Designing Exploration Contracts

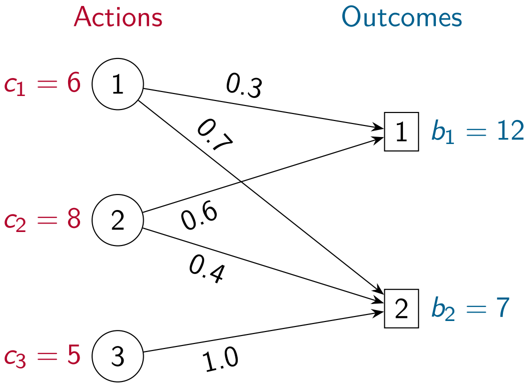
Martin Hoefer, Conrad Schecker, Kevin Schewior

RWTH Aachen University, Goethe University Frankfurt, University of Cologne

42nd International Symposium on Theoretical Aspects of Computer Science, Jena

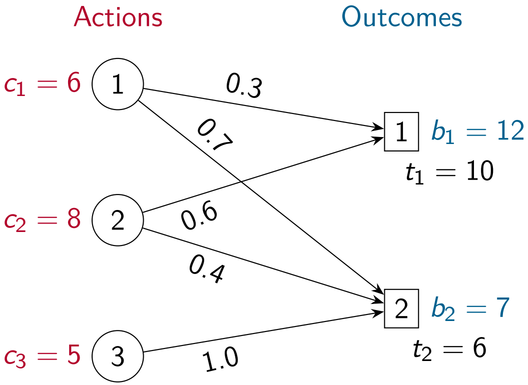
Contract Design

A **principal** $\mathcal{P}$ delegates a costly task to an **agent** $\mathcal{A}$



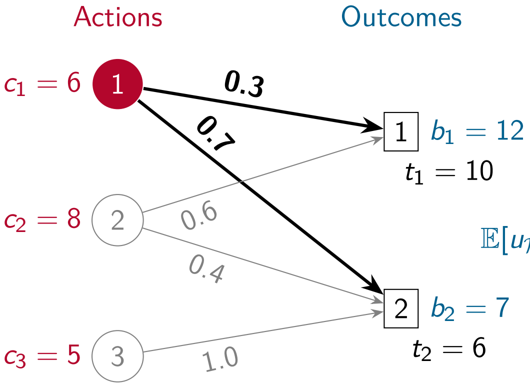
Contract Design

A **principal** $\mathcal{P}$ delegates a costly task to an **agent** $\mathcal{A}$ using a **contract** t.



Contract Design

A **principal** $\mathcal{P}$ delegates a costly task to an **agent** $\mathcal{A}$ using a **contract** t.

