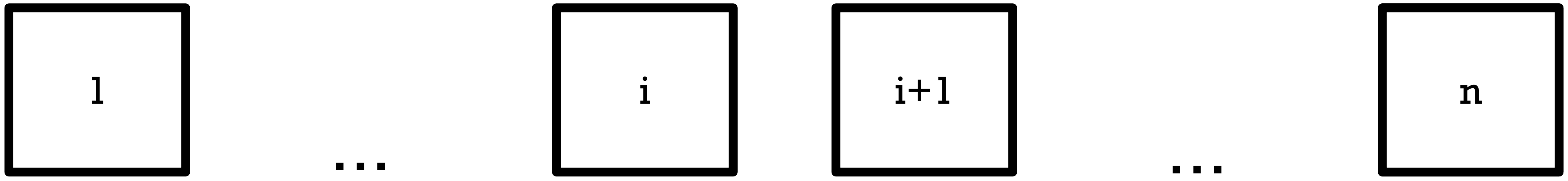

Value: $v_i \sim D_i$

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)

1 … i i+1 … n

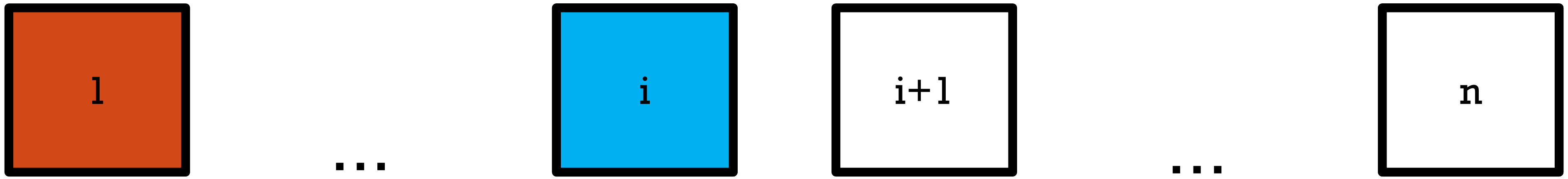
value = 2
cost = 1

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

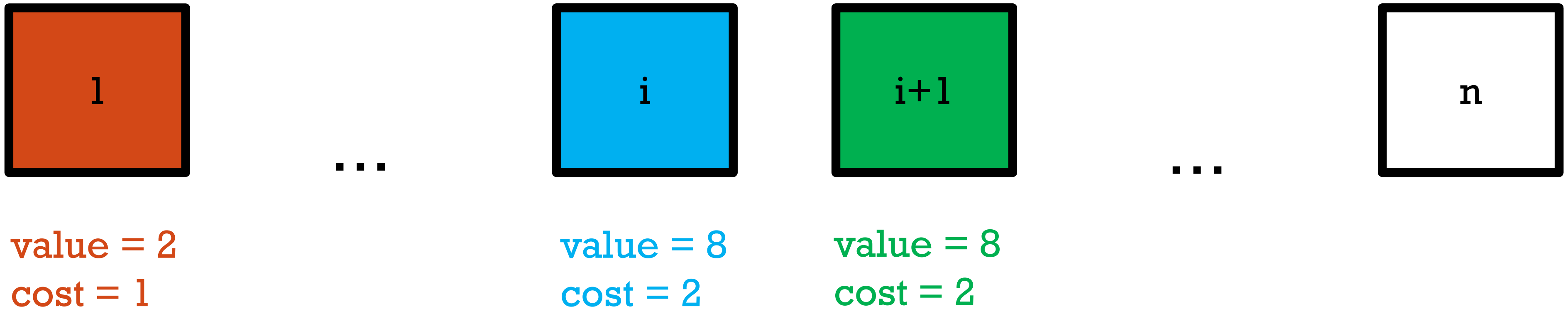
value = 8
cost = 2

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)

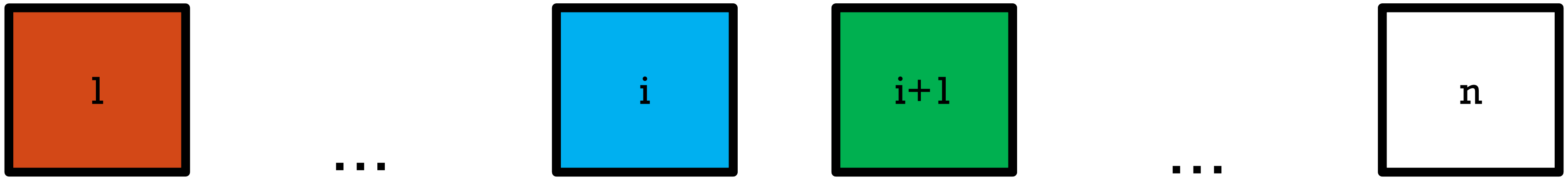



The agent can inspect boxes in any order they like, and their goal is to maximize their Expected Utility = E[value of selected box – sum of inspection costs]

*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 = 8
cost = 2

Utility = max(value) – sum(cost) = max(2, 8, 8) – (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: EXAMPLE

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

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

$c_2 = \epsilon$

# PANDORA'S BOX: EXAMPLE

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 algorithm:** open box 1 first, then open box 2 only when the value of box 1 is 0

# PANDORA'S BOX: EXAMPLE

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 algorithm:** open box 1 first, then open box 2 only when the value of box 1 is 0

