Algorithmic Contract Design

Michal Feldman
Tel Aviv University

ADFOCS 2024

MPI Summer School

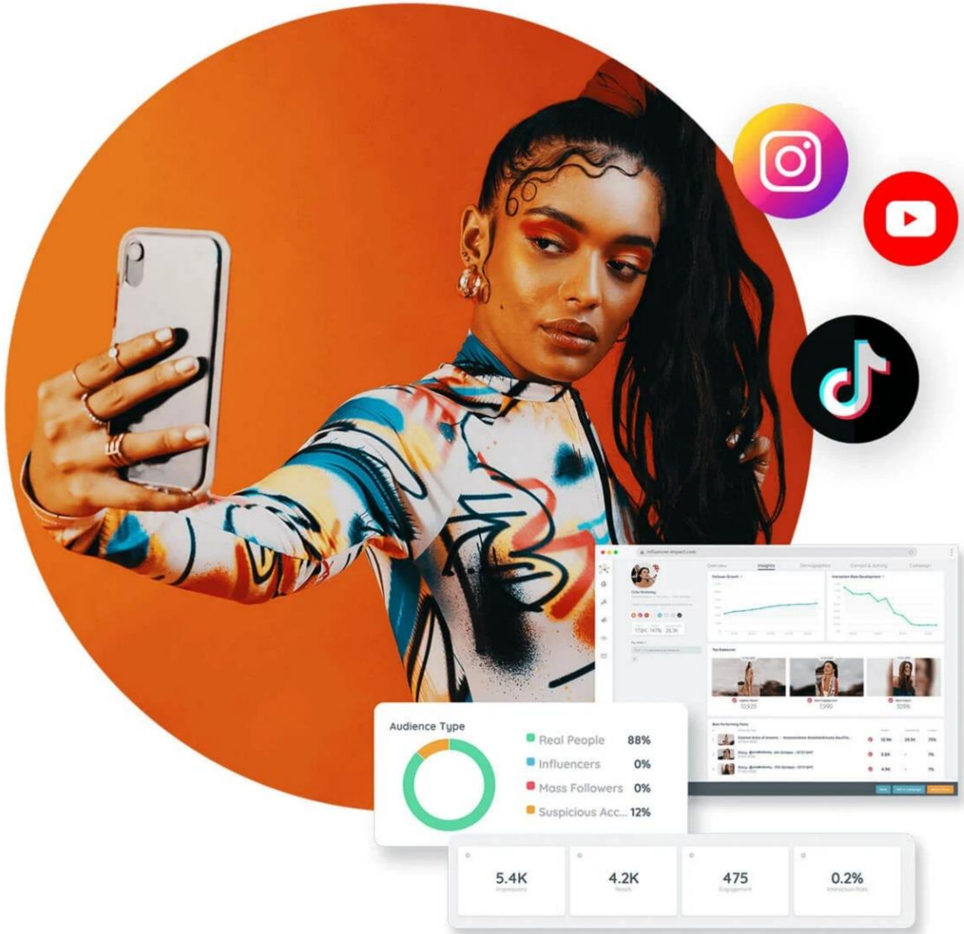
August 26, 2024

Saarbrücken, Germany

Example: Sponsored Content

- You want to pay an influencer to run a social media campaign
- Running a campaign requires effort
- You are buying a costly service with uncertain outcome (# views, etc.)

What/how should you pay the influencer for their effort?



Contract Design

One of the pillars of microeconomic theory
[Ross’73, Holmström’79]

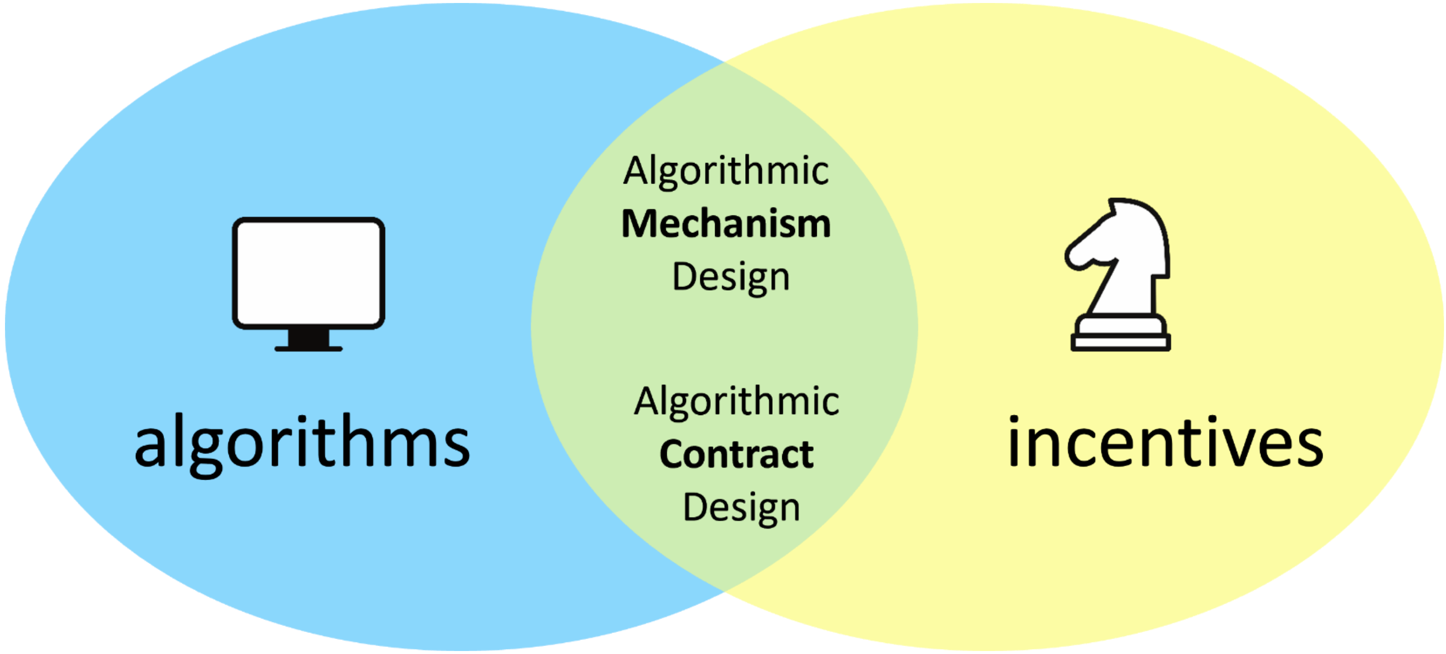
Holmström Hart

“The 2016 Nobel Prize in Economics was awarded Monday to Oliver Hart and Bengt Holmström for their work in contract theory — **developing a framework to understand agreements like insurance contracts, employer-employee relationships and property rights**.”

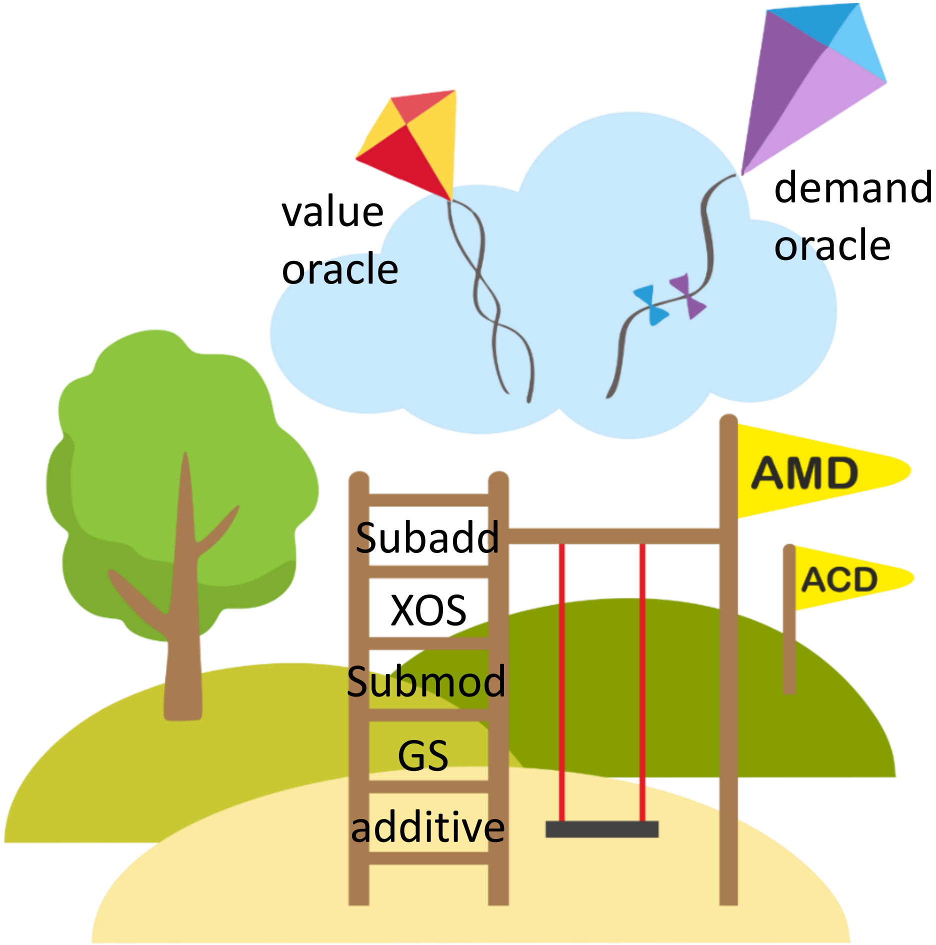
- As markets for services move online, they grow in scale and complexity (freelance services, legal services, marketing services, etc.)
- An algorithmic / computational approach is timely and relevant

Algorithms and Incentives

Hidden preferences



Hidden actions



[Nisan Ronen STOC'99]
[Lehmann Lehmann Nisan EC'01]
...

Algorithmic Contract Design: an Emerging Frontier

- Simple vs optimal contracts: [Carroll AER'15], [Duetting Roughgarden & Talgam-Cohen EC'19], [Alon Duetting Li Talgam-Cohen EC'23]
- Combinatorial contracts: [Babaioff Feldman Nisan Winter '12 (EC'06)], [Lavi & Shamash EC'19], [Duetting Roughgarden & Talgam-Cohen SODA'20], [Duetting Ezra Feldman & Kesselheim FOCS'21], [Alon Lavi Shamash & Talgam-Cohen EC'21], [Duetting Ezra Feldman & Kesselheim STOC'23], [Castiglioni et al. EC'23], [Duetting Feldman & Gal-Tzur SODA'24], [Deo-Campo Vuong et al. SODA'24], [Ezra Feldman Schlesinger ITCS'24], [Cacciamani et al. EC'24]
- Learning contracts: [Ho Slivkins & Vaughn EC'14], [Cohen Deligkas & Koren SAGT'22], [Zhu et al. EC'23], [Duetting Guruganesh Schneider & Wang ICML'23], [Chen et al. EC'24]
- Typed contracts: [Guruganesh Schneider & Wang EC'21], [Alon Duetting & Talgam-Cohen EC'21], [Castiglioni et al. EC '21], [Castiglioni et al. EC '22], [Guruganesh Schneider & Wang EC'23]
- Contract design for social good: [Li Immorlica & Lucier WINE'21], [Ashlagi Li & Lo Management Science'23]
- Ambiguous contracts: [Duetting Feldman Peretz Samuelson EC'23]

The Algorithmic/Computational Lens

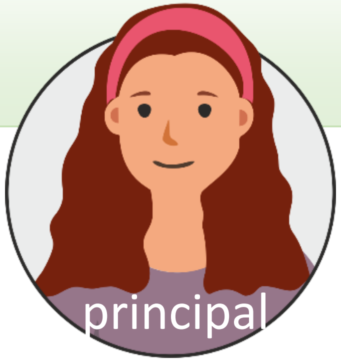
- The algorithmic lens has been traditionally useful
  - Reveals structure
  - Identifies tractability frontier
  - Informs the design of simple mechanisms

- Many examples in **Algorithmic Mechanism Design**
  - E.g., greedy algorithms, substitutes as a frontier of tractability, submodularity as simplicity frontier, hardness of NE, …

Plan for this Talk

- Part 1: The Fundamentals
  - The principal-agent model
  - Optimal contracts
  - Linear contracts
- Part 2: Combinatorial Contracts
  - Multiple actions [Duetting Ezra Feldman Kesselheim FOCS’21] [Duetting Feldman Gal-Tzur SODA’24], [Deo-Campo Vuong et al. SODA’24], [Ezra Feldman Schlesinger ITCS’24]
  - Multiple agents [Babaioff Feldman Nisan Winter ‘12 (EC’06)] [Duetting Ezra Feldman & Kesselheim STOC’23]
  - Combined problem [Duetting Ezra Feldman Kesselheim 2024]

The Principal-Agent Problem



Defines contract $t \in \mathbb{R}_+^m$

Chooses action $i \in [n]$

Gets reward $r_j \in \mathbb{R}_+$ for outcome $j \in [m]$
with probability f_{ij}