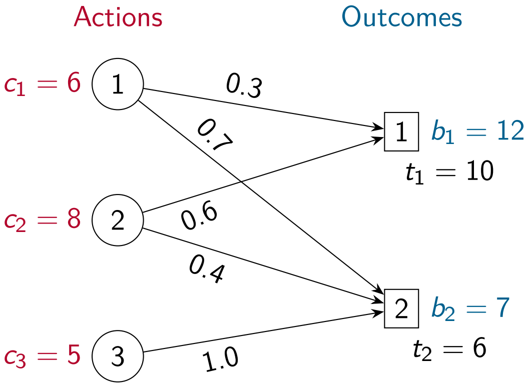


$$\mathbb{E}[u_{\mathcal{A}}] = 0.3 \cdot 10 + 0.7 \cdot 6 - 6$$

$$\mathbb{E}[u_{\mathcal{P}}] = 0.3 \cdot 2 + 0.7 \cdot 1$$

Contract Design

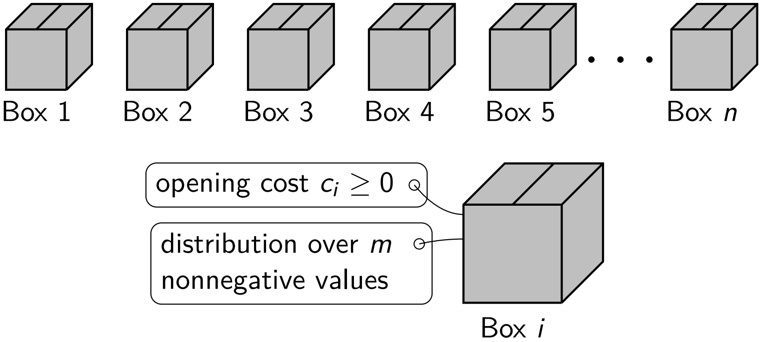
A **principal** $\mathcal{P}$ delegates a costly task to an **agent** $\mathcal{A}$ using a **contract** t.



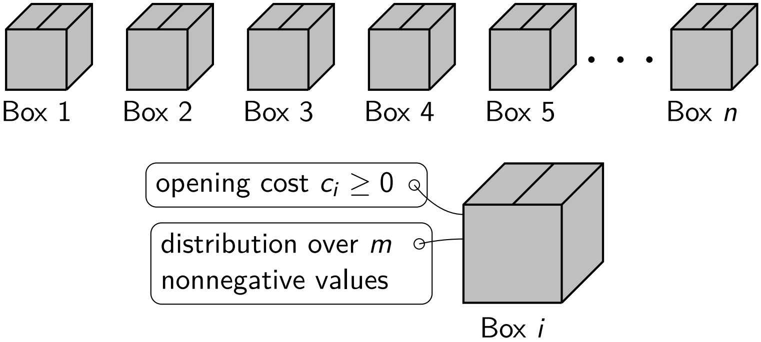
- Overview: *Algorithmic Contract Design*

  [Dütting, Feldman, Talgam-Cohen; Found. Trends Theor. Comput. Sci., 2024]
- What if $\mathcal{A}$ can **explore actions**?

Exploration: Pandora's Box Problem



Exploration: Pandora's Box Problem



- Open box i to reveal actual contained value, then stop or continue.
- **Adaptively** open (any) boxes in any order.
- In the end, maximum revealed value is taken.

Applications: Buying a house, hiring a job candidate, . . .

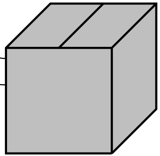
[Weitzman; Econometrica, 1979]

Pandora's Box Problem with Principal and Agent

A **principal** $\mathcal{P}$ delegates the exploration to an **agent** $\mathcal{A}$.

opening cost $c_i \geq 0$

value-pair $(a_{i,1}, b_{i,1})$ with prob. $p_{i,1}$
value-pair $(a_{i,2}, b_{i,2})$ with prob. $p_{i,2}$
...
value-pair $(a_{i,m}, b_{i,m})$ with prob. $p_{i,m}$

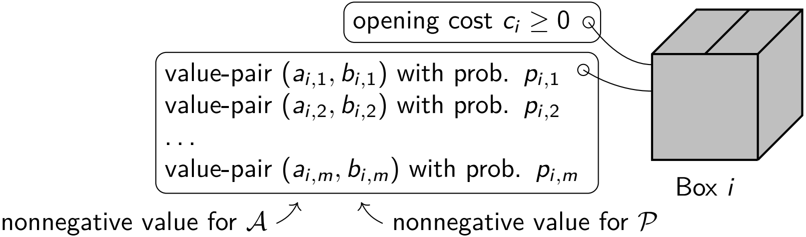


Box i

nonnegative value for $\mathcal{A}$ ↗ ↖ nonnegative value for $\mathcal{P}$

Pandora's Box Problem with Principal and Agent

A **principal** $\mathcal{P}$ delegates the exploration to an **agent** $\mathcal{A}$.