**Observation:**

- If we see a high value from box 1, there is no need to open box 2

# PANDORA'S BOX: EXAMPLE

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 algorithm:** open box 1 first, then open box 2 only when the value of box 1 is 0

**Observation:**

- If we see a high value from box 1, there is no need to open box 2
- No matter what value we see from box 2, we still want to open box 1

# PANDORA'S BOX: WHEN IS IT WORTH OPENING THE BOX

Maximum value seen so far: $a$

We will also call this **outside option**

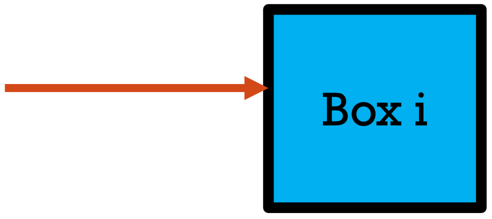
Box i

Value: $v_i \sim D_i$

Cost: $c_i$

# PANDORA'S BOX: WHEN IS IT WORTH OPENING THE BOX

Maximum value seen so far: $a$




Value: $v_i \sim D_i$

Cost: $c_i$

We will also call this **outside option**

Observation: given the outside option $a$, we should open box i only when *the expected marginal increase in value exceeds the cost*.

# Pandora's Box: When is it worth opening the box

Maximum value seen so far: $a$

Box i

Value: $v_i \sim D_i$

Cost: $c_i$

We will also call this **outside option**

Observation: given the outside option $a$, we should open box i only when *the expected marginal increase in value exceeds the cost*.

If we don't open the box: value = $a$

cost = $0$

# PANDORA'S BOX: WHEN IS IT WORTH OPENING THE BOX

Maximum value seen so far: $a$ → Box i

Value: $v_i \sim D_i$

Cost: $c_i$

We will also call this **outside option**

Observation: given the outside option $a$, we should open box i only when *the expected marginal increase in value exceeds the cost*.

If we don't open the box: value = $a$
cost = $0$

If we open the box: value = $\max(a\,, v_i)$
cost = $c_i$

# PANDORA'S BOX: WHEN IS IT WORTH OPENING THE BOX

Maximum value seen so far: $a$ → Box i

Value: $v_i \sim D_i$

Cost: $c_i$

We will also call this **outside option**

Observation: given the outside option $a$, we should open box i only when *the expected marginal increase in value exceeds the cost*.

If we don't open the box: value = $a$
cost = $0$

If we open the box: value = $\max(a, v_i)$
cost = $c_i$

Box i is worth opening only when $\mathrm{E}[\max(v_i, a)] - a > c_i$

$$\Leftrightarrow \quad \mathrm{E}[\max(v_i - a, 0)] = \mathrm{E}[(v_i - a)^+] > c_i$$

# PANDORA'S BOX: WHEN IS IT WORTH OPENING THE BOX

Maximum value seen so far: $a$ → Box i

Value: $v_i \sim D_i$

Cost: $c_i$

We will also call this **outside option**

Observation: given the outside option $a$, we should open box i only when *the expected marginal increase in value exceeds the cost*.

If we don't open the box: value = $a$
cost = $0$

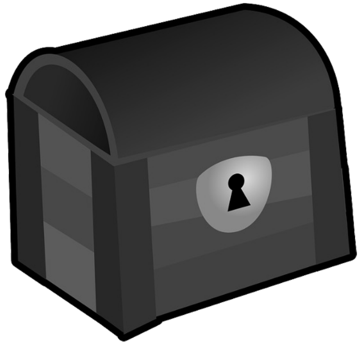
If we open the box: value = $\max(a\,, v_i)$
cost = $c_i$

Box i is worth opening only when $\mathrm{E}[\max(v_i, a)] - a > c_i$

$$\Leftrightarrow \quad \mathrm{E}[\max(v_i - a, 0)] = \mathrm{E}[(v_i - a)^+] > c_i$$

We will call $\sigma_i$ such that $\mathrm{E}[(v_i - \sigma_i)^+] = c_i$ **the strike price**