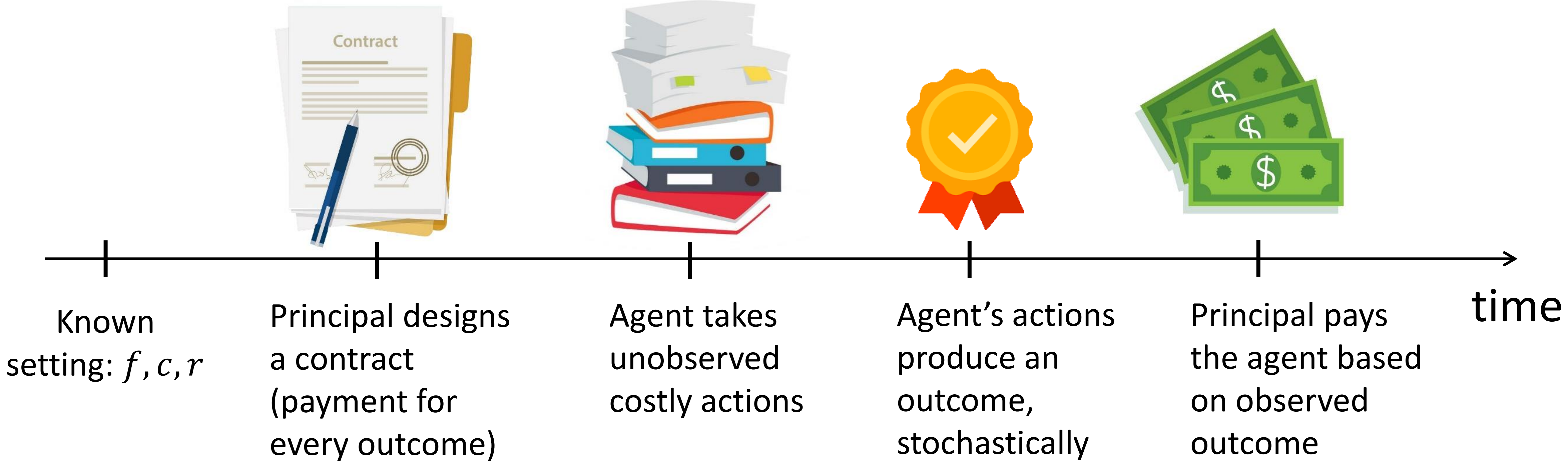
Incurs cost c_i

Expected payment: Pays $T_i = \sum_{j\in[m]} f_{ij} t_j$ Receives T_i

Expected utility: $R_i - T_i = \sum_{j\in[m]} f_{ij} r_j - T_i$ $T_i - c_i$

Defining features: hidden action, stochastic outcome, limited liability

Timing and Objective



Objective: maximize the expected utility of the principal

Example

	c_i	$r_0 = 0$	$r_1 = 2$	$r_2 = 2$
Action 0	0	1	0	0
Action 1	¼	½	½	0
Action 2	¼	½	0	½
Action 3	1	0	½	½

R_i
0
1
1
2

Optimal contract:

- Incentivize action 1
- Contract: $t = (0, 0.5, 0)$
- Expected payment: $T_1 = 0.25$
- Expected principal's utility = $R_1 - T_1 = 1 - 0.25 = 0.75$

Incentivizing action 3:

- Incentivize action 3
- Contract: $t = (0, 1.5, 1.5)$
- Expected payment: $T_3 = 1.5$
- Expected principal's utility = $R_3 - T_3 = 2 - 1.5 = 0.5$

Example

Example 2.1 (A simple principal-agent setting). *Consider a principal-agent setting with three actions $i = 1, 2, 3$ with costs, rewards, and probabilities as specified in the following table:*

	$r_1 = 0$	$r_2 = 1$	$r_3 = 7$	*cost*
action 1:	1	0	0	$c_1 = 0$
action 2:	0	$1/2$	$1/2$	$c_2 = 1$
action 3:	0	$1/6$	$5/6$	$c_3 = 2$

The expected rewards corresponding to the three actions are $R_1 = 0$, $R_2 = 1/2 \cdot 1 + 1/2 \cdot 7 = 4$, and $R_3 = 1/6 \cdot 1 + 5/6 \cdot 7 = 6$. Their expected welfares are $W_1 = R_1 - c_1 = 0$, $W_2 = R_2 - c_2 = 4 - 1 = 3$ and $W_3 = R_3 - c_3 = 6 - 2 = 4$. Consider the contract $\mathbf{t} = (0, 1, 3)$. The expected payment for action 1 under this contract is $T_1 = 0$, for action 2 it is $T_2 = 1/2 \cdot 1 + 1/2 \cdot 3 = 2$, and for action 3 it is $T_3 = 1/6 \cdot 1 + 5/6 \cdot 3 = 8/3$. The agent's expected utility is therefore maximimzed by action 2, which yields an expected utility of $T_2 - c_2 = 2 - 1 = 1$, compared to an expected utility of $T_1 - c_1 = 0$ for action 1 and an expected utility of $T_3 - c_3 = 8/3 - 2 = 2/3$ for action 3. The principal's expected utility under this contract is $R_2 - T_2 = 4 - 2 = 2$.

Key Results: Optimal Contracts

Theorem (folklore): Optimal contract can be computed in $poly(n, m)$ time through linear programming.

MIN-PAY problem

Input: Contract setting (f, c, r); an action i

Output: Minimum T_i that incentivizes action i

Observations:

- LP solvable
- Optimal contract solvable via n MIN-PAY problems

$$\min T_i$$

$$\text{s.t. } T_i - c_i \geq T_{i'} - c_{i'} \quad \forall i' \neq i \quad \text{(IC)}$$

Key Results: Optimal Contracts

Theorem (folklore): Optimal contract can be computed in $poly(n, m)$ time through linear programming.

But optimal contracts have been criticized:

- As solutions to LPs they are opaque, and lack structure
- They may be non-monotone

Important exception: With only two outcomes "success" and "failure", Linear (comission-based) contracts that set $t_j = \alpha \cdot r_j$ for all $j \in [m]$ are optimal

Example of Non-Monotonicity

Example 3.1 (Non-monotone optimal contract). *Consider the principal-agent setting depicted in the following table:*

	$r_1 = 0$	$r_2 = 3$	$r_3 = 9$	$r_4 = 12$	*cost*
action 1	1	0	0	0	$c_1 = 0$
action 2:	0	$1/3$	0	$2/3$	$c_2 = 1$
action 3:	0	0	$1/3$	$2/3$	$c_3 = 2$

In this setting the unique optimal contract for action $i \in \{1, 2, 3\}$ pays just enough for outcome i to cover the action's cost and nothing for the other two outcomes. The optimal contract is the best contract for incentivizing action 3, which is $\mathbf{t} = (0, 0, 6, 0)$. This contract is non-monotone as $r_3 < r_4$ but $t_3 > t_4$. In this example the non-monotonicity is caused by the fact that outcome 4 — the one with the highest reward — doesn't help differentiate between the two actions, and so it doesn't make sense for the principal to pay for that outcome.

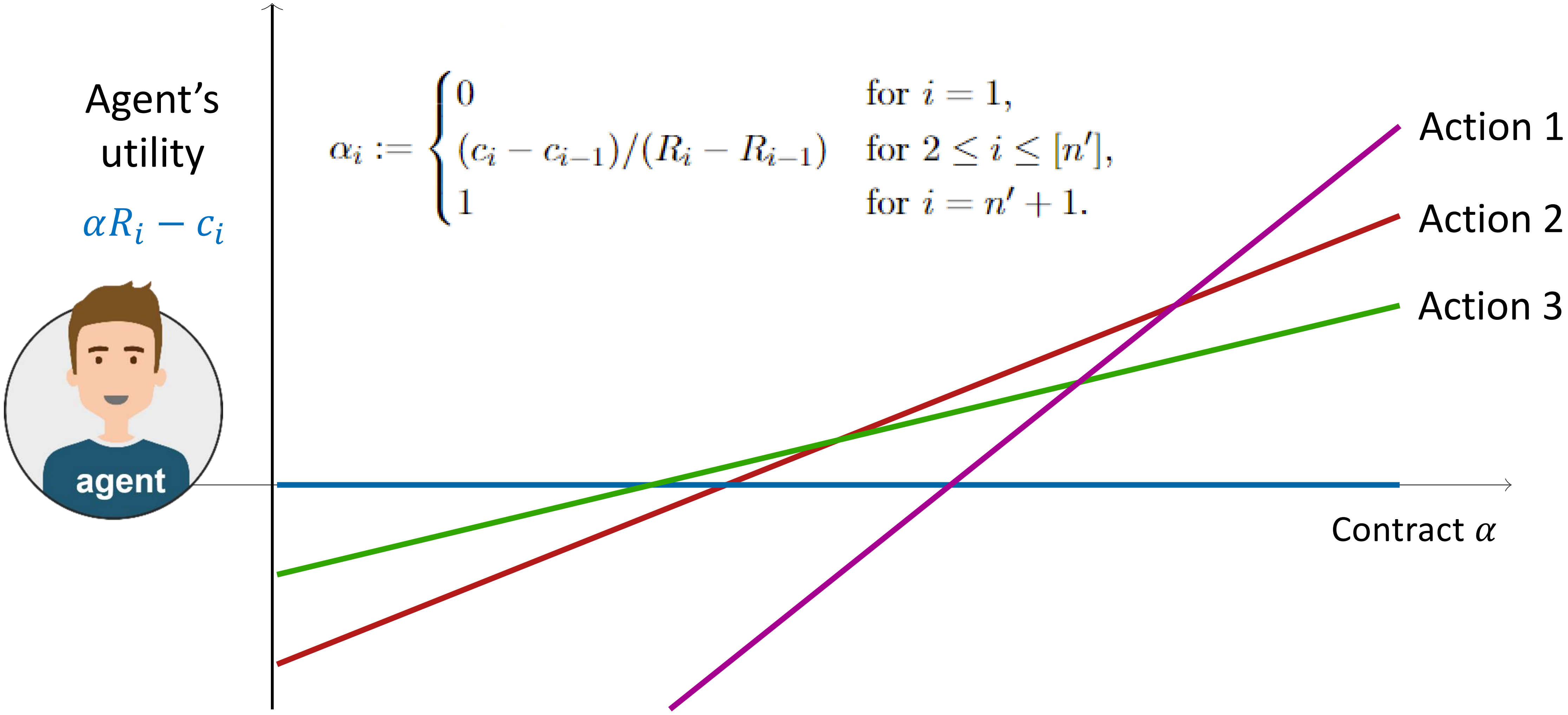
Key Results: Linear Contracts

Theorem [Duetting Roughgarden Talgam-Cohen‘19]: Linear contracts achieve a $\Theta(n)$ approximation to optimal contracts.

Theorem [Duetting Roughgarden Talgam-Cohen‘19]: Linear contracts are max-min optimal when only the expected rewards of the actions are known.

- Provides easy to interpret, ”robust optimization”-style analogue of [Carroll’15]
- In [Carroll’15] principal knows subset of actions, actual actions can be any superset

Tool: Upper Envelope (Agent's Perspective)

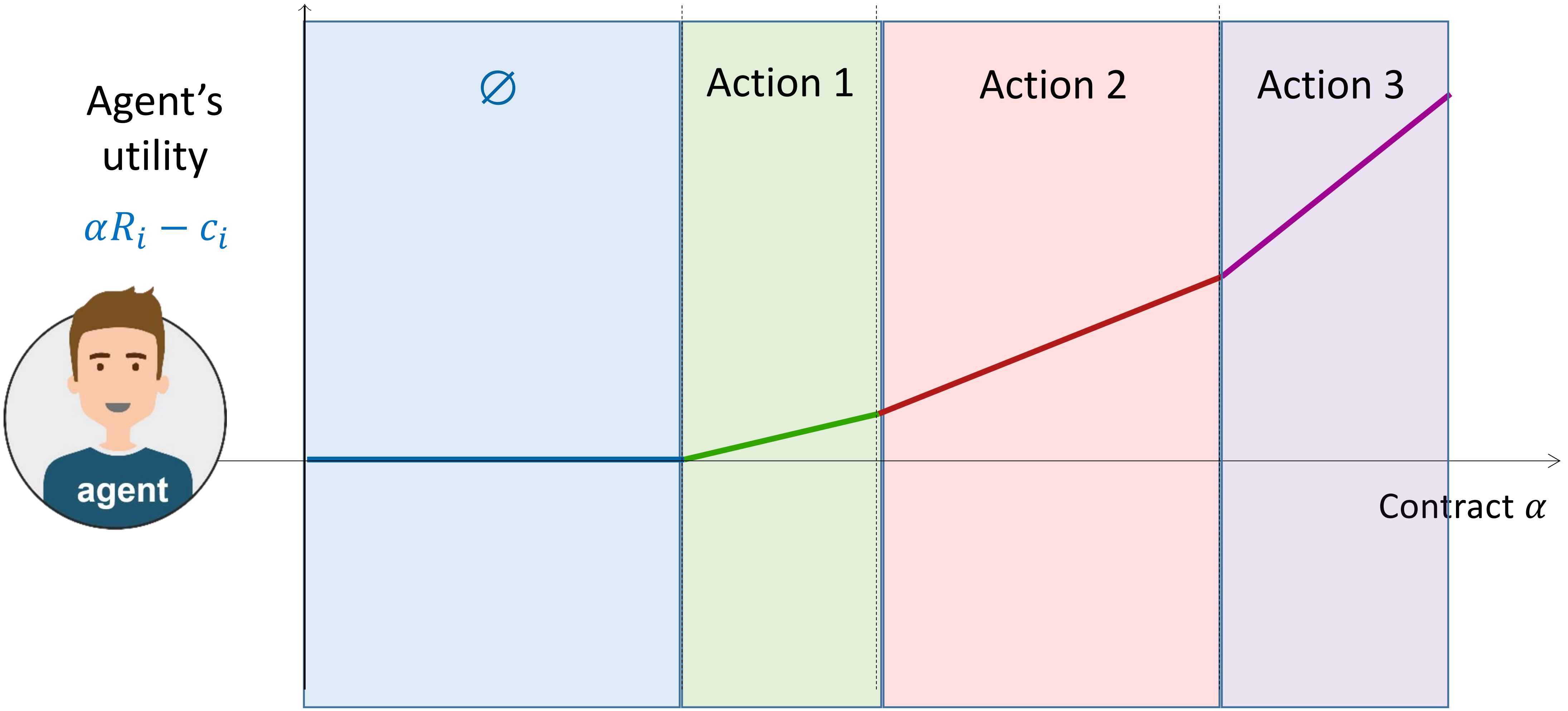


$$\alpha_i := \begin{cases} 0 & \text{for } i = 1, \\ (c_i - c_{i-1})/(R_i - R_{i-1}) & \text{for } 2 \le i \le [n'], \\ 1 & \text{for } i = n' + 1. \end{cases}$$