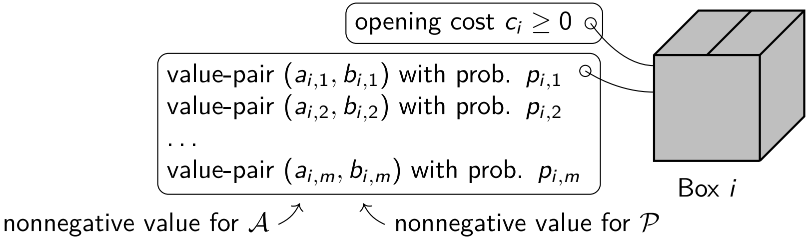
$\mathcal{P}$ commits to a **contract**, specifying payments $t_{i,j} \in [0, b_{i,j}]$ beforehand.

Suppose $\mathcal{A}$ takes outcome $(a_{i,j}, b_{i,j})$ in the end.

- $\mathcal{A}$ receives $a_{i,j} + t_{i,j}$, and bears all opening costs,
- $\mathcal{P}$ receives $b_{i,j} - t_{i,j}$.

Pandora's Box Problem with Principal and Agent

A **principal** $\mathcal{P}$ delegates the exploration to an **agent** $\mathcal{A}$.



$\mathcal{P}$ commits to a **contract**, specifying payments $t_{i,j} \in [0, b_{i,j}]$ beforehand.

Suppose $\mathcal{A}$ takes outcome $(a_{i,j}, b_{i,j})$ in the end.

- $\mathcal{A}$ receives $a_{i,j} + t_{i,j}$, and bears all opening costs,
- $\mathcal{P}$ receives $b_{i,j} - t_{i,j}$.

Goal: Find an optimal exploration contract for $\mathcal{P}$.

Index Policy for $\mathcal{A}$

- Calculate an **index** φ_i for every box i.
- Consider boxes in non-increasing order of φ_i.
- Suppose box i is considered. Let $v := \max_{(i',j')\text{ revealed}}\{a_{i',j'} + t_{i',j'}\}$.
  Open box i only if $v \leq \varphi_i$. Stop only if $v \geq \varphi_i$.
- Break ties in favor of $\mathcal{P}$.

[Weitzman; Econometrica, 1979]

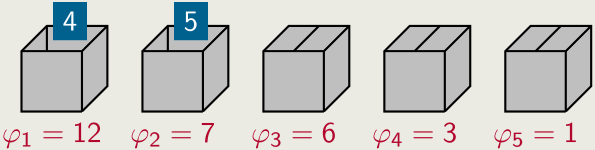
Index Policy for $\mathcal{A}$

- Calculate an **index** φ_i for every box i.
- Consider boxes in non-increasing order of φ_i.
- Suppose box i is considered. Let $v := \max_{(i',j') \text{ revealed}} \{a_{i',j'} + t_{i',j'}\}$.

  Open box i only if $v \leq \varphi_i$. Stop only if $v \geq \varphi_i$.
- Break ties in favor of $\mathcal{P}$.

[Weitzman; Econometrica, 1979]

Example



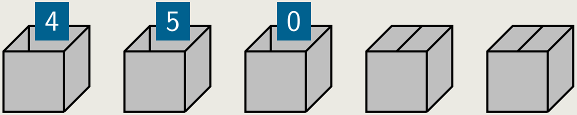
Index Policy for $\mathcal{A}$

- Calculate an **index** φ_i for every box i.
- Consider boxes in non-increasing order of φ_i.
- Suppose box i is considered. Let $v := \max_{(i',j') \text{ revealed}}\{a_{i',j'} + t_{i',j'}\}$.

  Open box i only if $v \leq \varphi_i$. Stop only if $v \geq \varphi_i$.
- Break ties in favor of $\mathcal{P}$.