# PANDORA'S BOX: OPTIMAL POLICY [WEITZ79]

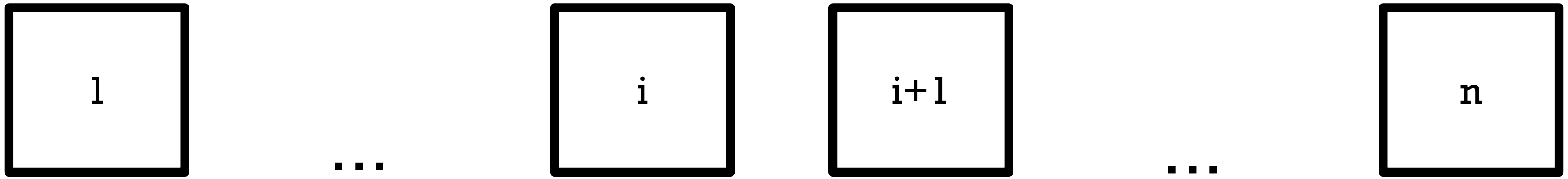


Value: $v_i \sim D_i$

Cost: $c_i$

**Strike price:** $\sigma_i$ such that $\mathrm{E}[(v_i - \sigma_i)^+] = c_i$

# PANDORA'S BOX: OPTIMAL POLICY [WEITZ79]



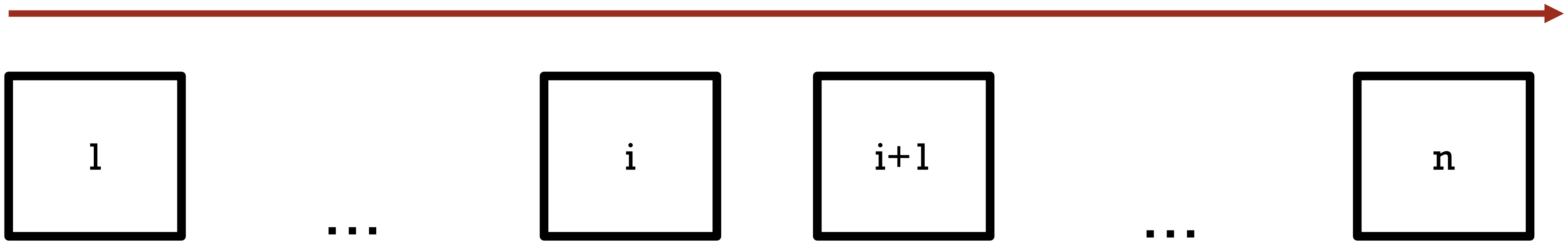

Value: $v_i \sim D_i$

Cost: $c_i$

Strike price: $\sigma_i \coloneqq \mathrm{E}[(v_i - \sigma_i)^+] = c_i$

# PANDORA'S BOX: OPTIMAL POLICY [WEITZ79]

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




Value: $v_i \sim D_i$

Cost: $c_i$

Strike price: $\sigma_i \coloneqq \mathrm{E}[(v_i - \sigma_i)^+] = c_i$

# PANDORA'S BOX: OPTIMAL POLICY [WEITZ79]

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

| 1 | … | i | i+1 | … | n |
|---|---|---|---|---|---|

Value: $v_i \sim D_i$

Cost: $c_i$

Strike price: $\sigma_i \coloneqq \mathrm{E}[(v_i - \sigma_i)^+] = c_i$

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?

# PANDORA'S BOX: IS INSPECTION NECESSARY?

Super busy person who needs a car the next day:

# PANDORA'S BOX: IS INSPECTION NECESSARY?

Super busy person who needs a car the next day:

What about just … wing it?

# PANDORA'S BOX: IS INSPECTION NECESSARY?

# PANDORA'S BOX: IS INSPECTION NECESSARY?

International student: campus visits are too costly and time consuming

# PANDORA'S BOX: IS INSPECTION NECESSARY?

International student: campus visits are too costly and time consuming

It's a great school, let's just go!

# GIVING THE AGENT THE FREEDOM OF CHOICE

# GIVING THE AGENT THE FREEDOM OF CHOICE

- **[GMS08]** Information Acquisition and Exploitation in Multichannel Wireless Networks.

# GIVING THE AGENT THE FREEDOM OF CHOICE

- **[GMS08]** Information Acquisition and Exploitation in Multichannel Wireless Networks.
- **[CL09]** Optimal channel probing and transmission scheduling for opportunistic spectrum access.

# GIVING THE AGENT THE FREEDOM OF CHOICE

- **[GMS08]** Information Acquisition and Exploitation in Multichannel Wireless Networks.
- **[CL09]** Optimal channel probing and transmission scheduling for opportunistic spectrum access.
- **[AKLS17]** Stochastic selection problems with testing.

# GIVING THE AGENT THE FREEDOM OF CHOICE

- **[GMS08]** Information Acquisition and Exploitation in Multichannel Wireless Networks.
- **[CL09]** Optimal channel probing and transmission scheduling for opportunistic spectrum access.
- **[AKLS17]** Stochastic selection problems with testing.
- **[Dov18]** Whether or not to open Pandora's box.

# GIVING THE AGENT THE FREEDOM OF CHOICE

- **[GMS08]** Information Acquisition and Exploitation in Multichannel Wireless Networks.
- **[CL09]** Optimal channel probing and transmission scheduling for opportunistic spectrum access.
- **[AKLS17]** Stochastic selection problems with testing.
- **[Dov18]** Whether or not to open Pandora's box.

# PANDORA'S BOX PROBLEM WITH NON-OBLIGATORY INSPECTION

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

(Introduced independently by GMS08, CL09, AKLS17, Dov18)

1 … i i+1 … n

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 WITH NON-OBLIGATORY INSPECTION (PNOI)

(Introduced independently by GMS08, CL09, AKLS17, Dov18)

1 … i i+1 … n

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*

# 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

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

$c_1 = 1/2$

Box 2

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

$c_2 = \epsilon$

**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

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

$$c_1 = 1/2$$

Box 2

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

$$c_2 = \epsilon$$

**Weitzman's policy:** 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

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

$c_1 = 1/2$

Box 2

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

$c_2 = \epsilon$

**Weitzman's policy:** 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

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

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

$c_2 = \epsilon$

Box 3

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

$c_3 = 1/2$

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
- [FLL22] NP-Hardness

**Theorem** [FLL Arxiv Preprint 22*]: 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: APPROXIMATELY OPTIMAL

Theorem [GMS 08]:

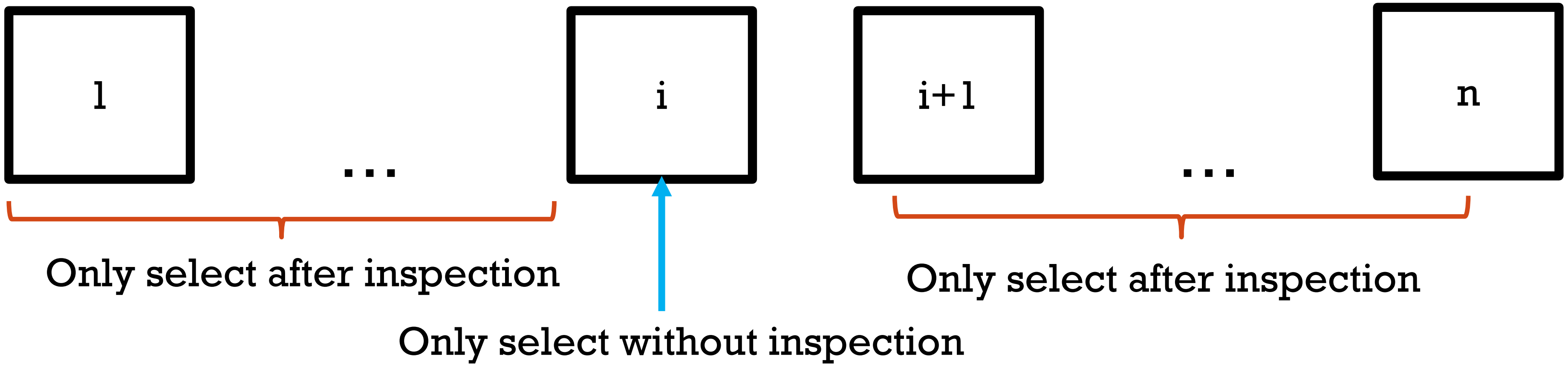
There exists a polynomial time policy which 0.8 approximates the optimal policy.

# PNOI: APPROXIMATELY OPTIMAL

Theorem [GMS 08]:

There exists a polynomial time policy which 0.8 approximates the optimal policy.

Non-adaptive order policy [BK19, GMS08]:



# 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 [ASZ20]:

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 [ASZ20]:

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 [ASZ20]:

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 (to appear in 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

$$v_2 = \begin{cases} 1 \ \text{w.p.}\ 1/2 \\ 0 \ \text{w.p.}\ 1/2 \end{cases}$$

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 [GMS08]**

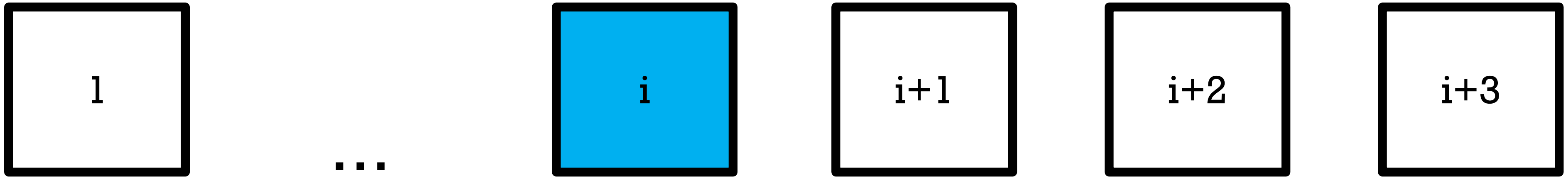
There exists an optimal policy for PNOI that has *at most one* back up box.

# PNOI: STRUCTURE OF THE OPTIMAL POLICY

**Structural Theorem [GMS08]**

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

Step one: select a backup box (if any)

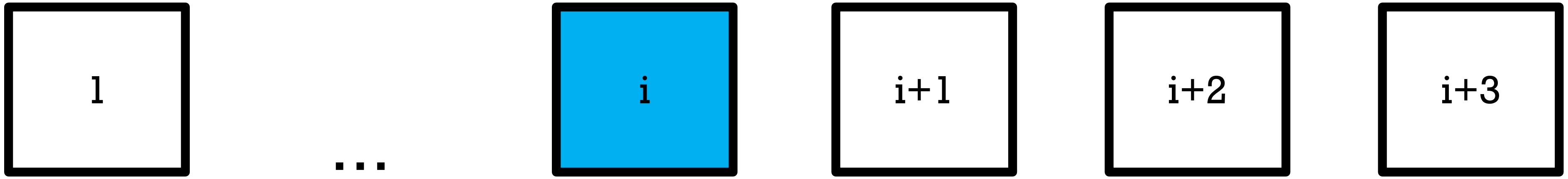


# PNOI: STRUCTURE OF THE OPTIMAL POLICY

**Structural Theorem [GMS08]**

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

Step one: select a backup box (if any)