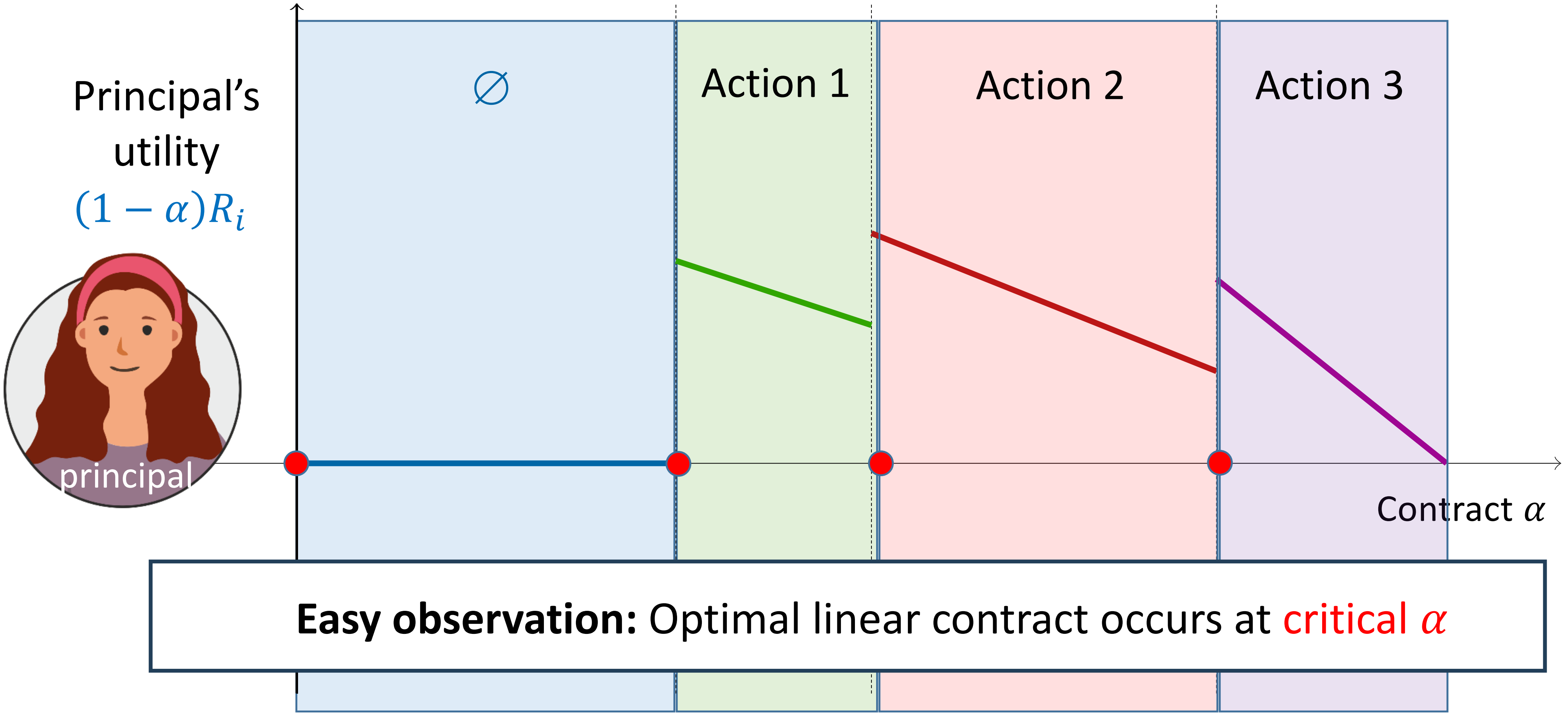
Tool: Upper Envelope (Agent's Perspective)

Agent's utility

$\alpha R_i - c_i$



Tool: Upper Envelope (Principal's Perspective)

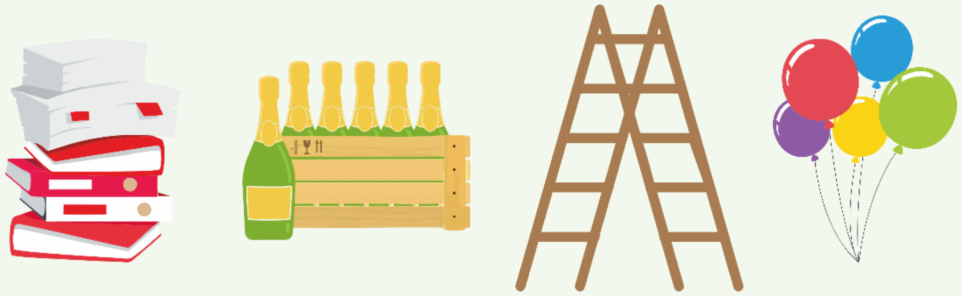


Easy observation: Optimal linear contract occurs at critical α

Rest of the Talk

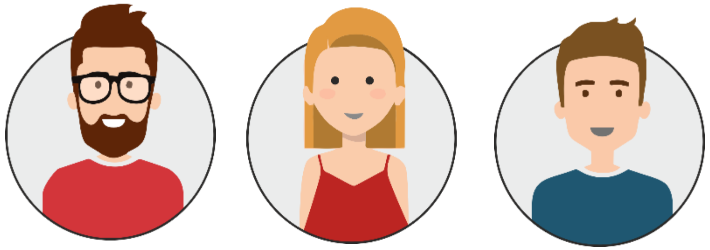
- Part 1: The Fundamentals
  - The principal-agent model
  - Optimal contracts
  - Linear contracts
- Part 2: Combinatorial Contracts
  - Multiple actions [Duetting Ezra Feldman Kesselheim FOCS'21] [Duetting Feldman Gal-Tzur SODA'24], [Deo-Campo Vuong et al. SODA'24], [Ezra Feldman Schlesinger ITCS'24]
  - Multiple agents [Babaioff Feldman Nisan EC'12] [Duetting Ezra Feldman & Kesselheim STOC'23]
  - Combined problem [Duetting Ezra Feldman Kesselheim 2024]

Sources of Complexity in Contract Design



Multiple actions

[Duetting Ezra Feldman & Kesselheim FOCS'21] [Duetting Feldman & Gal-Tzur SODA'24], [Deo-Campo Vuong et al. SODA'24], [Ezra Feldman Schlesinger ITCS'24]

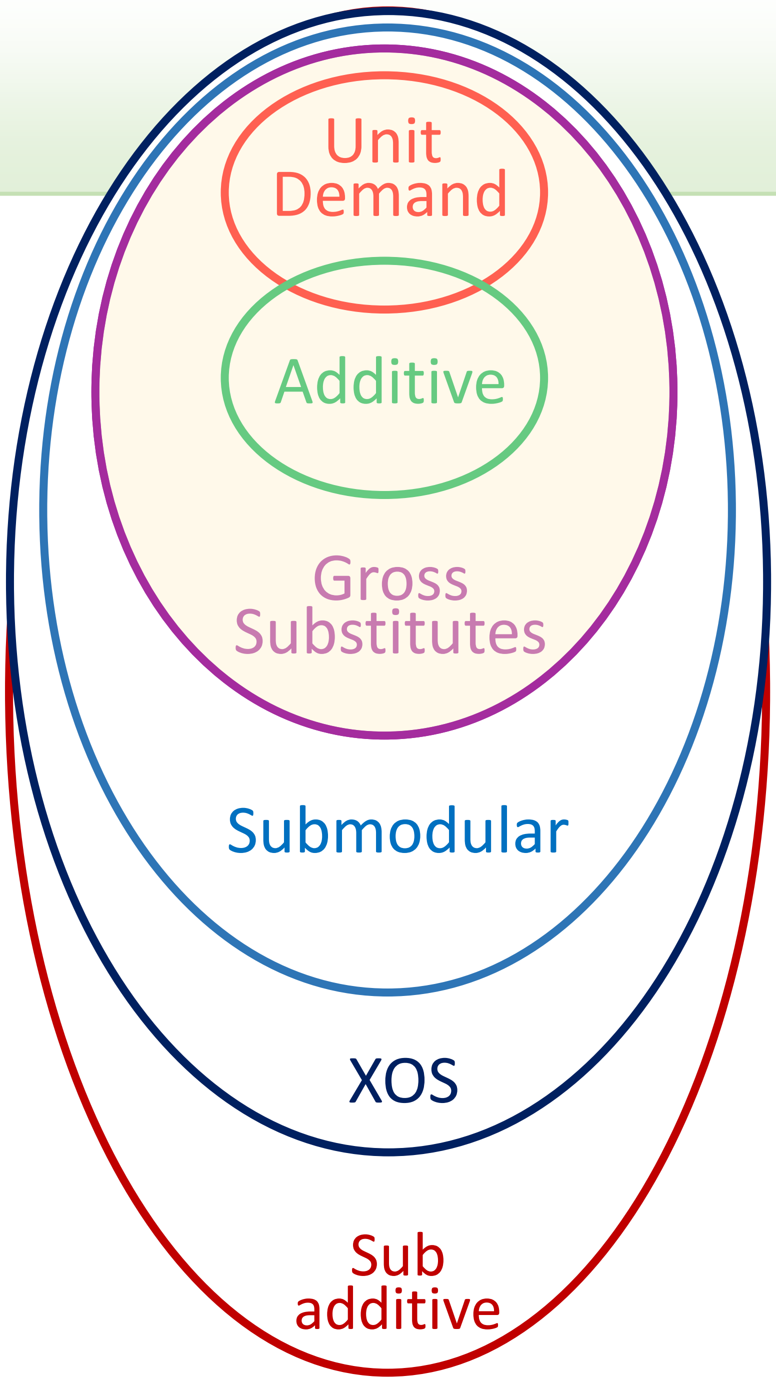


Multiple agents

[Babaioff Feldman Nisan Winter '12 (EC'12)] [Duetting Ezra Feldman & Kesselheim STOC'23] [Ezra Feldman Schlesinger ITCS'24]

Single Agent, Many Actions [DEFK'21]

- n actions $A = \{1, \dots, n\}$, agent chooses a set S
- $c(i) \geq 0$: cost of action i
- $c(S) = \sum_{i \in S} c(i)$ [additive cost]
- Binary outcome: {0,1} (reward 1 for success)
- f: $2^A \to [0,1]$ success probability function
  - $f(S)$: success probability for actions $S \subseteq A$
  - Not necessarily additive

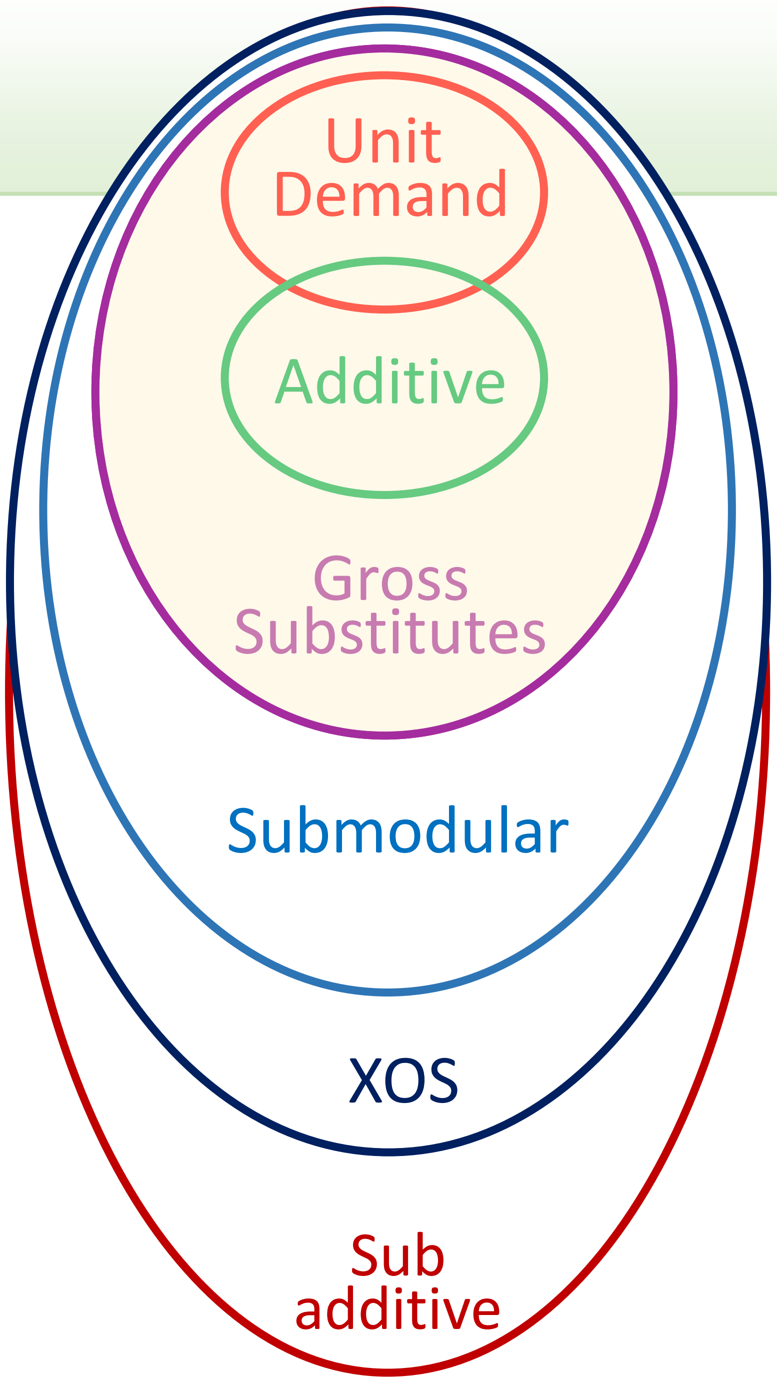


Single Agent, Many Actions [DEFK'21]

- n actions $A = \{1, \dots, n\}$, agent chooses a set S
- $c(i) \geq 0$: cost of action i
- $c(S) = \sum_{i \in S} c(i)$ [additive cost]
- Binary outcome: {0,1} (reward 1 for success)
- $f: 2^A \to [0,1]$ success probability function

Submodular: for every $S \subseteq T, j \notin T$, $f(j \mid S) \geq f(j \mid T)$
[decreasing marginal value]