[Weitzman; Econometrica, 1979]

Example



$\varphi_1 = 12$ $\varphi_2 = 7$ $\varphi_3 = 6$ $\varphi_4 = 3$ $\varphi_5 = 1$

The Index φ_i

For the **index** φ_i of box $i \in [n]$, it holds

$$\sum_{j \in [m]} p_{i,j} \cdot \max\{0, a_{i,j} + t_{i,j} - \varphi_i\} = c_i .$$

The Index φ_i

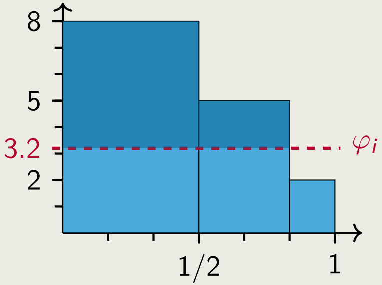
For the **index** φ_i of box $i \in [n]$, it holds

$$\sum_{j\in[m]} p_{i,j} \cdot \max\{0, a_{i,j} + t_{i,j} - \varphi_i\} = c_i.$$

Example

Consider some box i with

- $c_i = 3$,
- $a_{i,j} + t_{i,j} = \begin{cases} 5+3, & \text{w.p. } 1/2 \\ 0+5, & \text{w.p. } 1/3 \\ 1+1, & \text{w.p. } 1/6 \end{cases}.$



The Index φ_i

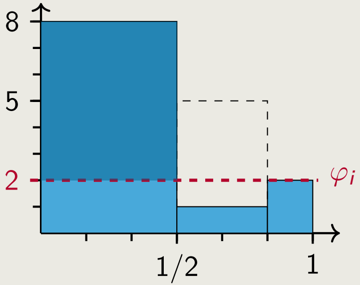
For the **index** φ_i of box $i \in [n]$, it holds

$$\sum_{j \in [m]} p_{i,j} \cdot \max\{0, a_{i,j} + t_{i,j} - \varphi_i\} = c_i.$$

Example

Consider some box i with

- $c_i = 3$,
- $a_{i,j} + t_{i,j} = \begin{cases} 5 + 3, & \text{w.p. } 1/2 \\ 0 + 1, & \text{w.p. } 1/3. \\ 1 + 1, & \text{w.p. } 1/6 \end{cases}$



Linear Contracts

Definition

A **linear contract** is given by an $\alpha \in [0, 1]$ such that

$$t_{i,j} = \alpha \cdot b_{i,j}$$

holds for all $(i, j) \in [n] \times [m]$.

Linear Contracts

Definition

A **linear contract** is given by an $\alpha \in [0, 1]$ such that

$$t_{i,j} = \alpha \cdot b_{i,j}$$

holds for all $(i, j) \in [n] \times [m]$.

- Goal: Find $\alpha \in [0, 1]$ such that $\mathcal{P}$'s expected utility is maximized.
- Linear contracts are simple and practical. An important subclass!

[Dütting, Roughgarden, and Talgam-Cohen. EC 2019.]

Computation

Theorem

An optimal linear exploration contract can be computed in poly-time.

Computation

Theorem

An optimal linear exploration contract can be computed in poly-time.

Proof outline.

- Only consider critical α-values where $\mathcal{A}$'s action changes.

Computation

Theorem

An optimal linear exploration contract can be computed in poly-time.

Proof outline.

- Only consider critical α-values where $\mathcal{A}$'s action changes.
  - $\mathcal{A}$ might behave identically under contracts α and $\alpha - \varepsilon$.
  - Then $\mathcal{P}$ pays $\alpha - \varepsilon$ instead of α for the same expected outcome.
  - If there is no such $\varepsilon > 0$, then α is criticial.

Computation

Theorem

An optimal linear exploration contract can be computed in poly-time.

Proof outline.

- Only consider critical α-values where $\mathcal{A}$'s action changes.
  - $\mathcal{A}$ might behave identically under contracts α and $\alpha - \varepsilon$.
  - Then $\mathcal{P}$ pays $\alpha - \varepsilon$ instead of α for the same expected outcome.
  - If there is no such $\varepsilon > 0$, then α is criticial.
- Efficient enumeration of critical α-values would be sufficient:
  For every critical α, compute the expected value of $\mathcal{P}$.