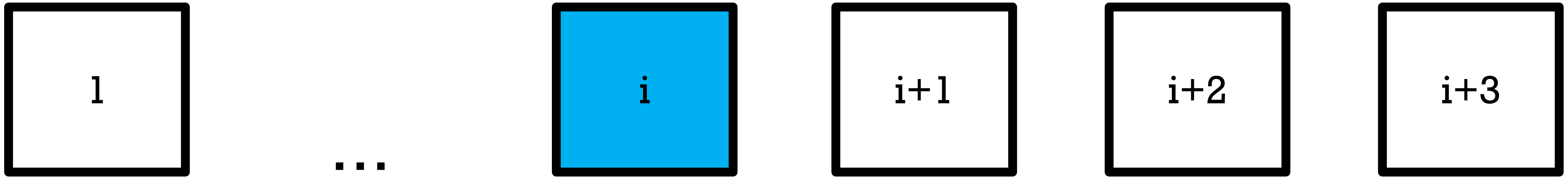

Step two: find optimal policy given that box i is the unique backup box

# PNOI: STRUCTURE OF THE OPTIMAL POLICY

**Structural Theorem [GMS08]**

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

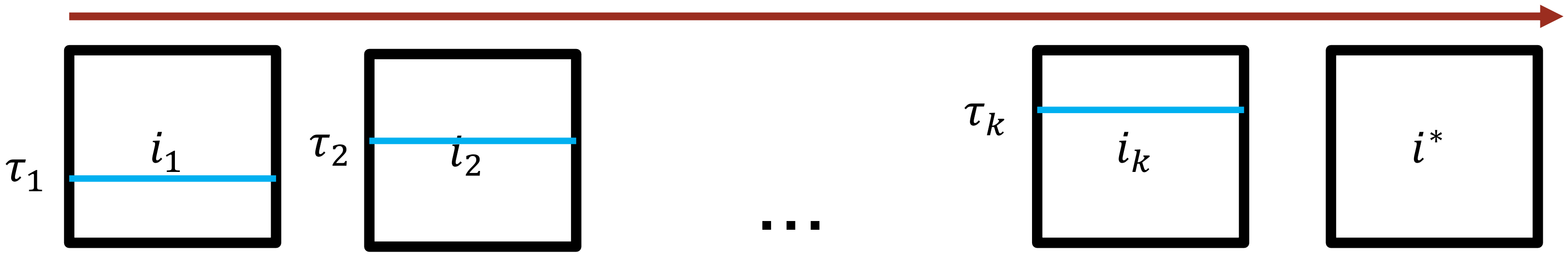
# PNOI: STRUCTURE OF THE OPTIMAL POLICY

**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 $(\tau_1, \cdots, \tau_k)$

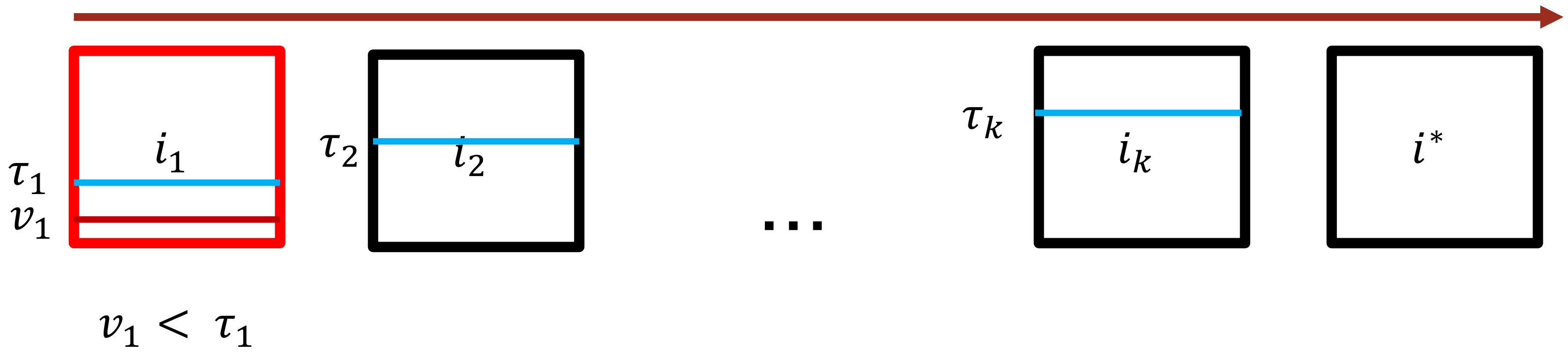


# PNOI: STRUCTURE OF THE OPTIMAL POLICY

**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

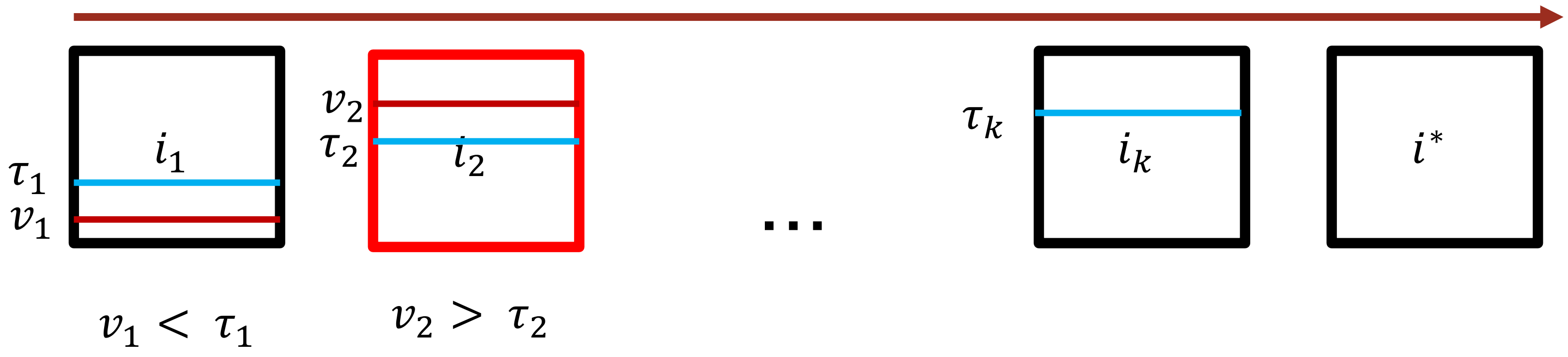

If we reach the end of the order, claim the backup box closed without inspection.