Subadditive: for every S, T, $f(S) + f(T) \geq f(S \cup T)$

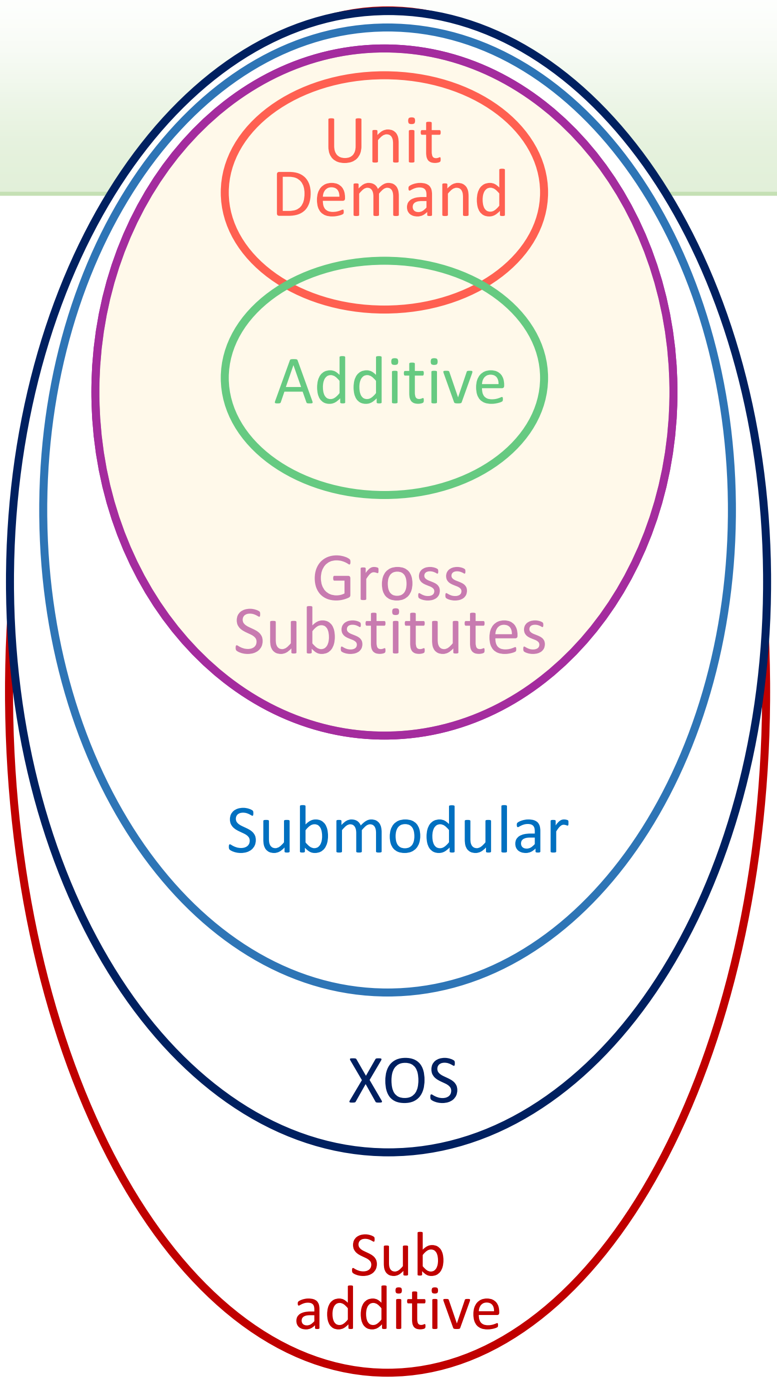


Single Agent, Many Actions [DEFK'21]

- n actions $A = \{1, \ldots, n\}$, agent chooses a set S
- $c(i) \geq 0$: cost of action i
- $c(S) = \sum_{i \in S} c(i)$ [additive cost]
- Binary outcome: $\{0,1\}$ (reward 1 for success)
- $f: 2^A \rightarrow [0,1]$ success probability function

Demand set $D(f, p)$: a set S maximizing $f(S) - \sum_{i \in S} p_i$

Gross substitutes: Suppose $q \geq p$. Then, for every $i \in D(f,p)$ s.t. $p_i = q_i$, it holds that $i \in D(f,q)$

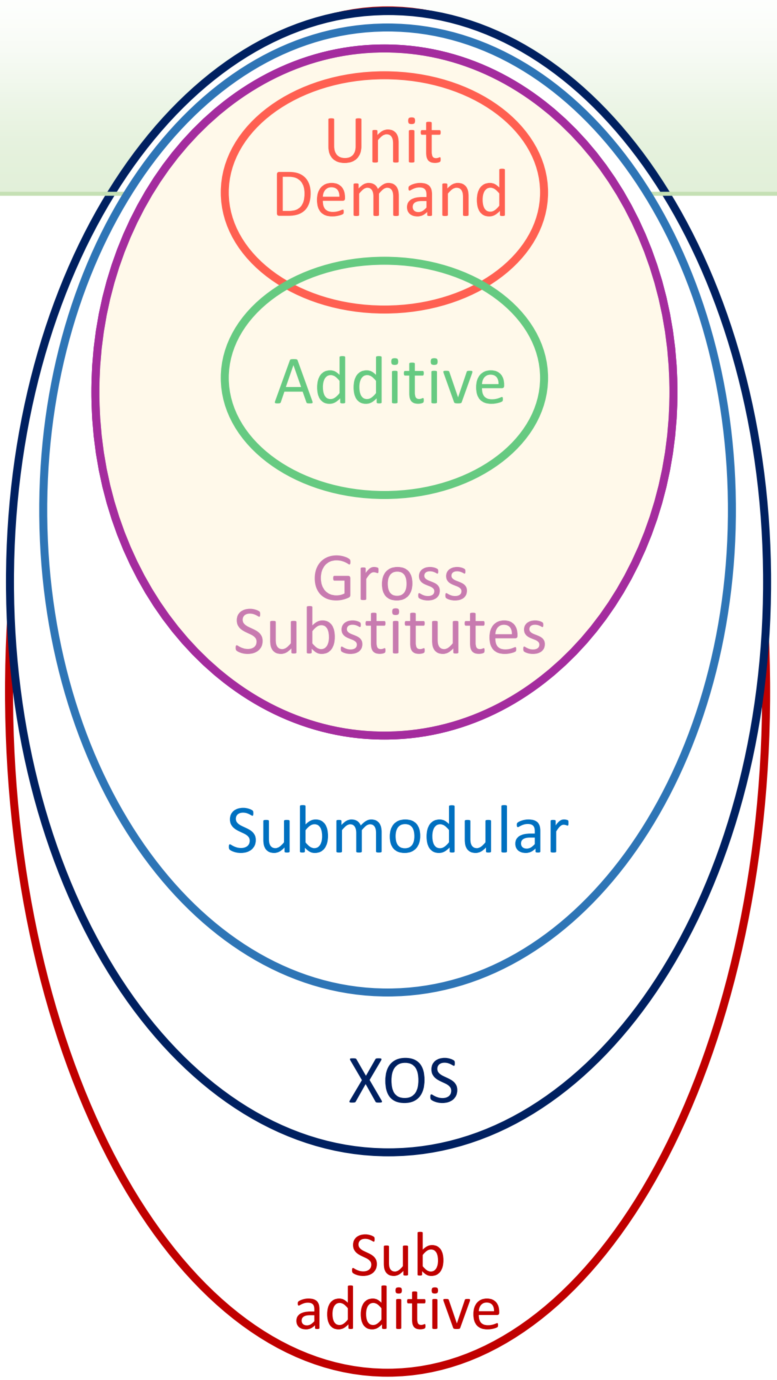


Optimization Problem

- n actions $A = \{1, \ldots, n\}$, agent chooses a set S
- $c(i) \geq 0$: cost of action i
- $c(S) = \sum_{i \in S} c(i)$ [additive cost]
- Binary outcome: {0,1} (reward 1 for success)
- $f: 2^A \rightarrow [0,1]$ success probability function

Optimal Contract Problem:
Find α that maximizes $(1-\alpha)f(S_\alpha)$ [principal's utility]
where S_α maximizes $\alpha f(S) - c(S)$ [agent's utility]

Value Oracle: Receives S, returns $f(S)$.

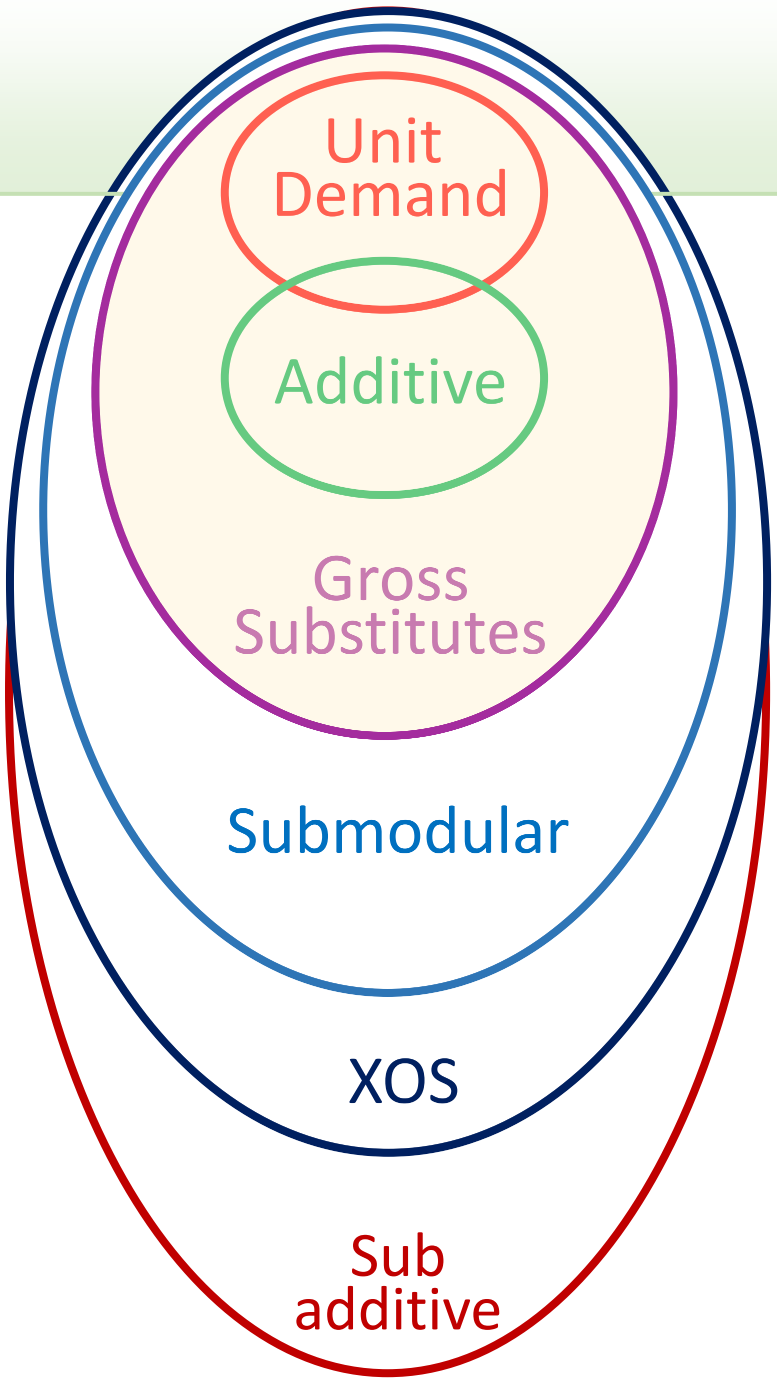


Oracle Access

Optimal Contract Problem:
Find α that maximizes $(1-\alpha)f(S_\alpha)$ [principal's utility]
where S_α maximizes $\alpha f(S) - c(S)$ [agent's utility]

Value Oracle: Receives S, returns $f(S)$.

Demand Oracle: Given "prices" $p_1, \ldots, p_n$, return sets S maximizing $f(S) - \Sigma_{i \in S} p_i$.