Computation

Efficient enumeration of critical α-values is actually possible:

Computation

Efficient enumeration of critical α-values is actually possible:

- Consider the index policy.

  $\rightarrow$ $\mathcal{A}$'s behavior changes only at intersections of...

  ... some $v_{i,j}(\alpha) := a_{i,j} + \alpha \cdot b_{i,j}$ with some index $\varphi_{i'}(\alpha)$.

  ... some index $\varphi_i(\alpha)$ with some other index $\varphi_{i'}(\alpha)$.

Computation

Efficient enumeration of critical α-values is actually possible:

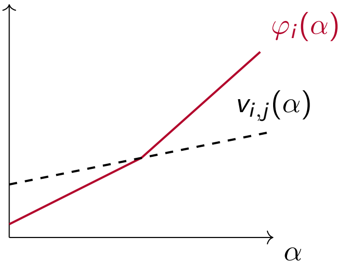
- Consider the index policy.

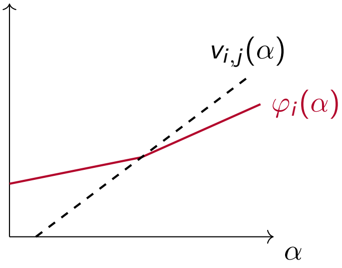
  $\rightarrow$ $\mathcal{A}$'s behavior changes only at intersections of...

  ... some $v_{i,j}(\alpha) := a_{i,j} + \alpha \cdot b_{i,j}$ with some index $\varphi_{i'}(\alpha)$.

  ... some index $\varphi_i(\alpha)$ with some other index $\varphi_{i'}(\alpha)$.

- The index φ_i as a function of α is monotone, convex, piece-wise linear with at most $2m+1$ linear segments.





General Contracts

Efficient computation of optimal **general** exploration contracts?
Open problem, but it's possible for some interesting subclasses!

General Contracts

Efficient computation of optimal **general** exploration contracts?
Open problem, but it's possible for some interesting subclasses!

- Non-intrinsic agent, where $a_{i,j} = 0$.
  - Standard in most contract design problems.
  - Optimal contract can be obtained from the index policy of $\mathcal{P}$.

General Contracts

Efficient computation of optimal **general** exploration contracts?
Open problem, but it's possible for some interesting subclasses!

- Non-intrinsic agent, where $a_{i,j} = 0$.
  - Standard in most contract design problems.
  - Optimal contract can be obtained from the index policy of $\mathcal{P}$.
- Binary boxes with two value-pairs and $a_{i,2} = b_{i,2} = 0$.

  [Bechtel, Dughmi, and Patel; EC 2022]

  - The first good outcome is taken.
  - Set up payments $t_{i,1}$ to induce optimal order of boxes.
  - Begin with minimum payments, then increase them greedily.

General Contracts

Efficient computation of optimal **general** exploration contracts?
Open problem, but it's possible for some interesting subclasses!

- Non-intrinsic agent, where $a_{i,j} = 0$.
  - Standard in most contract design problems.
  - Optimal contract can be obtained from the index policy of $\mathcal{P}$.
- Binary boxes with two value-pairs and $a_{i,2} = b_{i,2} = 0$.
  [Bechtel, Dughmi, and Patel; EC 2022]
  - The first good outcome is taken.
  - Set up payments $t_{i,1}$ to induce optimal order of boxes.
  - Begin with minimum payments, then increase them greedily.
- IID-boxes with a single positive $\mathcal{P}$-value.
  - Phase 1 where $\mathcal{A}$ immeadiately stops on a good $\mathcal{P}$-outcome.
  - Phase 2 where $\mathcal{P}$ *bets* that $\mathcal{A}$ does not stop at all.

Thank you!