# 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



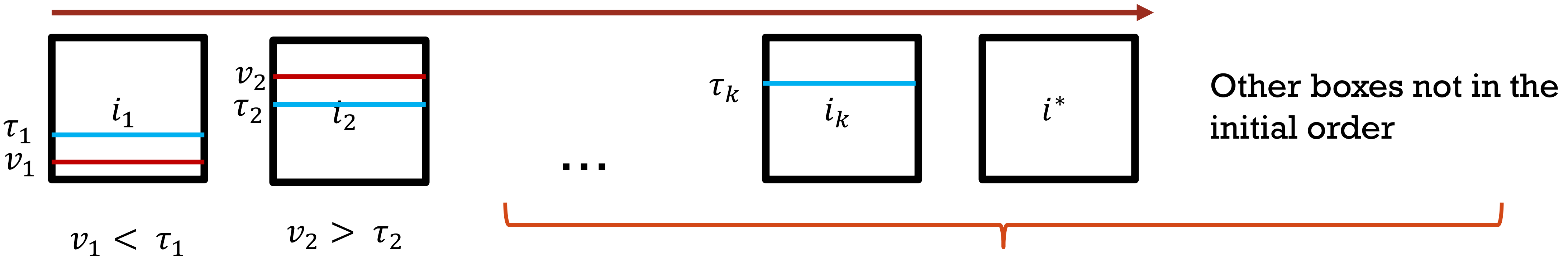

If we see a value above threshold, the policy enters phase two

# 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




Other boxes not in the initial order

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

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=$\tau_1, \cdots, \tau_k$)

**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.

**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

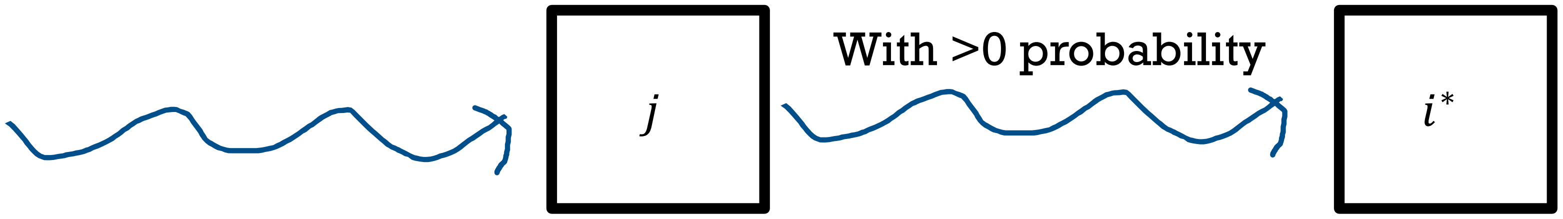
- Consider an optimal policy that uses at most one backup box.
- If the policy does not use backup box, then Weitzman policy is the optimal policy

# 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 $i^*$

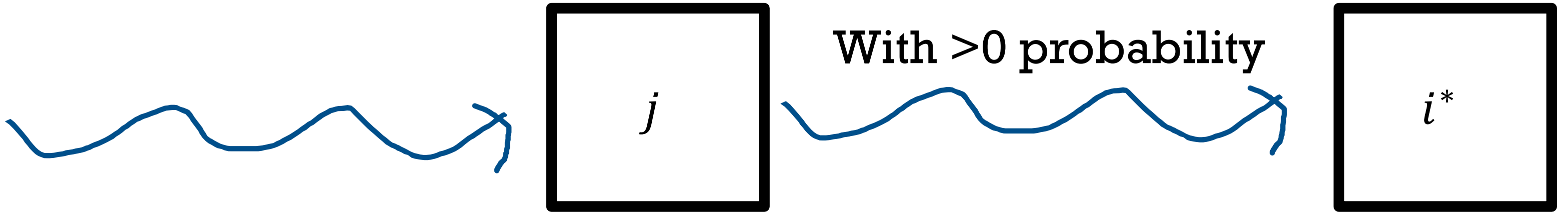
# PROOF SKETCH OF OPTIMALITY

- 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 $i^*$

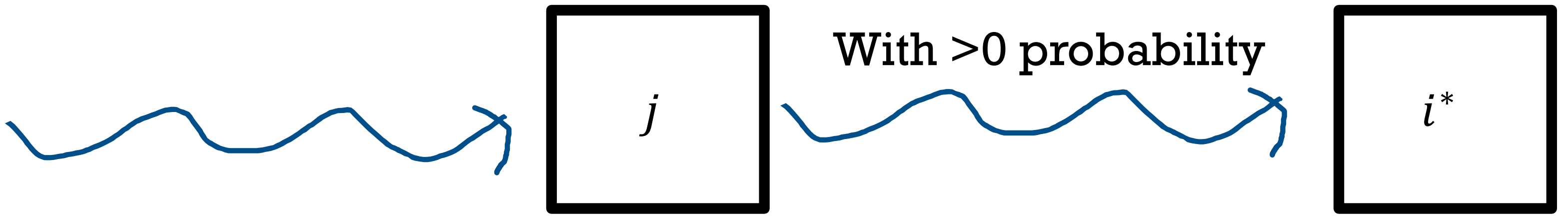



If after opening box j, we still may claim backup box closed with some probability, then:

- 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 $i^*$



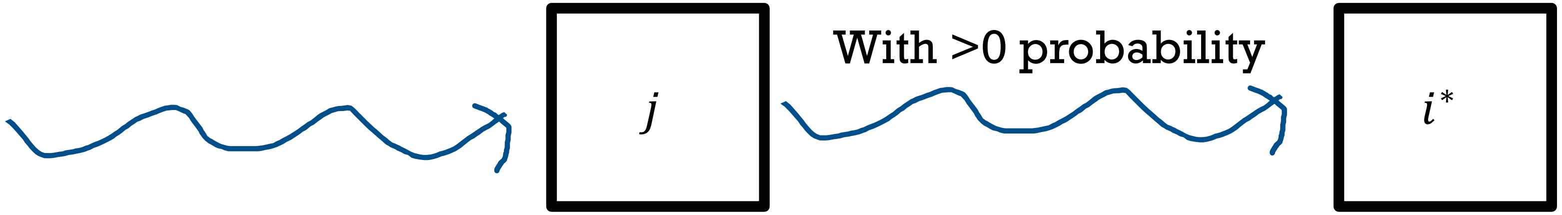

If after opening box j, we still may claim backup box closed with some probability, then:

- 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 $v_j = 0$

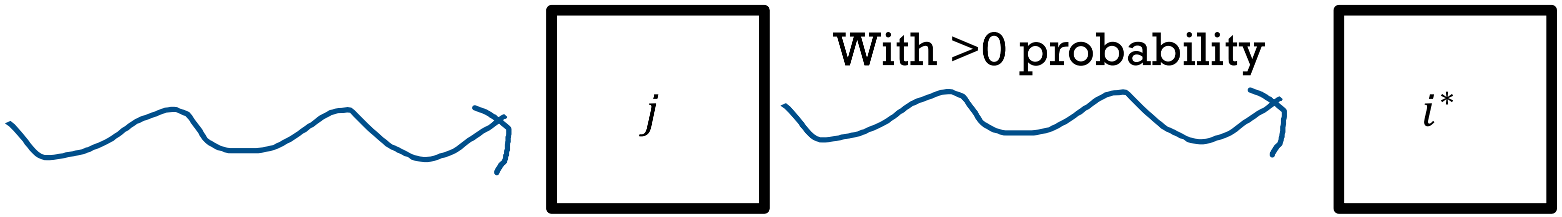
# PROOF SKETCH OF OPTIMALITY

- 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

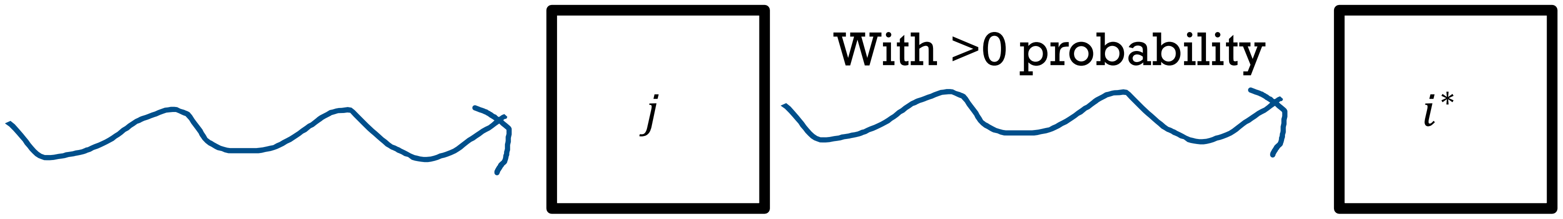
- Consider an optimal policy that uses at most one backup box.
- If the policy does not use backup box, then Weitzman policy is the optimal policy
- If the policy uses backup box, let this box be $i^*$




Let $\tau_j$ 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 $\tau_j$ 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.

- Stochastic dynamic program formulated in [FLX18] has a PTAS
- We restrict the search space to finding approximately optimal two-phase policy
- We can reduce our problem to stochastic dynamic program in [FLX18]

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

# STOCHASTIC DYNAMIC PROGRAM [FLX18]

$v_0 = 0$

take an action $a_0$

get reward $g(v_0, a_0)$

$v_1 = f(v_0, a_0)$
$= 1$

…

$v_n = f(v_{n-1}, a_{n-1})$
$= 5$

Final reward: $h(v_\mathrm{n})$

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

# STOCHASTIC DYNAMIC PROGRAM [FLX18]

$v_0 = 0$ $\xrightarrow{}$ $v_1 = f(v_0, a_0) = 1$ $\xrightarrow{}$ … $\xrightarrow{}$ $v_n = f(v_{n-1}, a_{n-1}) = 5$

take an action $a_0$

get reward $g(v_0, a_0)$

…

Final reward: $h(v_\mathrm{n})$

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

[FLX18] 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 $\epsilon$)
- Immediate reward $g(v_i, a_i)$ has *expectation* $\geq 0$
- Final reward $h(v_n) \geq 0$