Main Results

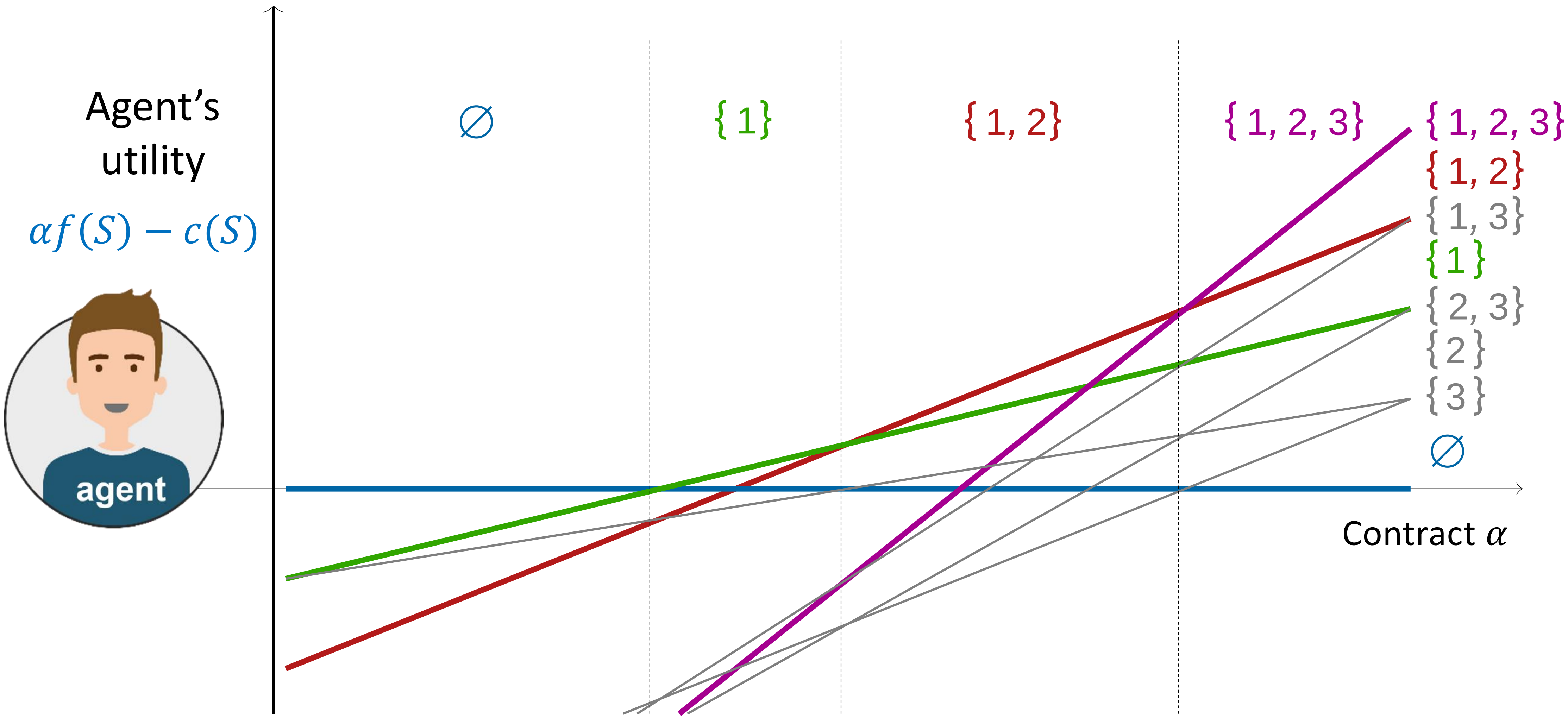
Theorem [Duetting Ezra Feldman Kesselheim'21]:

- A polynomial-time algorithm for gross substitutes functions (with value oracle access to f)
- For submodular functions (i.e., decreasing marginal value), it is NP-hard to compute the optimal contract

Gross substitutes constitutes a frontier, similar to:

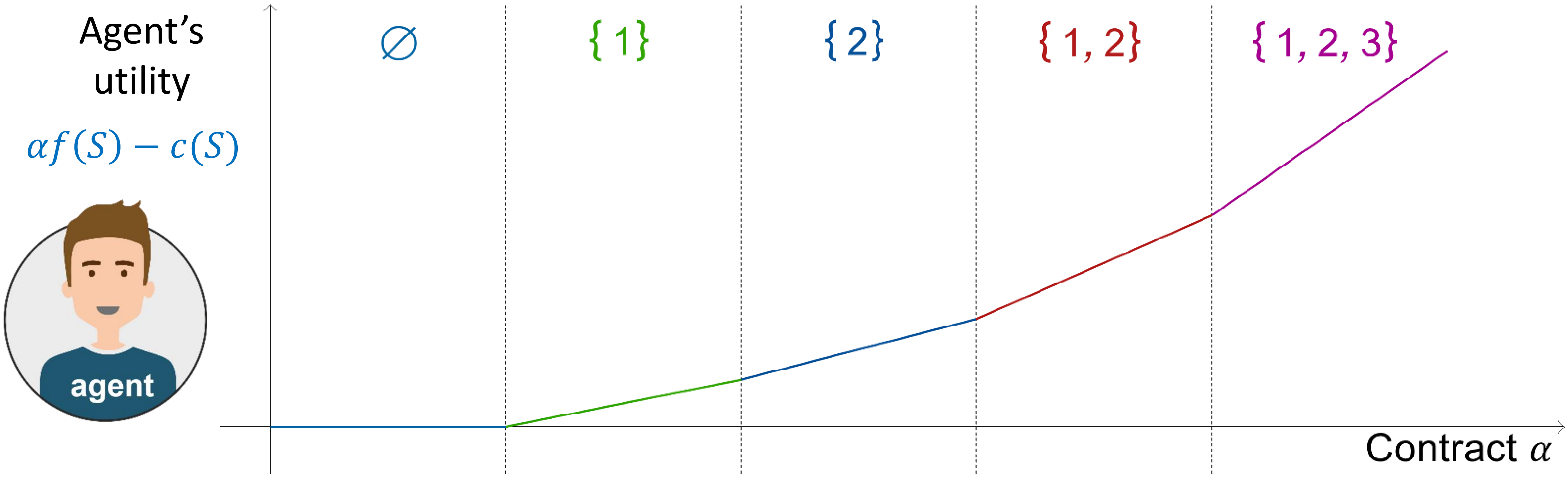
- Welfare maximization tractability in combinatorial auctions [Nisan Segal 2006]
- Market equilibrium existence [Kelso Crawford 1982, Gul Stacchetti 1999]

Upper-Envelope Approach



[Figure is for additive f]

Upper-Envelope Approach



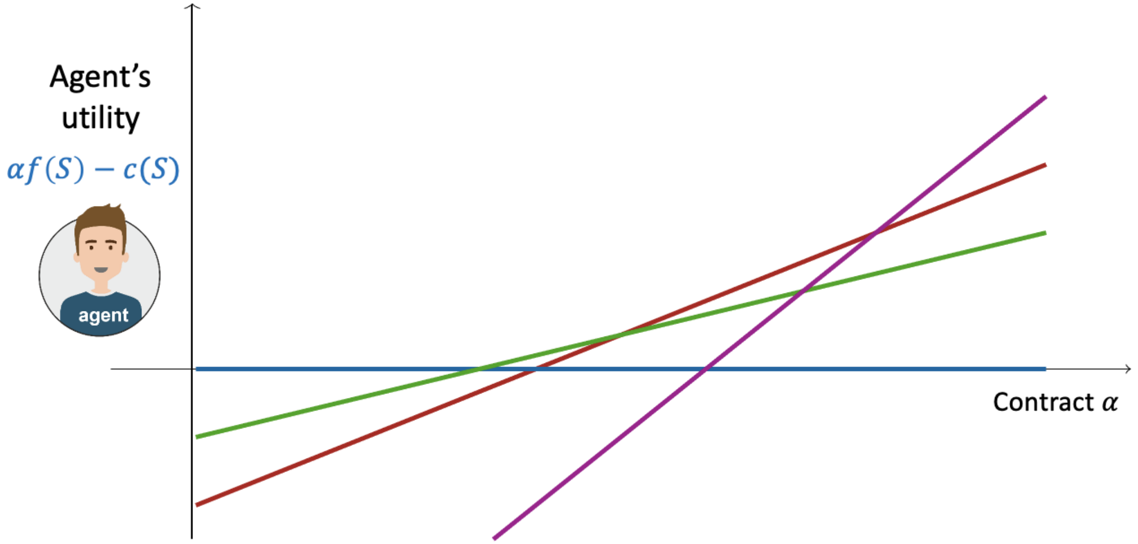
[Figure is for gross substitutes f]

Idea for an Algorithm

- **Recall:** Can restrict attention to set of critical α's
  (i.e., transition points of agent's best response)
- **Naïve algorithm:** Go over all critical α's and take the best; requires:
  - computing agent's best response
  - computing next critical α
  - an upper bound on number of critical α's

Theorem: For gross substitutes f, this yields a polynomial-time algorithm.

Step 0: The Agent's Best Response Problem



The agent's problem: given α,

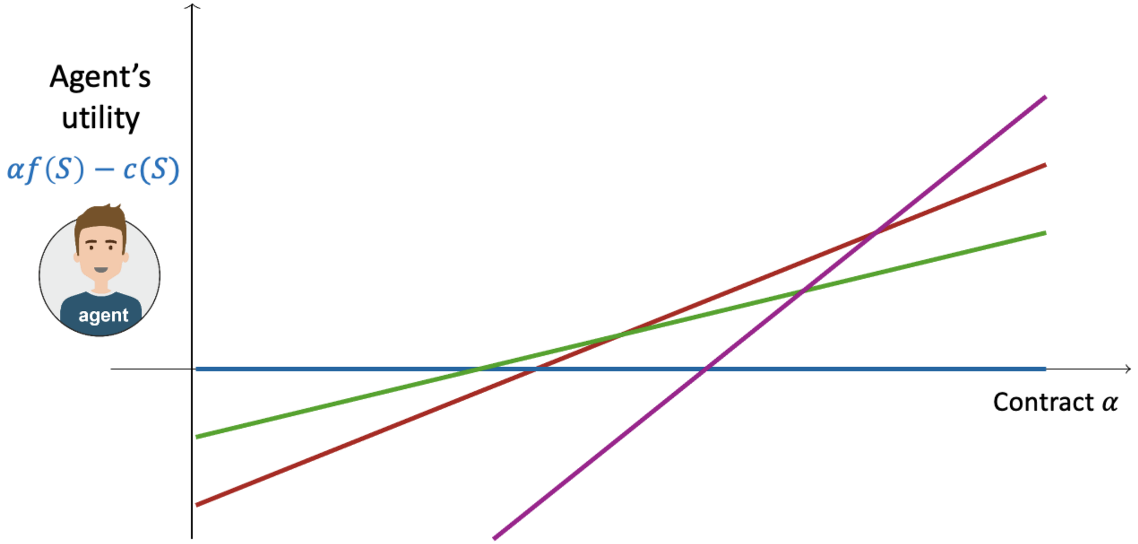
find S that maximizes $\alpha f(S) - c(S)$

$$\Leftrightarrow$$

find S that maximizes $f(S) - \frac{1}{\alpha} c(S)$

Demand set at "prices" c/α (in markets for goods)

Step 0: The Agent's Best Response Problem



The agent's problem: given α,

find S that maximizes $\alpha f(S) - c(S)$

$\Leftrightarrow$

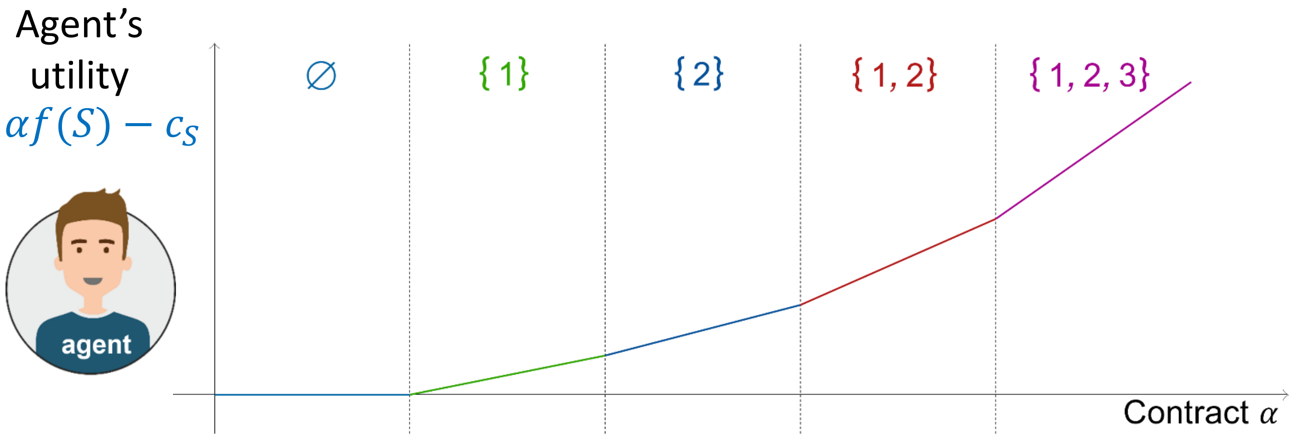
find S that maximizes $f(S) - \frac{1}{\alpha} c(S)$

- Demand set $D(f, p)$: a set S maximizing utility $\mathrm{u}(S) = f(S) - \sum_{j \in S} p_j$
- Key property of gross substitutes:
  - GREEDY algorithm solves the demand set problem (add element with maximal marginal utility) [e.g., Paes Leme 2017]

Step 1: Next Critical α

- Fixing tie-breaking, we get an ordered demand set $S_\alpha = (a_1, a_2, \dots, a_d)$
- Let $S_\alpha = (a_1, a_2, \dots, a_d)$ and $S_{\alpha'}$ be respective demand sets of α, α'
- Either: $S_\alpha[i] \neq S_{\alpha'}[i]$ for some $i \leq d$, or $|S_{\alpha'}| > d$
- Suffices to consider poly-many potential values for α' (for each action and index), and take the smallest one that is larger than α

Step 2: Poly-Many Critical α's



The agent's problem: given α,

find S that maximizes $\alpha f(S) - c(S)$

$$\Leftrightarrow$$

find S that maximizes $f(S) - \frac{1}{\alpha} c(S)$

Key Lemma: at each critical point:

- an action is added to S, or
- an action from S is replaced by one with higher cost

(obtained by perturbing cost, so that GREEDY has at most one tie-breaking)

Potential argument:

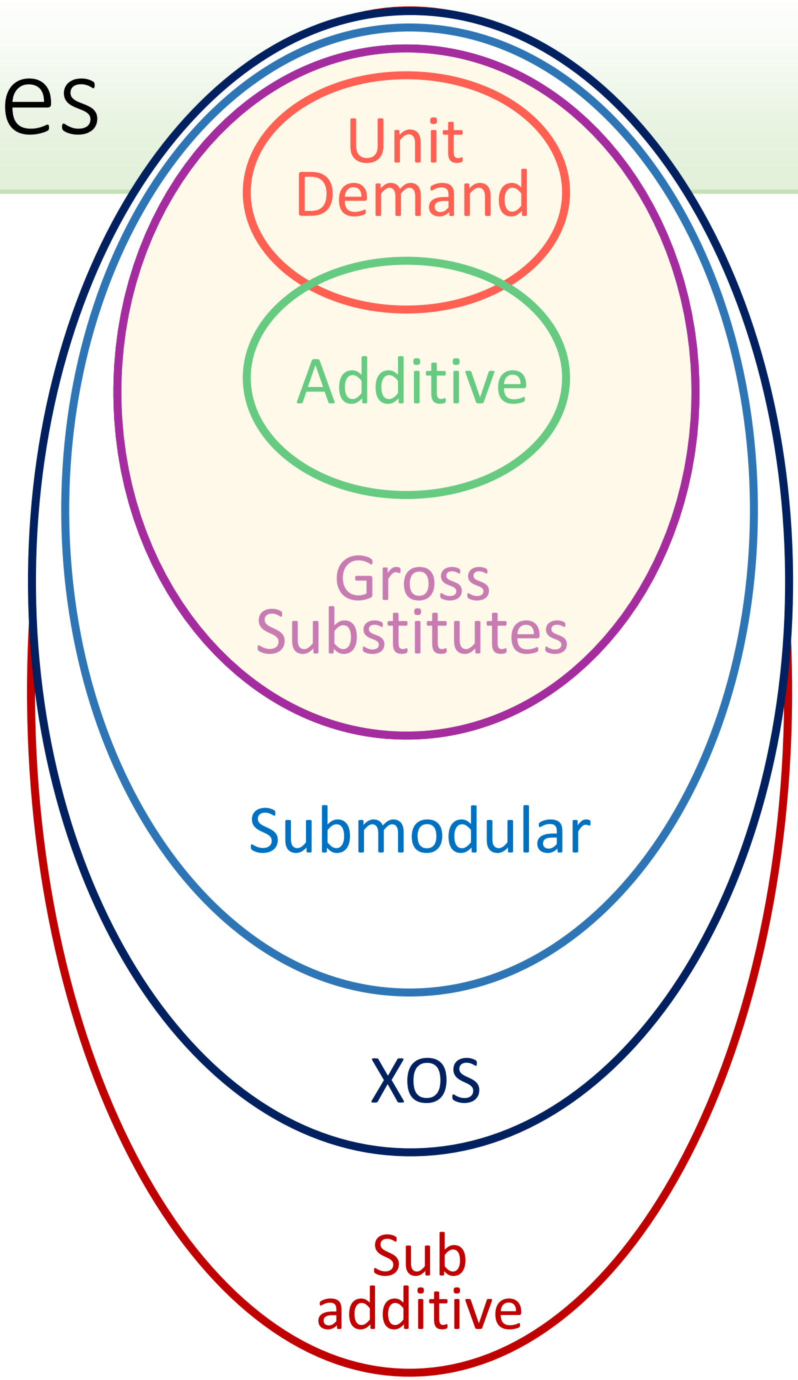
- Reorder actions: $c(a_1) < \cdots < c(a_n)$
- Define $\phi(a_i) = i, \phi(S) = \sum_{a \in S} \phi(a)$
- ϕ is an integer $\leq n(n+1)/2$, which increases at every critical α
- Conclusion: $O(n^2)$ critical points for GS
- (this is tight)

Beyond Gross Substitutes

Submodular: $f(i \mid S) \geq f(i \mid T)$ for $S \subseteq T, j \notin T$
(decreasing marginal value)

XOS: maximum over additive
(aka: fractionally subadditive)

Subadditive: $f(S) + f(T) \geq f(S \cup T)$



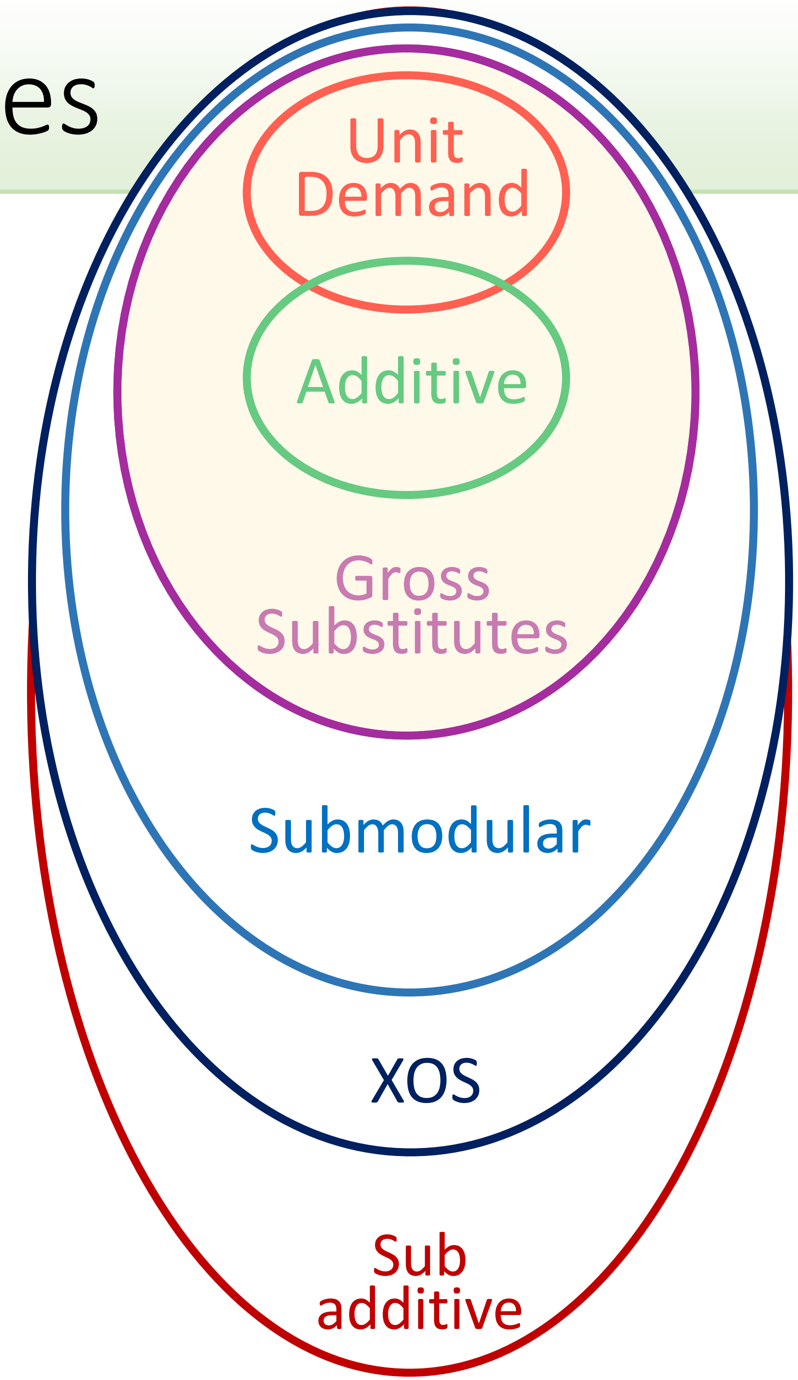
Beyond Gross Substitutes

Inapproximability results [Ezra F Schlesinger'24]:

- No constant-approximation for submodular rewards with value queries (assuming P $\neq$ NP)
- No better than $\Omega(\sqrt{n})$ approximation for XOS rewards with value queries (assuming P $\neq$ NP)

With demand oracle access (given action "prices" $p_1, \dots, p_n$, return S maximizing $f(S) - \sum_{i \in S} p_i$):

- FPTAS for any f [Duetting Ezra F Kesselheim '24]
- But not OPT [Duetting F Gal-Tzur Rubinstein '24]



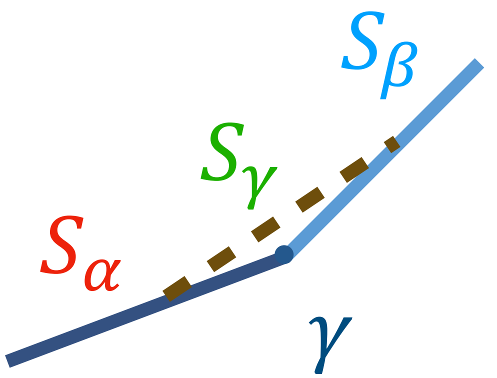
Beyond Complement-Free

- **Recall Naïve algorithm:** Go over all critical α's and take the best; requires:
  - computing agent's best response
  - computing next critical α
  - an upper bound on number of critical α's

Theorem: For supermodular f, this yields a polynomial-time algorithm.

Enumerating all Critical Values

- Theorem [DFG'24]: For every f, c, a demand oracle (i.e., agent's BR) is sufficient for enumerating all critical values
- Algorithm: For a segment $[\alpha, \beta]$, use the oracle to get S_α and S_β.
  - If $S_\alpha = S_\beta$: the utility is linear in $[\alpha, \beta]$
  - Otherwise, query again at $\gamma = \frac{c(S_\alpha) - c(S_\beta)}{f(S_\alpha) - f(S_\beta)}$
  - If $S_\gamma = S_\beta$: the utility is linear in $[\alpha,\gamma)$ and in $[\gamma,\beta]$
  - Otherwise, there are more than 2 linear pieces; solve recursively for $[\alpha,\gamma]$ and $[\gamma,\beta]$



- Proof: by induction on the number of critical values in the segment
- Upshot: For every monotone f, c, a demand oracle and poly-many critical values are sufficient to find the optimal contract

The Agent's Best Response Problem

- Agent's utility function: $\mathrm{u_A}(\alpha, S) = \alpha f(S) - c(S)$
- If f is supermodular, then $\mathrm{u_A}(\alpha,\cdot)$ is supermodular
- Maximizing $\mathrm{u_A}(\alpha,\cdot)$ is equivalent to minimizing $-\mathrm{u_A}(\alpha,\cdot)$, which is submodular: known to admit a poly-time algorithm
- Note: this argument holds even if c is submodular

Poly-Many Critical α's

Claim 4.2. *Let c be a monotone submodular cost function and f a monotone supermodular reward function, then for any two contracts $\alpha < \alpha'$ and two corresponding sets in the agent's demand S_α, $S_{\alpha'}$ it holds that $S_\alpha \subseteq S_{\alpha'}$.*

Proof. If $S_\alpha = S_{\alpha'}$ the claim obviously hold. Otherwise, assume that $S_{\alpha'}$ is a maximal best-response for contract α' (this is in line with our tie-breaking assumption), and also that $S_\alpha \setminus S_{\alpha'} = R$ is such that $R \neq \emptyset$, we will show that a contradiction is reached. By the fact that S_α is optimal for α, it must be that

$$u_a(\alpha, R \mid S_\alpha \cap S_{\alpha'}) = u_a(\alpha, S_\alpha) - u_a(\alpha, S_\alpha \cap S_{\alpha'}) \geq 0$$

By the supermodularity of f and submodularity of c it holds that $f(R \mid S_\alpha \cap S_{\alpha'}) \leq f(R \mid S_{\alpha'})$ and $c(R \mid S_\alpha \cap S_{\alpha'}) \geq c(R \mid S_{\alpha'})$. Putting everything together we get

$$\begin{aligned} u(\alpha', R \mid S_{\alpha'}) &= \alpha' f(R \mid S_{\alpha'}) - c(R \mid S_{\alpha'}) \\ &\geq \alpha' f(R \mid S_\alpha \cap S_{\alpha'}) - c(R \mid S_\alpha \cap S_{\alpha'}) \\ &\geq \alpha f(R \mid S_\alpha \cap S_{\alpha'}) - c(R \mid S_\alpha \cap S_{\alpha'}) \\ &= u(\alpha, R \mid S_\alpha) \\ &\geq 0, \end{aligned}$$

where the second inequality follows from the monotonicity of f, which imply $f(R \mid S_\alpha \cap S_{\alpha'}) \geq 0$. Thus, we can add R to $S_{\alpha'}$ while not losing utility, contradicting its maximality. □

Multipe Actions: Overview

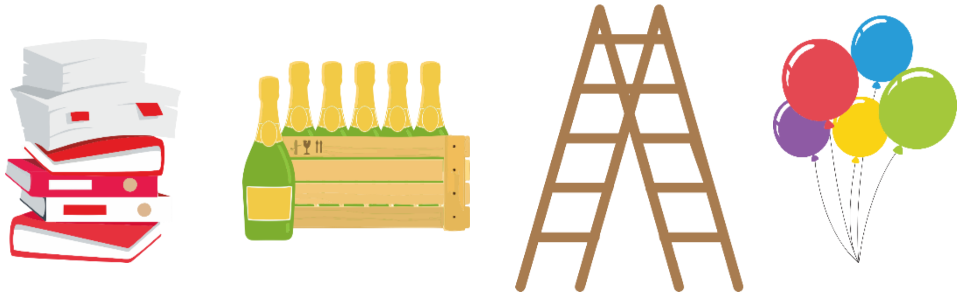
Multiple actions	Value Oracle		Value and Demand Oracle	
	Lower bound (pos)	Upper bound (neg)	Lower bound (pos)	Upper bound (neg)
Gross-substitutes	1 Dütting et al. [2021]	1	1	1
Submodular		No constant approx (if P≠NP) Ezra et al. [2024a]	FPTAS	> 1 Dütting et al. [2024c]
XOS		No better than $\Omega(n^{1/2})$ (if P≠NP) Ezra et al. [2024a]	FPTAS	> 1
Subadditive		No better than $\Omega(n^{1/2})$	FPTAS Dütting et al. [2021] Dütting et al. [2024a]	> 1
Super-modular	1 Dütting et al. [2024b] Deo-Campo Vuong et al. [2024]	1	1	1

Dutting et al. 2021: Dutting Ezra Feldman Kesselheim. Combinatorial Contracts. FOCS'21
Dutting et al. 2024a: Dutting Ezra Feldman Kesselheim. Multi-Agent Combinatorial Contracts. Working paper
Dutting et al. 2024b: Dutting Feldman Gal-Tzur. Combinatorial Contracts Beyond Gross Substitutes. SODA'24
Dutting et. al 2024c: Dutting Feldman Gal-Tzur Robinstein. The Query Complexity of Contracts. Working paper
Deo-Campo Vuong et al. 2024: D.-C. Vuong Dughmi Patel Prasad. On Supermpodular Contracts. SODA'24
Ezra. et al. 2024: Ezra Feldman Schlesinger. The (In)Approximability of Combinatorial Contracts. ITCS'24

Multiple Actions: Summary

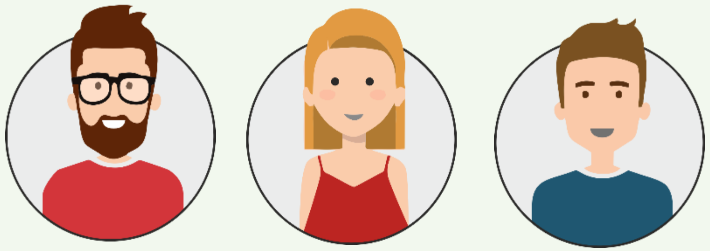
- **Key take-aways:**
  - Gross substitutes is a "frontier of tractability" for combinatorial contracts
  - Interesting connection to combinatorial auctions
- **Open problems:**
  - Tight bounds for submodular, XOS, and subadditive with value queries?
  - Beyond binary outcome?

Sources of Complexity in Contract Design



Combinatorial actions

[Duetting Ezra Feldman & Kesselheim FOCS’21] [Duetting Feldman & Gal-Tzur SODA’24], [Deo-Campo Vuong et al. SODA’24], [Ezra Feldman Schlesinger ITCS’24]



Multiple agents

[Babaioff Feldman Nisan EC’12] [Duetting Ezra Feldman & Kesselheim STOC’23] [Ezra Feldman Schlesinger ITCS’24]

Combinatorial Agency Model

[Babaioff Feldman Nisan 2006, DEFK'23]

- n agents
- Binary action: $A_i = \{0,1\}$
  (0: no effort, 1: effort)
- Cost c_i: cost of effort (no effort = no cost)
- Binary outcome: {0,1}
- Reward 1 for success, 0 for failure
- Success probability function $f: \{0,1\}^n \rightarrow [0,1]$

Contracts and Objective

- Optimal (=linear) contract: $\alpha = (\alpha_1, \ldots, \alpha_n)$
  - $\alpha_i \geq 0$: payment to agent i for success
- **Agent's perspective:** Agent i prefers to exert effort (in equilibrium) iff

$$\underbrace{\alpha_i f(S) - c_i}_{\text{agent } i\text{'s utility under effort}} \geq \underbrace{\alpha_i f(S - \{i\})}_{\text{agent } i\text{'s utility under no effort}}$$

Contracts and Objective

- Optimal (=linear) contract: $\alpha = (\alpha_1, \dots, \alpha_n)$
  - $\alpha_i \geq 0$: payment to agent i for success
- **Agent's perspective:** Agent i prefers to exert effort (in equilibrium) iff

$$\alpha_i f(S) - c_i \geq \alpha_i f(S - \{i\})$$

$$\Rightarrow \quad \alpha_i = \frac{c_i}{f(i \mid S - \{i\})}$$ is the best way to incentivize agent i

"margin" of i w.r.t. S:
$f(i \mid S - i)$
$= f(S) - f(S - i)$

Contracts and Objective

- Optimal (=linear) contract: $\alpha = (\alpha_1, \ldots, \alpha_n)$
  - $\alpha_i \geq 0$: payment to agent i for success
- **Agent's perspective:** Agent i prefers to exert effort (in equilibrium) iff

$$\alpha_i f(S) - c_i \geq \alpha_i f(S - \{i\})$$

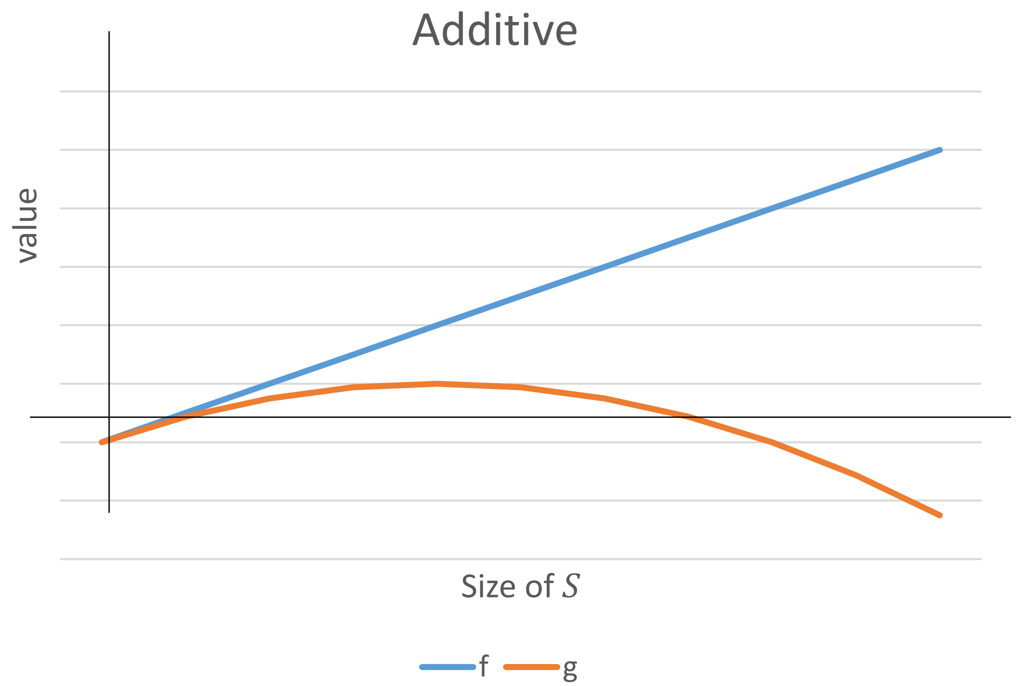
$$\Rightarrow \quad \alpha_i = \frac{c_i}{f(i \mid S - \{i\})}$$ is the best way to incentivize agent i

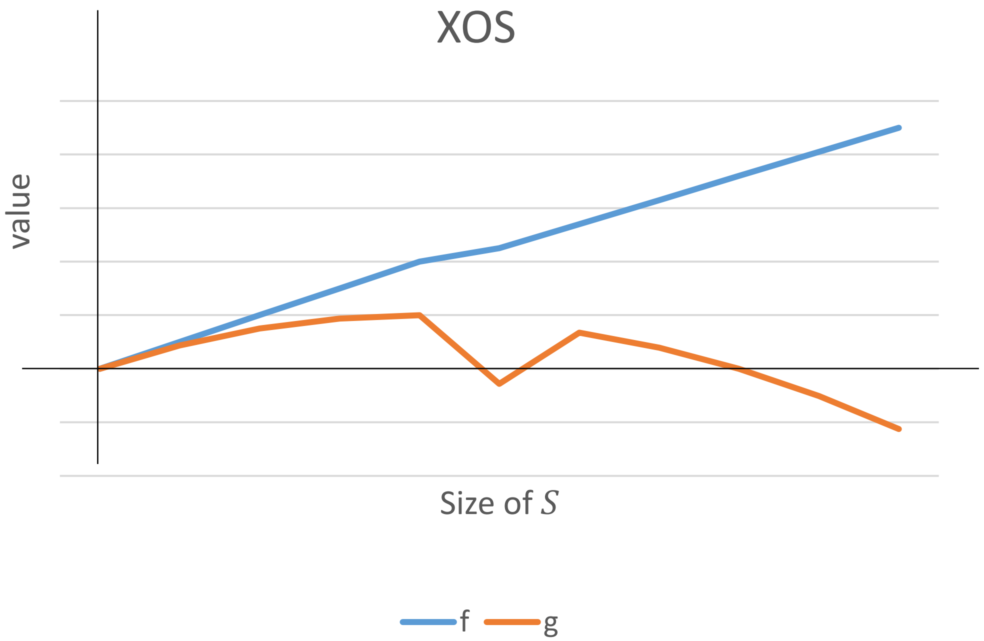
- **Principal's perspective:** Find the set of agents S that maximizes

$$g(S) = f(S)\left(1 - \sum_{i \in S} \frac{c_i}{f(i \mid S - \{i\})}\right)$$

- **Problem**: Compute optimal contract for submodular/XOS/subadditive f
- **Challenge**: Even if f is highly structured, g may be highly non-structured

Contracts and Objective



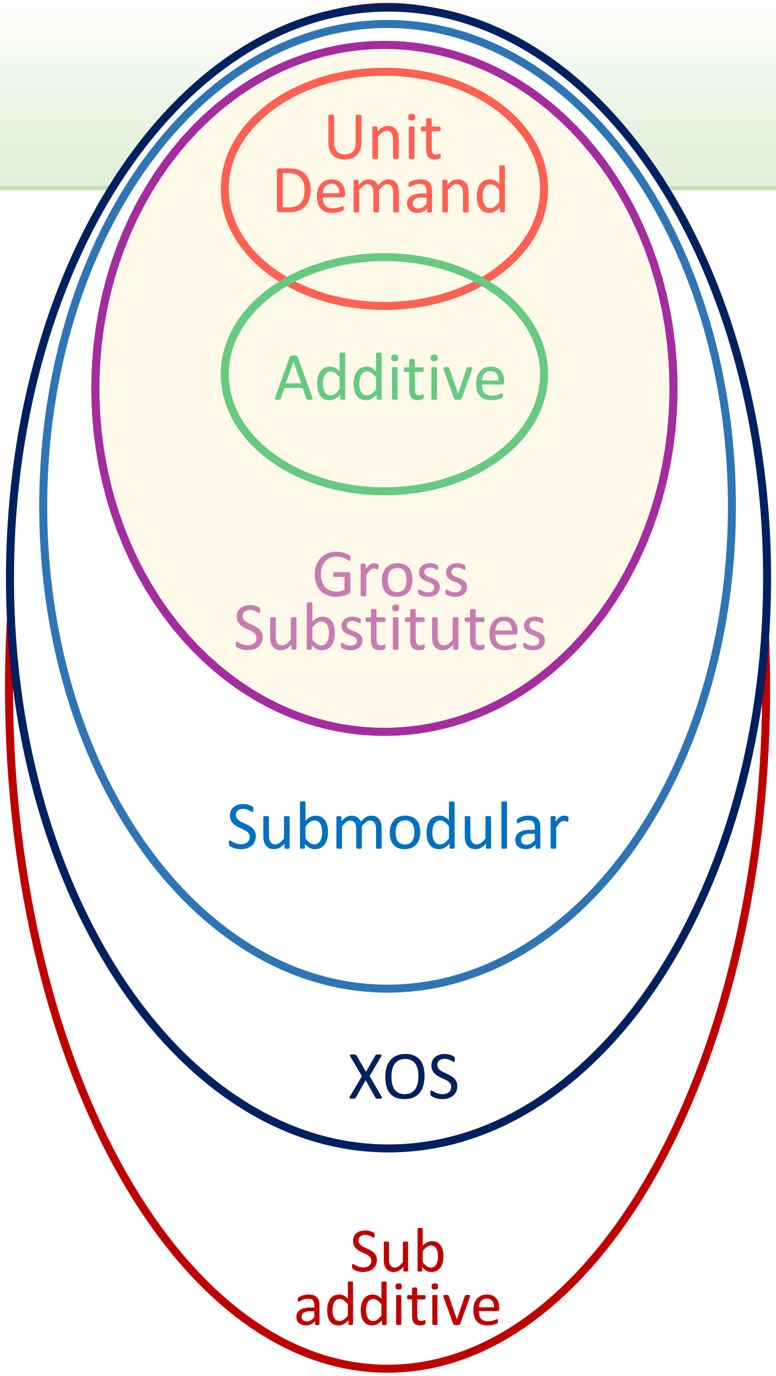


Submodular/XOS/Subadditive f

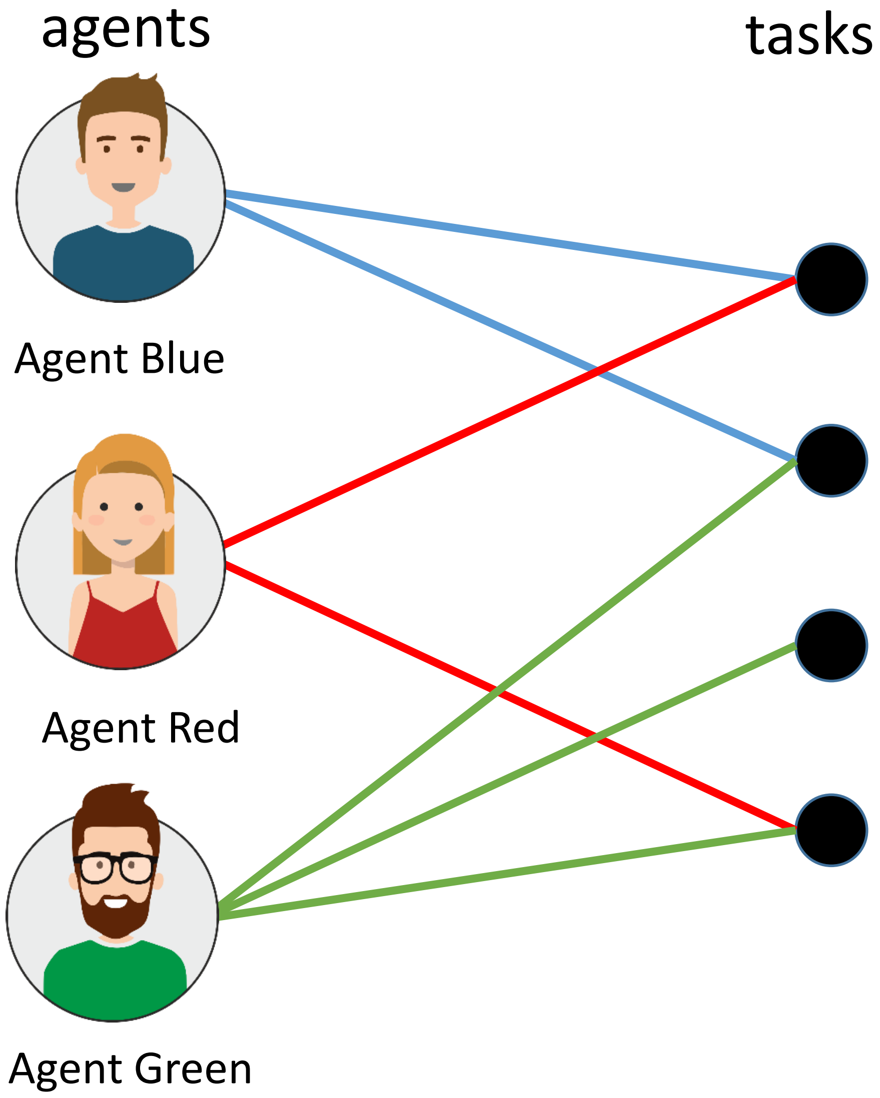
Submodular: $f(i \mid S) \geq f(i \mid T)$ for $S \subseteq T, j \notin T$
(decreasing marginal value)