# REDUCING TO STOCHASTIC DYNAMIC PROGRAM

$v_0 = 0$ → $v_1 = f(v_0, a_0) = 1$ → … → $v_n = f(v_{n-1}, a_{n-1}) = 5$

take an action $a_0$

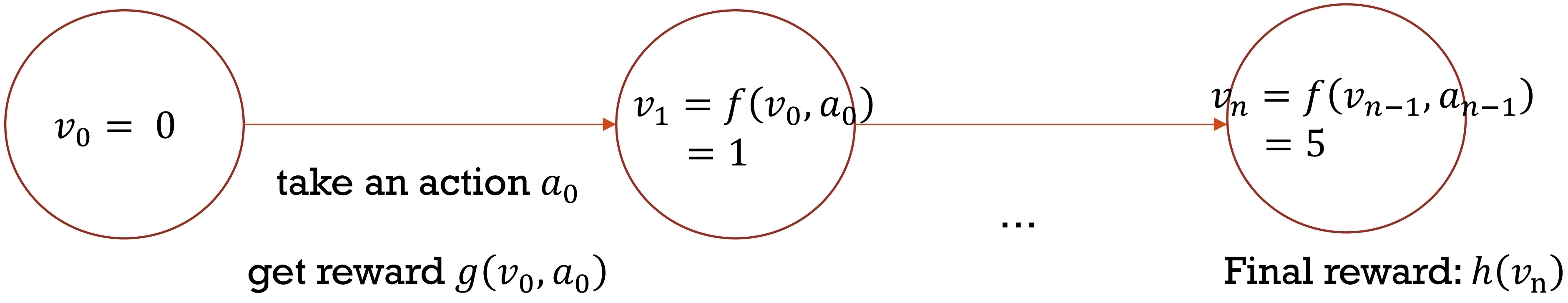
get reward $g(v_0, a_0)$

…

Final reward: $h(v_\mathrm{n})$

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

$v_0 = 0$ $\longrightarrow$ $v_1 = f(v_0, a_0) = 1$ $\longrightarrow$ … $\longrightarrow$ $v_n = f(v_{n-1}, a_{n-1}) = 5$

take an action $a_0$

get reward $g(v_0, a_0)$

…

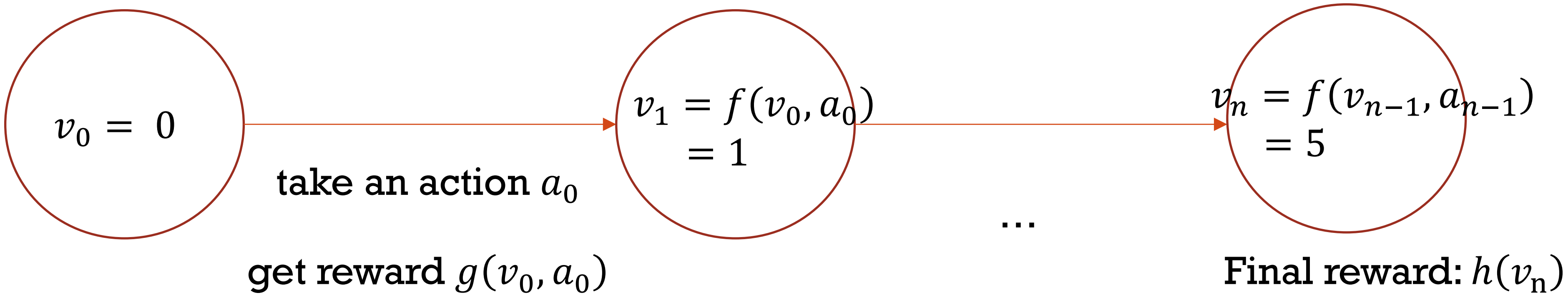
Final reward: $h(v_{\mathrm{n}})$

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 $\tau_i$
- (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 $\tau_i$
- (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

$v_0 = 0$ $\xrightarrow{\text{take an action } a_0}$ $v_1 = f(v_0, a_0) = 1$ $\rightarrow \cdots \rightarrow$ $v_n = f(v_{n-1}, a_{n-1}) = 5$

get reward $g(v_0, a_0)$

Final reward: $h(v_n)$

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 $\tau_i$
- (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

$v_0 = 0$ → $v_1 = f(v_0, a_0) = 1$ → … → $v_n = f(v_{n-1}, a_{n-1}) = 5$

take an action $a_0$

get reward $g(v_0, a_0)$

…

Final reward: $h(v_n)$

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 $\tau_i$
- (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 $\tau_i$
- (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) $\mathrm{E}[v_{i^*}]$

[FLX18] 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 $\epsilon$)
- 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 $\tau_i$
- (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) $\mathrm{E}[v_{i^*}]$

[FLX18] 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 $\epsilon$)
- 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 $\tau_i$
- (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) $\mathrm{E}[v_{i^*}]$

[FLX18] 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 $\epsilon$)
- 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 $\tau_i$
- (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) $\mathrm{E}[v_{i^*}]$

[FLX18] 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 $\epsilon$) ✗
- 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 $\tau_i$
- (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) $\mathrm{E}[v_{i^*}]$

[FLX18] 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 $\epsilon$) ✗
- Immediate reward $g(v_i, a_i)$ has *expectation* $\geq 0$ ?
- 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 small set
- 2) Only $n^{\{poly\left(\frac{1}{\epsilon}\right)\}}$ 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 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 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 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 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!