XOS: maximum over additive
(aka: fractionally subadditive)

Subadditive: $f(S) + f(T) \geq f(S \cup T)$



Coverage Function (submodular)



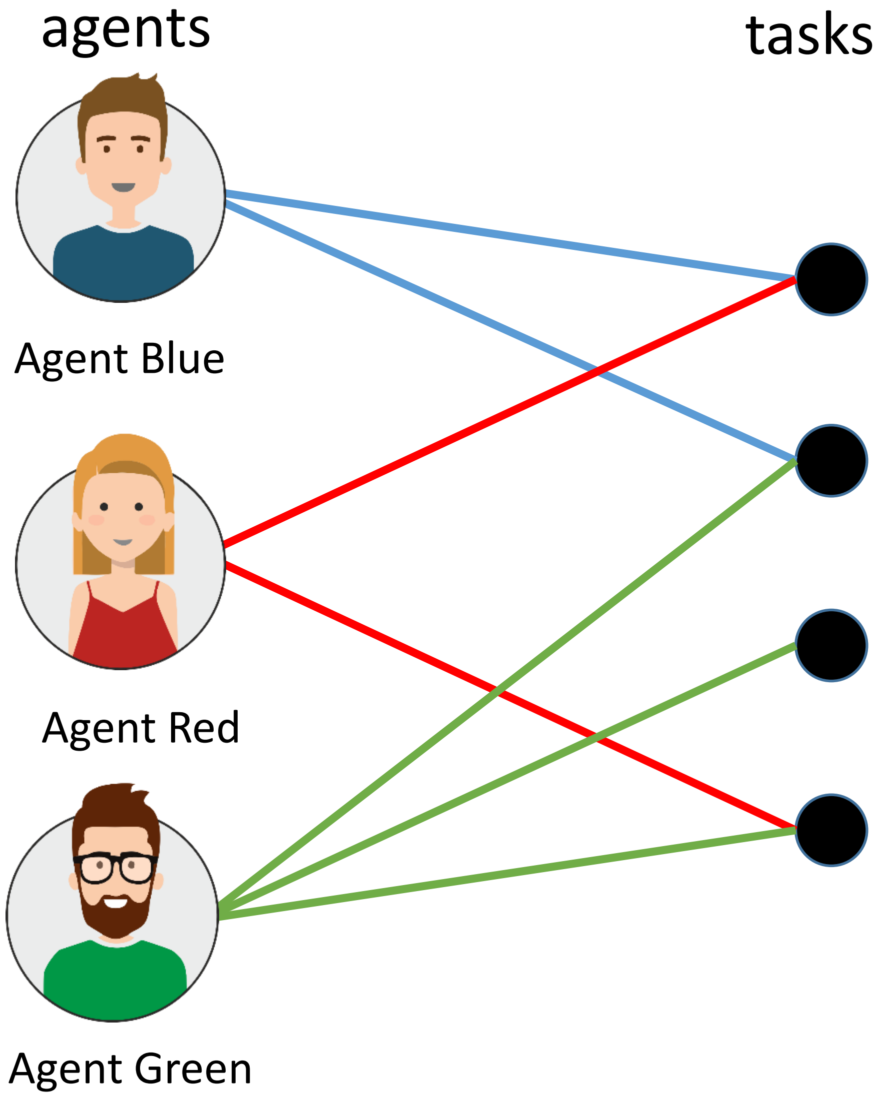
f(set of agents) $=$
tasks covered by these agents

e.g.:

$f($  $) = 2$

$f($  |  $) = 1$

Coverage Function (submodular)



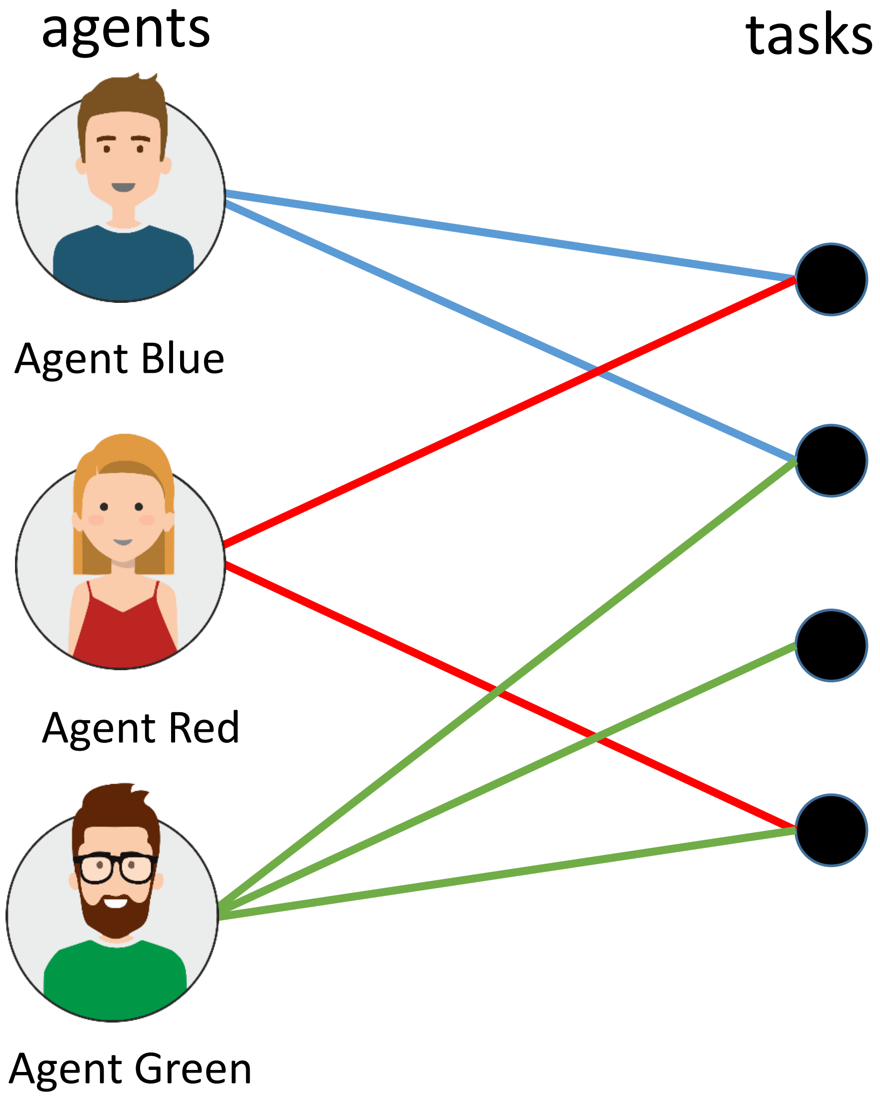
Principal's objective:

$$g(S) = f(S)\left(1 - \sum_{i \in S} \frac{c_i}{f(i \mid S - \{i\})}\right)$$

Total # tasks covered by S

tasks covered uniquely by agent i

Coverage Function (submodular)



Principal's objective:

$$g(S) = f(S)\left(1 - \sum_{i \in S} \frac{c_i}{f(i \mid S - \{i\})}\right)$$

Total # tasks covered by S

tasks covered uniquely by agent i

Unique coverage is hard to approximate within a constant factor [Demaine Feige Hajiaghayi Salavatipour 2006]

Main Results

(+) There is a poly-time algorithm for finding a constant-approximation contract for submodular f, using value oracle, and for XOS f, using value and demand oracles [DEFK'23]

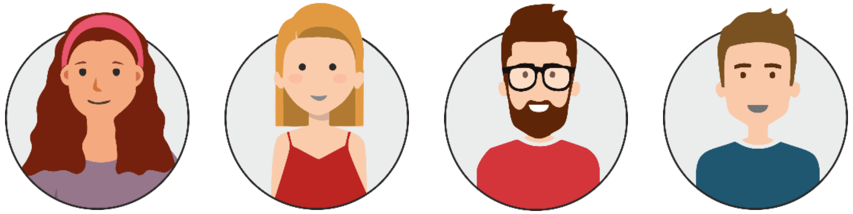
(-) No better than constant-approximation, even for submodular f, and even with both value and demand oracles [DEFK'23, DEFK'24]

- For additive f, it is NP-hard to find the optimal contract, but there is a an FPTAS
- No better than $\Omega(\sqrt{n})$-approximation for subadditive f (even for f constant close to submodular)

Proof Sketch: constant approximation for XOS

Goal: Find a set U satisfying $g(U) \geq \text{const} \cdot g(S^*)$

c_i
p_i



Let T be the demand set under prices $p_i = \frac{1}{2}\sqrt{c_i f(S^*)}$

Lemma 1: $f(T) \geq \frac{1}{2} f(S^*)$

Lemma 2: For every set U, if $f(i \mid U - \{i\}) \geq \sqrt{2 c_i f(U)}$ $\forall i \in U$, then $g(U) \geq \frac{1}{2} f(U)$

By definition: $f(i \mid T - \{i\}) \geq p_i = \frac{1}{2}\sqrt{c_i f(S^*)}$

$\Rightarrow$ we want: $\frac{1}{2}\sqrt{c_i f(S^*)} \geq \sqrt{2 c_i f(T)}$

so that:

$$g(T) \geq \frac{1}{2} f(T) \geq \frac{1}{4} f(S^*) \geq \frac{1}{4} g(S^*)$$

Problem: $f(T)$ may be too large

Idea: remove agents from T until inequality is satisfied

Problem: marginals may decrease (unlike submodular)

Theorem [scaling property of XOS]: for every set T and every $\Psi < f(T)$, can compute a set $U \subseteq T$ such that

$$\frac{1}{2}\Psi \leq f(U) \leq \Psi \quad \text{and} \quad f(i \mid U - \{i\}) \geq \frac{1}{2} f(i \mid T - \{i\})$$

Setting $\Psi = \frac{1}{32} f(S^*)$ now gives $f(i \mid U - \{i\}) \geq \frac{1}{4}\sqrt{c_i f(S^*)} \geq \sqrt{2 c_i f(U)}$, yielding:

$$g(U) \geq \frac{1}{2} f(U) \geq \text{const} \cdot f(S^*) \geq \text{const} \cdot g(S^*)$$

■

Multiple Agents: Overview

Multiple agents	Value Oracle		Value and Demand Oracle	
	Lower bound (pos)	Upper bound (neg)	Lower bound (pos)	Upper bound (neg)
Additive	FPTAS Dütting et al. [2023a]	OPT is NP-hard	FPTAS	OPT is NP-hard Dütting et al. [2023a]
Gross-substitutes	Constant approx	OPT is NP-hard	Constant approx	OPT is NP-hard
Submodular	Constant approx Dütting et al. [2023a]	No PTAS (if P$\neq$NP) Ezra et al. [2024a]	Constant approx	No PTAS Dütting et al. [2024a]
XOS		No better than $\Omega(n^{1/6})$ Ezra et al. [2024a]	Constant approx Dütting et al. [2023a]	NO PTAS Dütting et al. [2023a]
Subadditive	$O(n)$-approx	No better than $\Omega(n^{1/6})$		No better than $\Omega(n^{1/2})$ Dütting et al. [2023a]
Super-modular		No constant approx		No constant approx Deo-Campo Vuong et al. [2024]

Dutting et al. 2023a: Dutting Ezra Feldman Kesselheim. Multi-Agent Contracts. FOCS’21
Dutting et al. 2024a: Dutting Ezra Feldman Kesselheim. Multi-Agent Combinatorial Contracts. Working paper
Deo-Campo Vuong et al. 2024: D.-C. Vuong Dughmi Patel Prasad. On Supermpodular Contracts. SODA’24
Ezra. et al. 2024: Ezra Feldman Schlesinger. The (In)Approximability of Combinatorial Contracts. ITCS’24

Multiple Agents: Summary

- **Key take-aways:**
  - Submodular as a frontier for poly-time constant-factor approximation
  - Non-standard use of prices & demand queries
  - New scaling property of XOS functions, that may be of independent interest
- **Open problems:**
  - Gap between upper and lower bounds for GS
    - One of the few problems that is hard for GS
    - Does it admit an (F)PTAS?
  - Beyond binary outcome?

New: Multiple Agents & Multiple Actions

Multiple agents, each of which takes a set of actions [Duetting Ezra Feldman Kesselheim'24]:
Provably very different from either of the special cases

- Constraints on the α_i's incentivizing S are 2-directional ⇒ No simple formula for the α_i's
- Equilibrium existence is non-trivial (requires potential function argument)
- Not all sets can be incentivized
- For submodular f , if others do less, it might be beneficial to do less
- …

Main result (+): Poly-time O(1)-approximation for submodular with value and demand queries

Main result (-): No PTAS for submodular, with value and demand queries

Results require very different tools than ones used in previous special cases

Main Take Aways

- Contract theory is a new frontier in AGT
- Complexity and approximation shed new light on contract design
- Interesting connections to combinatorial auctions and other combinatorial optimization problems
  - E.g., gross substitutes as tractability frontier
  - E.g., submodular as frontier for approximation
- Many fundamental problems still open

Thank You!

Resources

- EC’19 & STOC’22 Tutorials (Duetting and Talgam-Cohen]
- Forthcoming (FnTTCS): Algorithmic Contract Theory: A Survey (Duetting Feldman Talgam-Cohen)

24th Max Planck Advanced Course on the Foundations of Computer Science

26 - 30 August 2024, Saarbrücken, Germany

ADFOCS 2024

Algorithmic Game Theory

Paul Duetting

Google Research, Zurich

Prophet Inequalities

Elias Koutsoupias

University of Oxford

Mechanism Design

Michal Feldman

Tel-Aviv University

Algorithmic Contract Theory

Bernhard von Stengel

London School of Economics and Political Science (LSE)

Equilibrium Computation in Games

max planck institut informatik

Early registration deadline: **July 31, 2024**